

构建自定义的**WinCE**

构建自定义的WinCE

- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ 配置
- ▶ 其他

Windows Embedded CE操作系统设计

- ▶ 为一个新的硬件设备定制WinCE6.0操作系统，一般需要完成以下几个主要步骤：
 1. 针对特定的硬件设备创建板级支持包(Board Support Package缩写为BSP)，BSP必须包括BOOTLOADER、OEM适配层(OEM Adaptation Layer缩写为OAL)和一些必要的驱动。
 2. 利用创建的BSP，定制一个系统设计(OS Design)。即通过VS2005创建一个Platform Builder的工程。该工程可编译产生最终的运行时映像文件(Run-time Image)。
 3. 针对板上的外围设备创建相关驱动，并添加到BSP中。
 4. 通过创建子工程和Catalog Items的方式，修改OS Design。
 5. 编译OS Design，下载编译得到的运行时映像文件到目标设备。此时，可通过远程调试工具进行调试。
 6. 在完成所有的调试工作之后，导出该运行时映像对应的SDK(Software Development Kit)，应用程序的开发人员可基于此SDK编写该设备的应用程序。

Windows Embedded CE操作系统设计

▶ WinCE6.0的开发环境是Platform Builder6.0

它作为VS2005的一个插件来用。所以，需先安装VS2005，SPI。PB6.0目前不支持VS2008。然后顺序安装PB6.0、PB6.0的SPI、R2。虽然微软没有严格说明，但还是建议将这些软件都默认安装到C盘，以免出现意外。

安装完成后，C盘的根目录下产生一个WINCE600的目录，目录里包括WinCE6.0的源代码、BSP和一些编译WinCE时需要用到的工具和库文件。(见下页)

另外还产生一个C:\Program Files\Microsoft Platform Builder\6.00的目录，这里主要包括PB6.0的开发环境工具、帮助文件和其他一些开发时用到的工具。



Windows Embedded CE操作系统设计

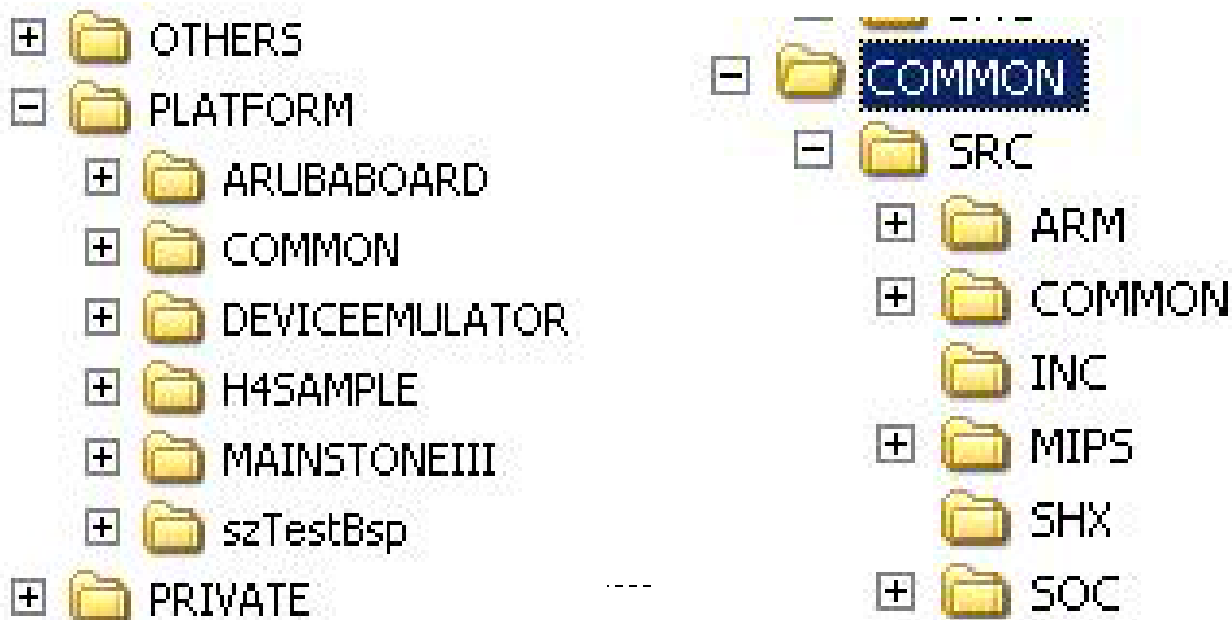
WINCE600的目录组织

目录	说明
CRC	存放了一个PB6.0安装时用到的校验文件 crc.ini
OSDesigns	用来存放PB的工程，对应PB5.0中的 PBWORKSPACES
OTHERS	存放了一些运行库、用于编译操作系统的二 进制文件、注册表文件、批处理文件
PLATFORM	存放了和硬件平台相关的BSP及MCU相关的代 码和其他一些文件
PRIVATE	存放了WinCE6.0开放的源代码
PUBLIC	存放了WinCE6.0的相关组件，这里是纯软件 的代码和库
SDK	存放了用于编译WinCE6.0的相关工具和DLL文 件



Windows Embedded CE操作系统设计

PLATFORM目录下除了COMMON目录外，其他都是特定硬件平台的BSP，而COMMON目录中则包涵了不同体系结构的相关代码(ARM、X86、MIPS、SHX)、与平台无关的代码(COMMON)和SOC的相关代码(SOC)，在移植BSP的过程中，这些代码最好别直接修改。



构建自定义的WinCE

- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ 配置
- ▶ 其他

平台向导(Platform Wizard)

- ▶ PB提供了二个向导

- ▶ WinCE 6的定制向导

- 帮助你一步步的创建一个基于Windows CE的平台。

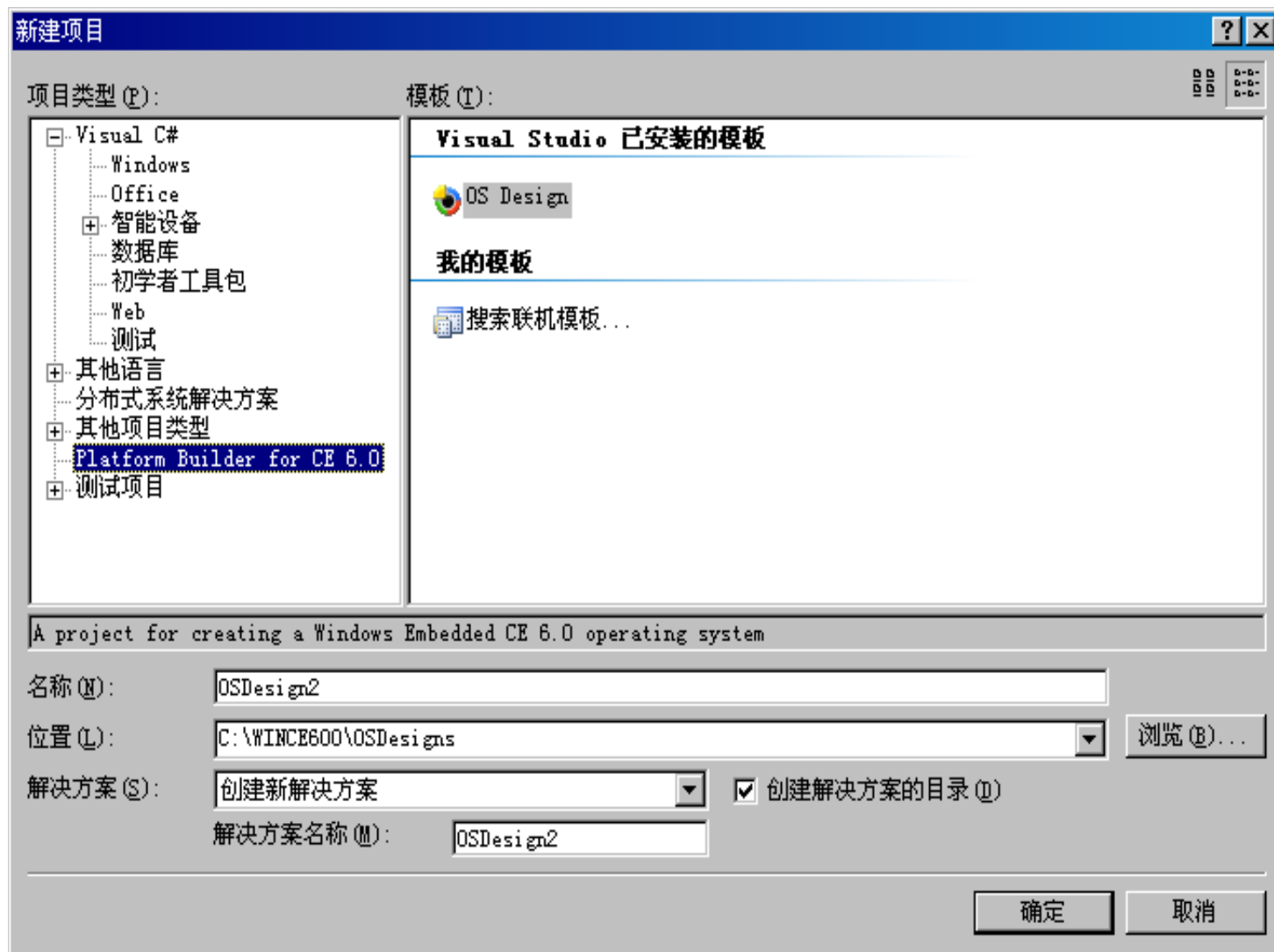
- ▶ 子项目向导

- 在PB下，以子项目形式创建应用程序、驱动等。

- ▶ 下面是WinCE 6定制向导的展示

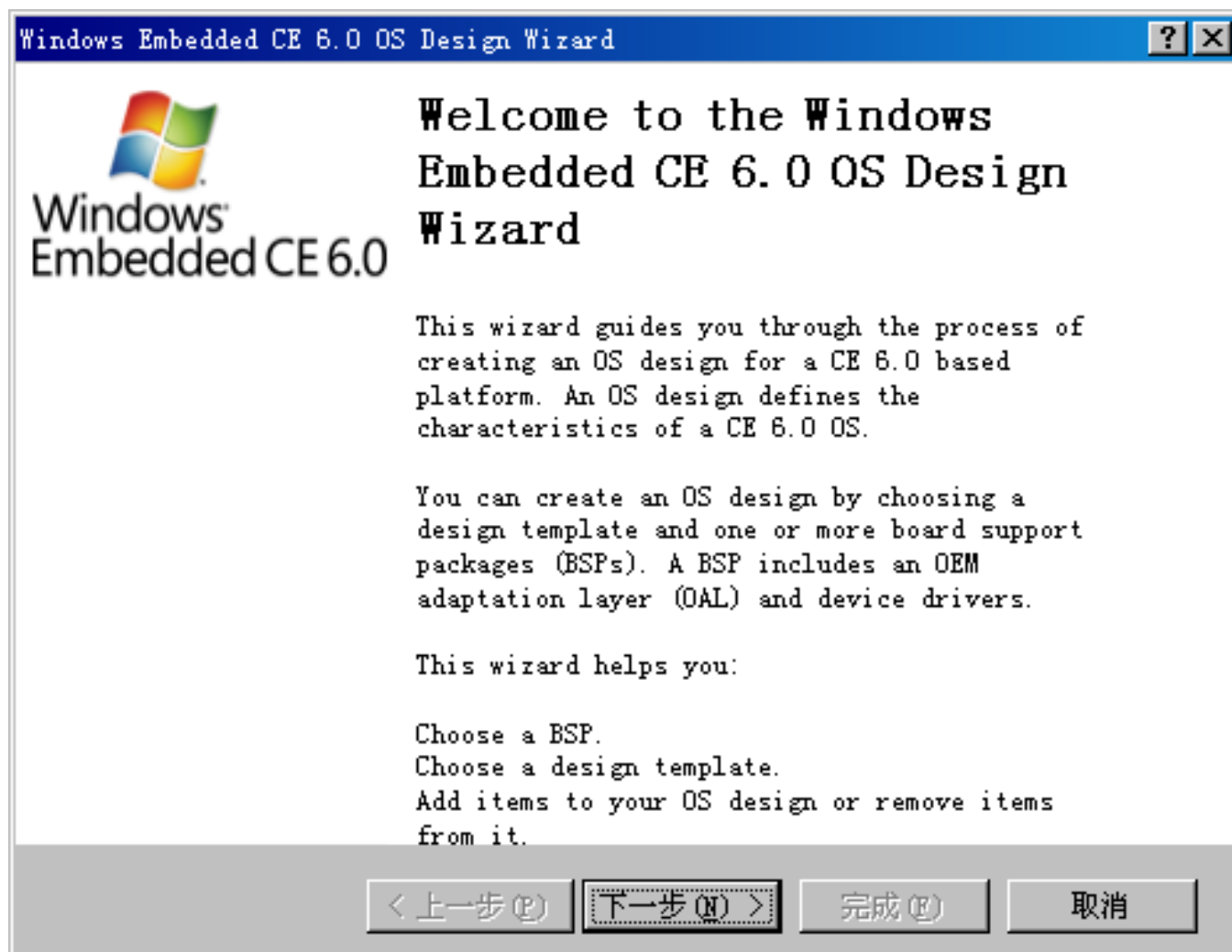
平台向导(Platform Wizard)

新建 WinCE 6 项目



平台向导(Platform Wizard)

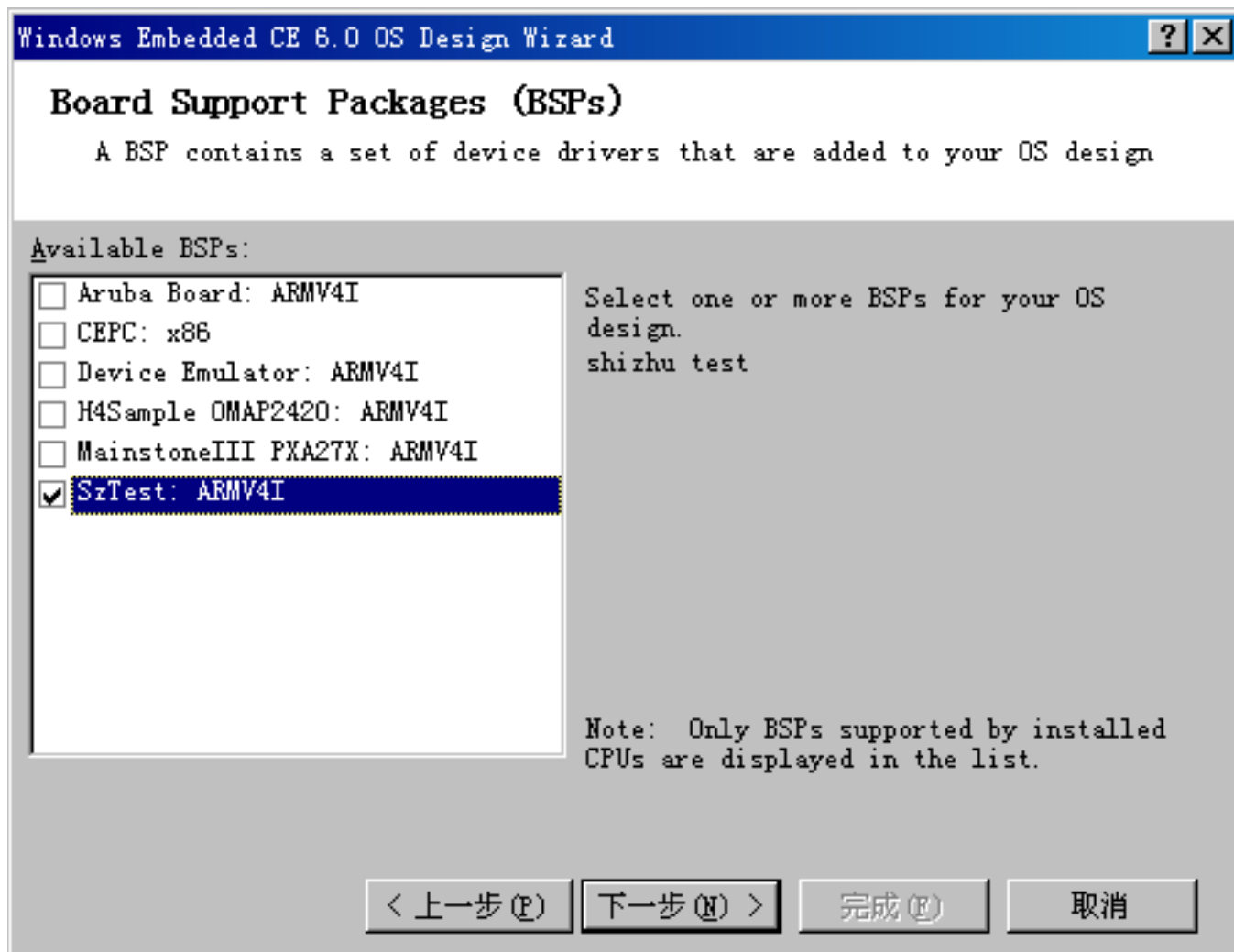
向导
欢迎
信息



平台向导(Platform Wizard)

▶ 选择BSP

这个界面中，需选择该平台所用的BSP（板级支持包），一般来说，一个平台只会针对一个硬件平台，而BSP与硬件平台是一一对应的关系。



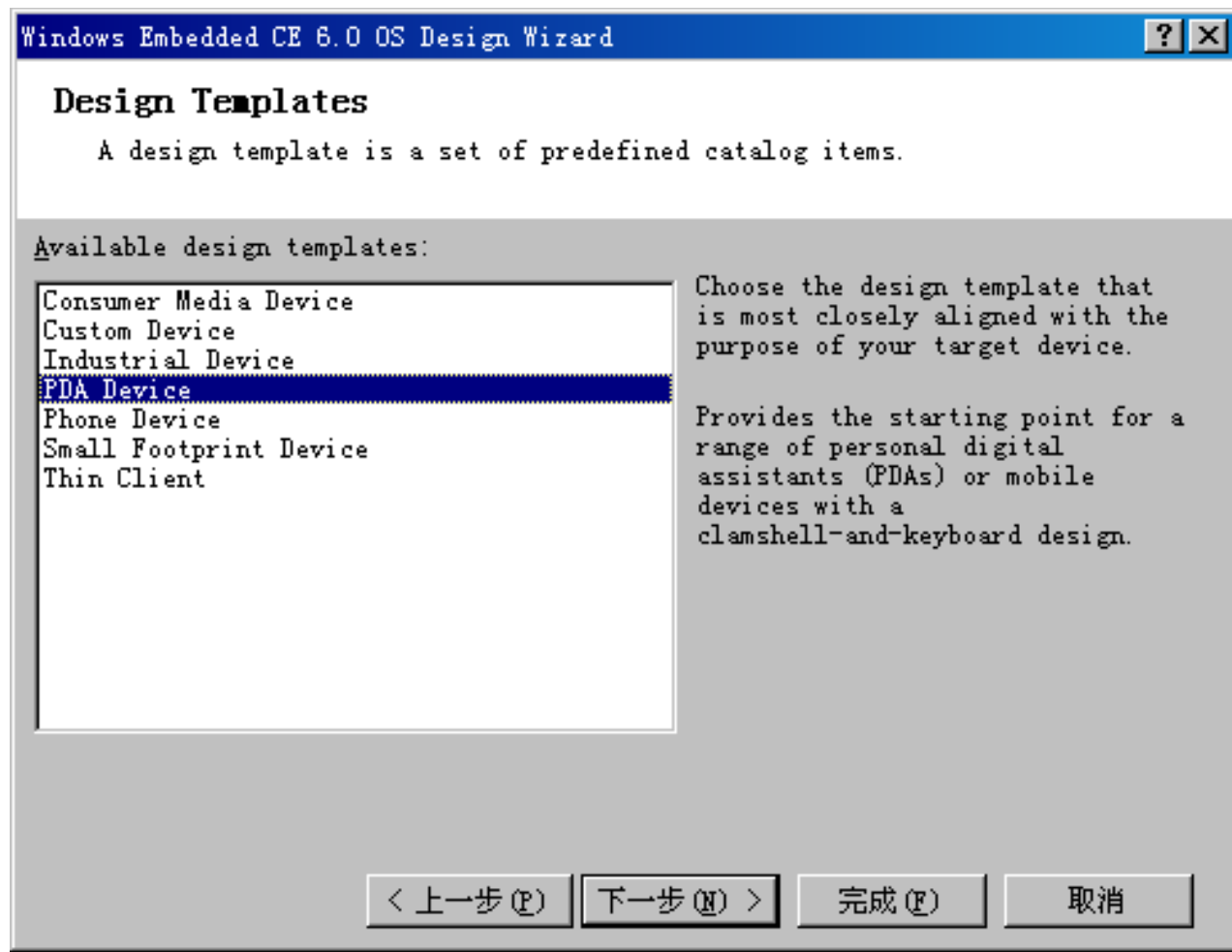
平台向导(Platform Wizard)

▶ 选择设备类型

有很多默认设备类型的模板，如：PDA

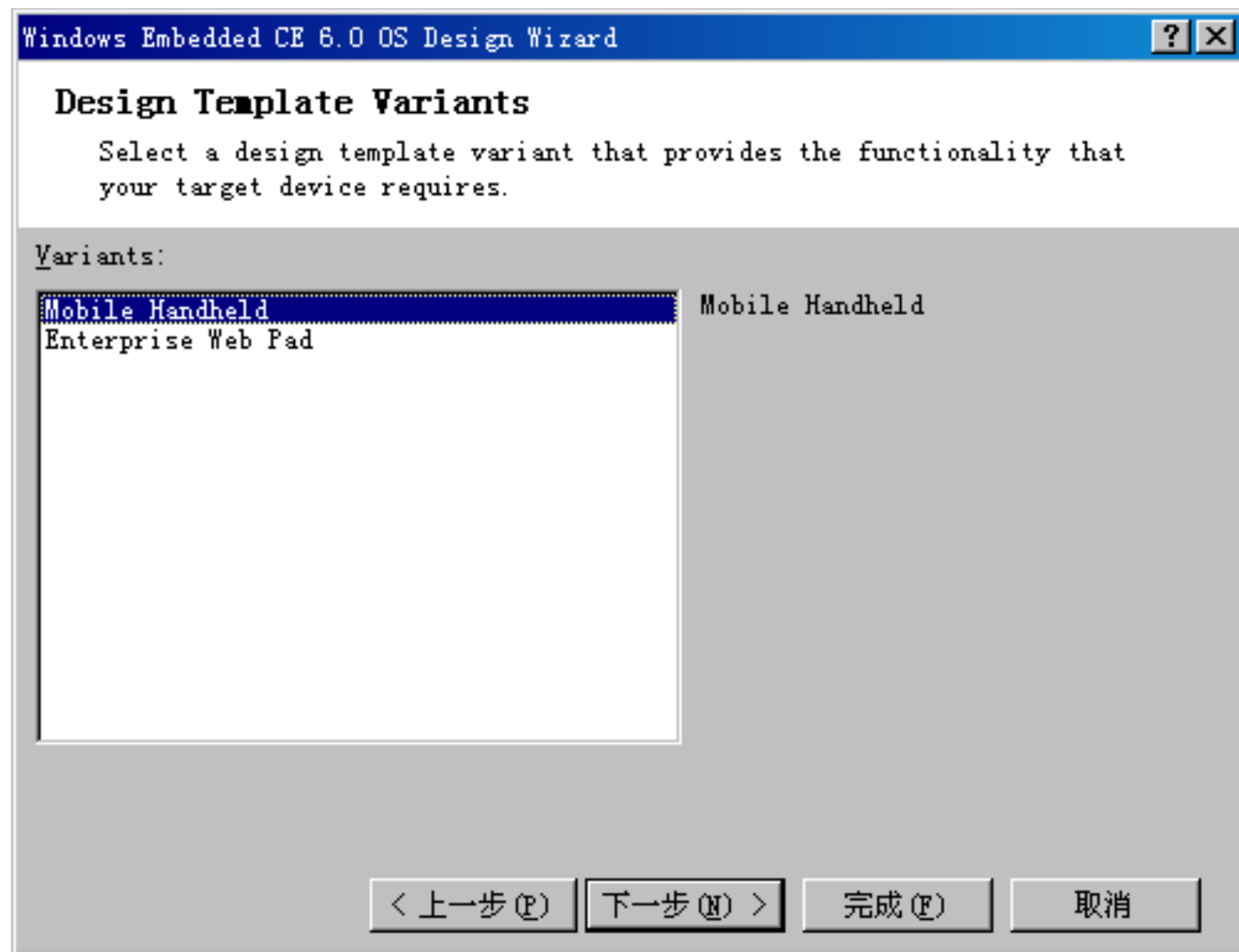
Device、Phone Device等。这些设计模板已经根据不同的设备类型进行了预先的定制，加入了适当的组件。

如果想完全从头设计一个平台，可以选择“Custom Device”



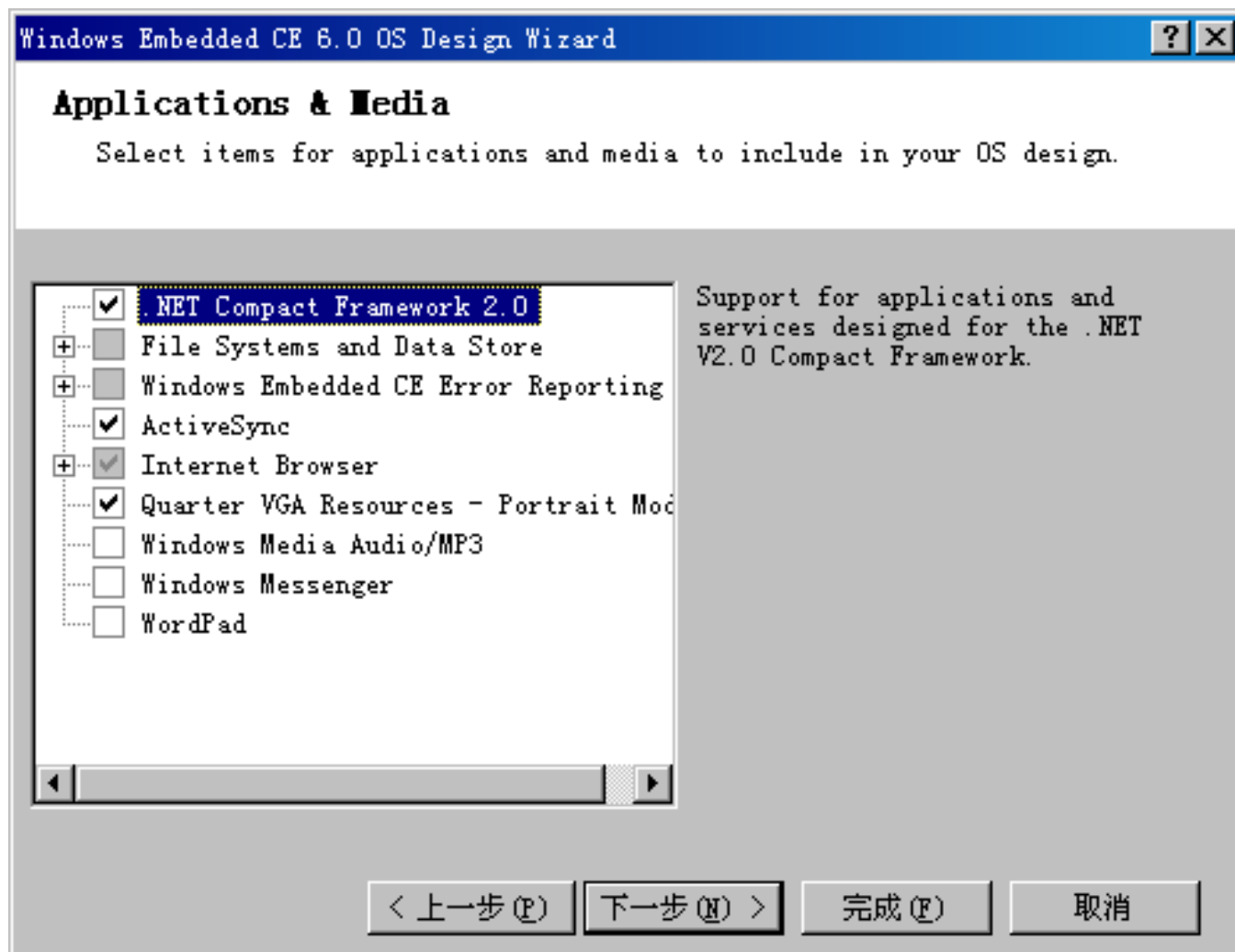
平台向导(Platform Wizard)

- ▶ 选择设备类型
选择最终的设计模板



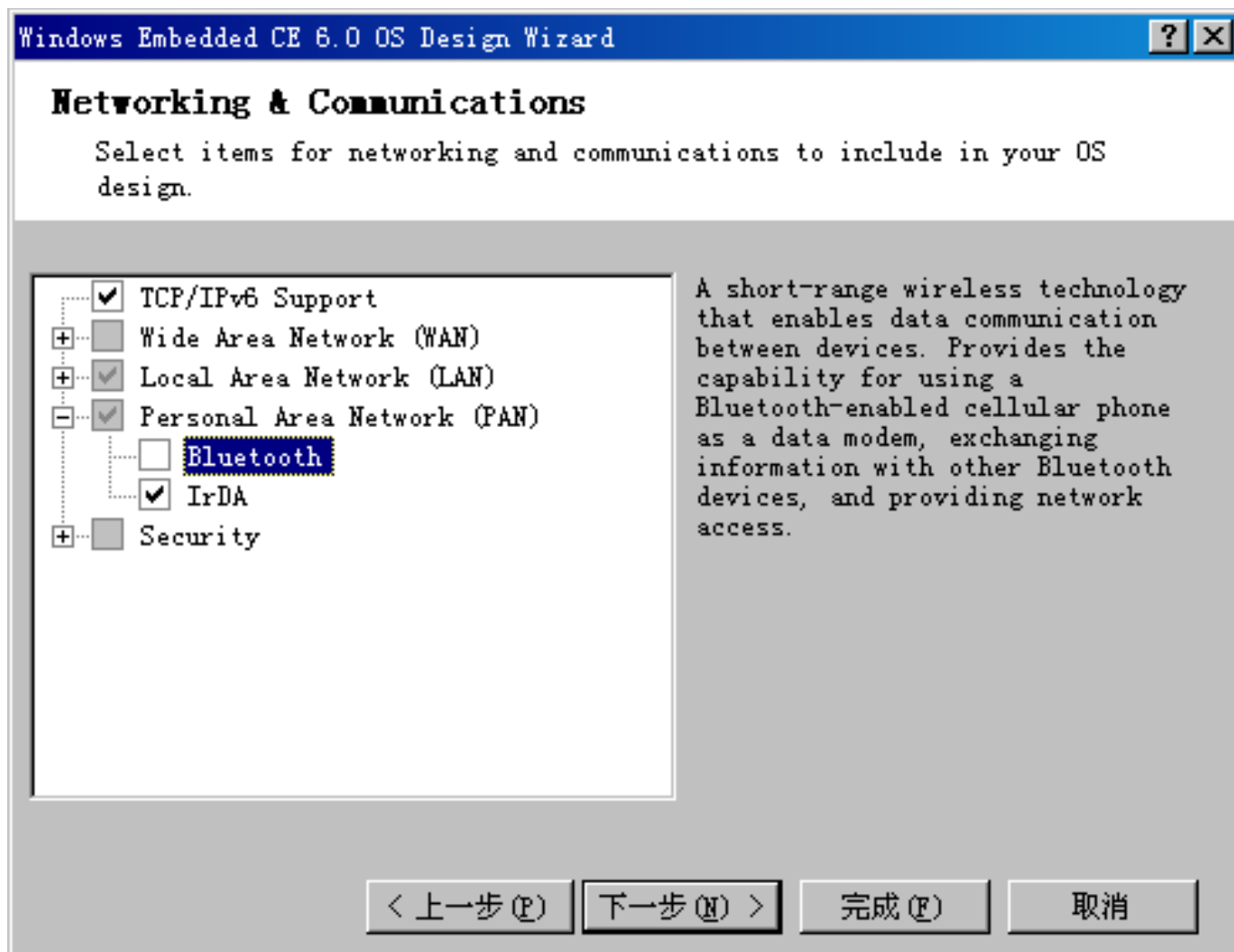
平台向导(Platform Wizard)

▶ 应用程序和多媒体组件选择



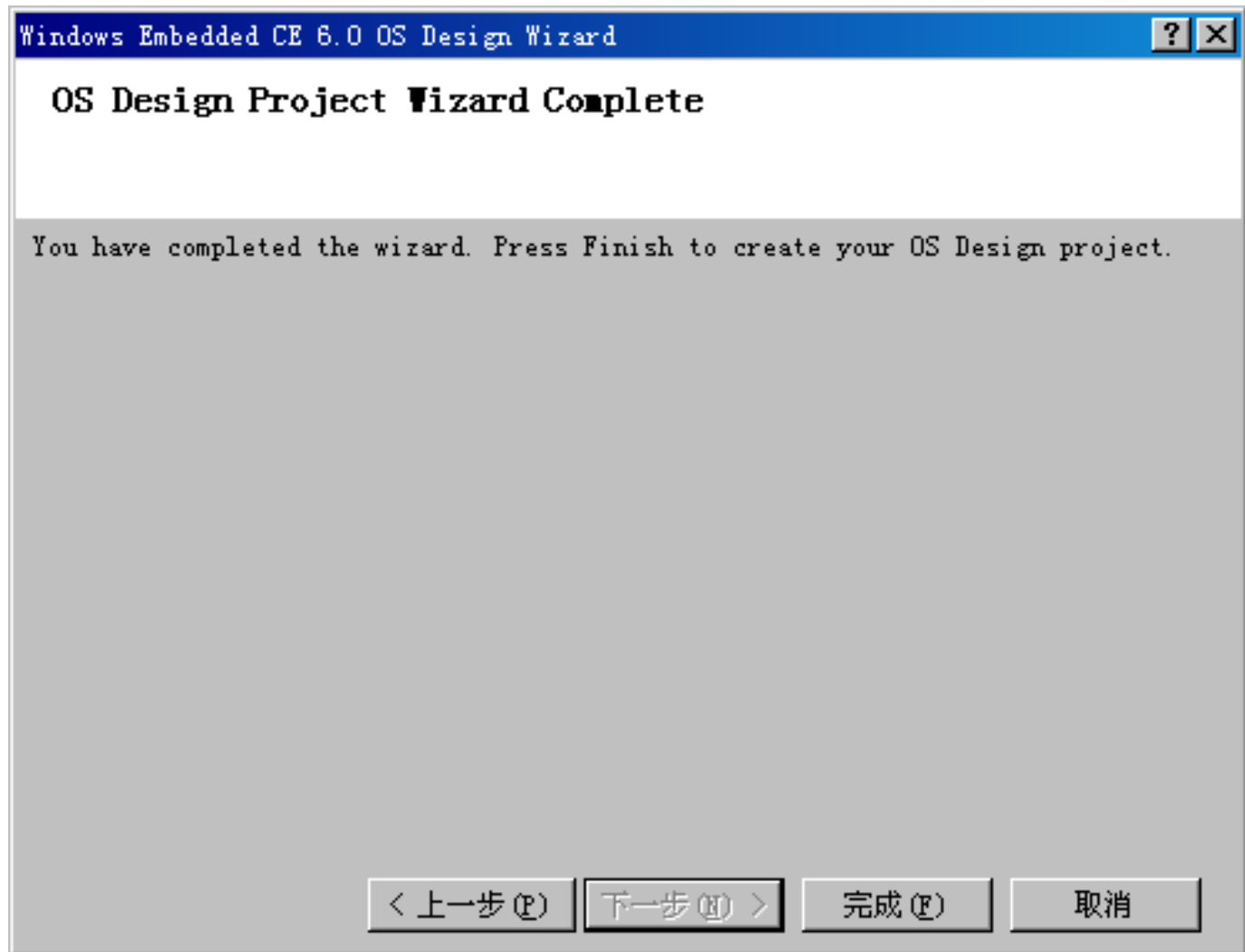
平台向导(Platform Wizard)

► 网络和通信组件选择



平台向导(Platform Wizard)

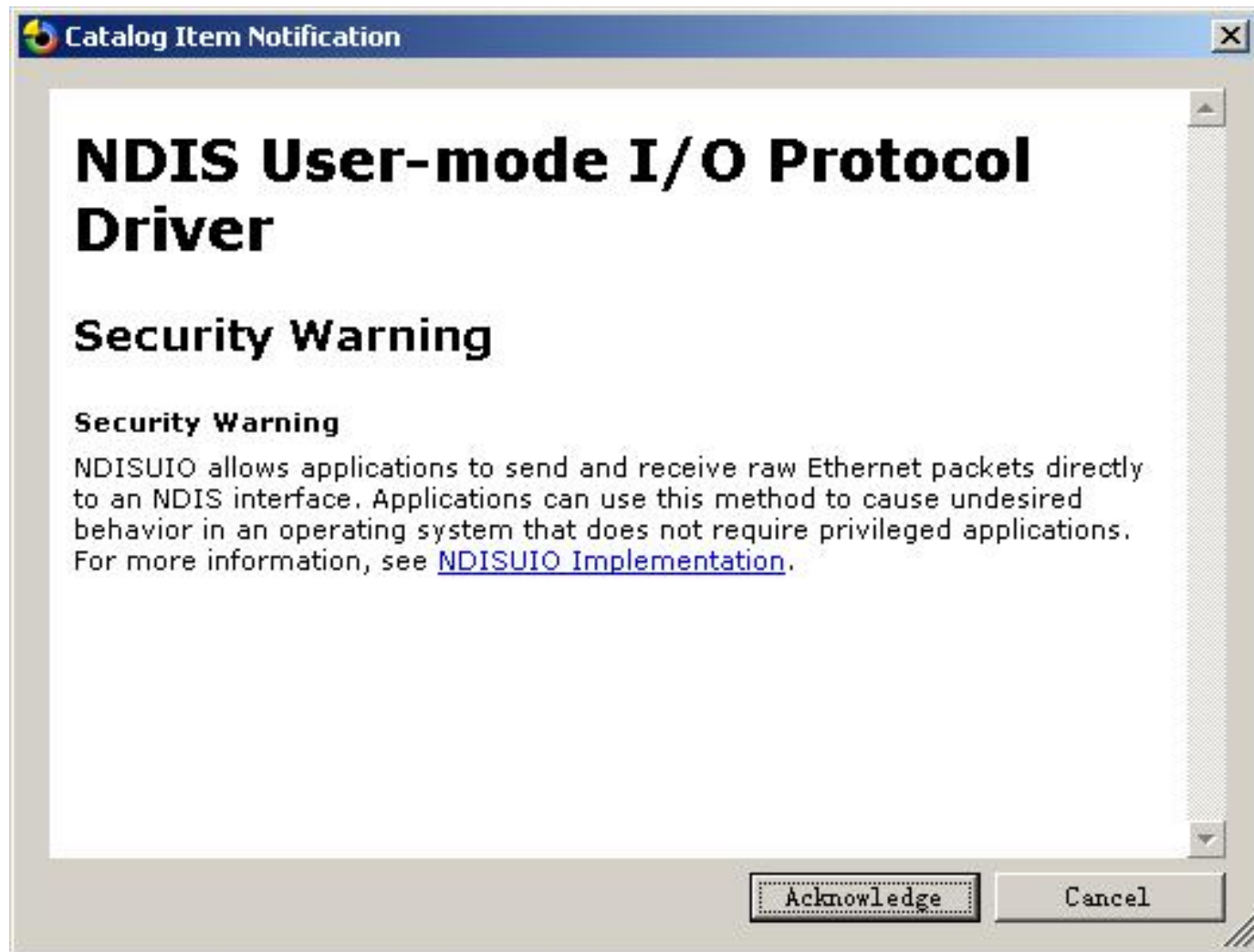
完成
创建
向导



平台向导(Platform Wizard)

▶ 安全警告

根据所选
组件，
可能会
弹出“
安全警告”



平台向导(Platform Wizard)

- ▶ 根据需要，对生成的设计进行编辑
- ▶ 生成Windows Embedded CE（10多分钟）

附：生成中文的

- ▶ 打开catalog items view
- ▶ 选择core os->cebase->international->locale specific support->chinese simplified->gb18030...和Monotype...复选上，继续展开fonts->simSun...->单选simsun & Nsimsun
- ▶ 在属性配置中选择locale,清除所有复选框，选上english和中文,默认使用中文。
- ▶ 编译
- ▶ 设置Connectivity options,包括内存和分辨率等的设置。

平台向导(Platform Wizard)

- ▶ 生成后，在发布目录中有代表OS的bin或nb0文件
 - ▶ nk.bin：二进制WinCE操作系统映像
 - ▶ eboot.bin：（可选）以太网BootLoader映像
 - ▶ nk.nbX：（可选）X可能是0，1，...视配置而定，这类文件可被烧进Flash ROM中，可直接从Flash ROM中本地执行。
 - ▶ Eboot.nbX：（可选）支持XIP的以太网BootLoader映像


一般，如某硬件平台具有XIP的Flash，则在开发时用Eboot下载nk.bin调试，在最终发布产品时，把nk.nbX烧到Flash中

平台向导(Platform Wizard)

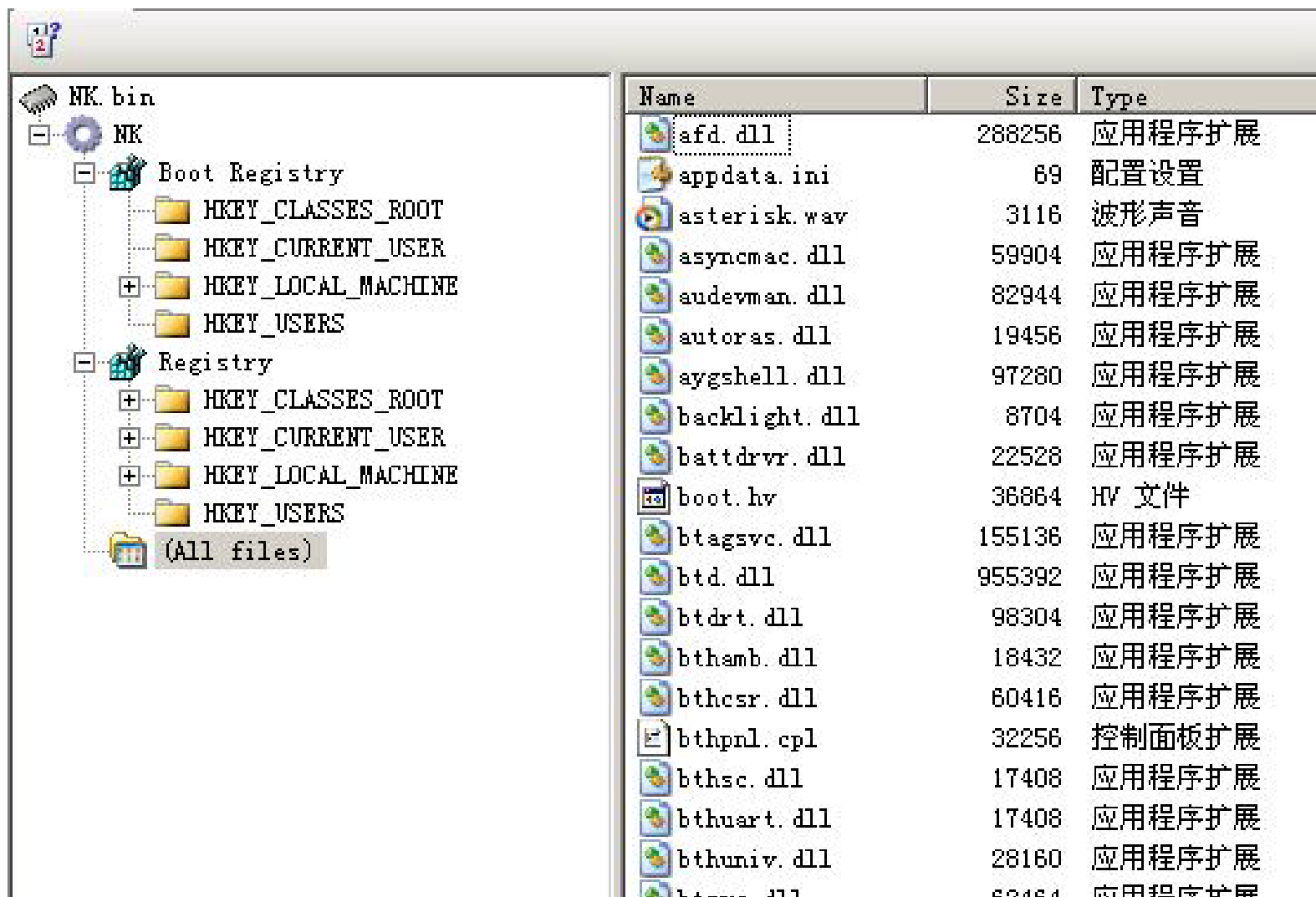
▶ 查看、比较映像文件

可查看已生成映像的注册表和其它文件，如下页图。

操作步骤：文件/打开，选对话框的“文件类型”为“Windows Embedded CE Run-Time Images”，然后找到已生成的nk.bin。

在查看映像界面中，可点击 ，选择另一映像文件进行比较。不同文件会用不同颜色标记。（红色为一，黄色为另一）

平台向导(Platform Wizard)



The screenshot displays the Platform Wizard interface. On the left, a tree view shows the file structure under 'NK bin'. The tree is expanded to show 'NK', which contains 'Boot Registry' and 'Registry'. Both 'Boot Registry' and 'Registry' contain subfolders for 'HKEY_CLASSES_ROOT', 'HKEY_CURRENT_USER', 'HKEY_LOCAL_MACHINE', and 'HKEY_USERS'. A '(All files)' folder is also visible at the bottom of the tree.

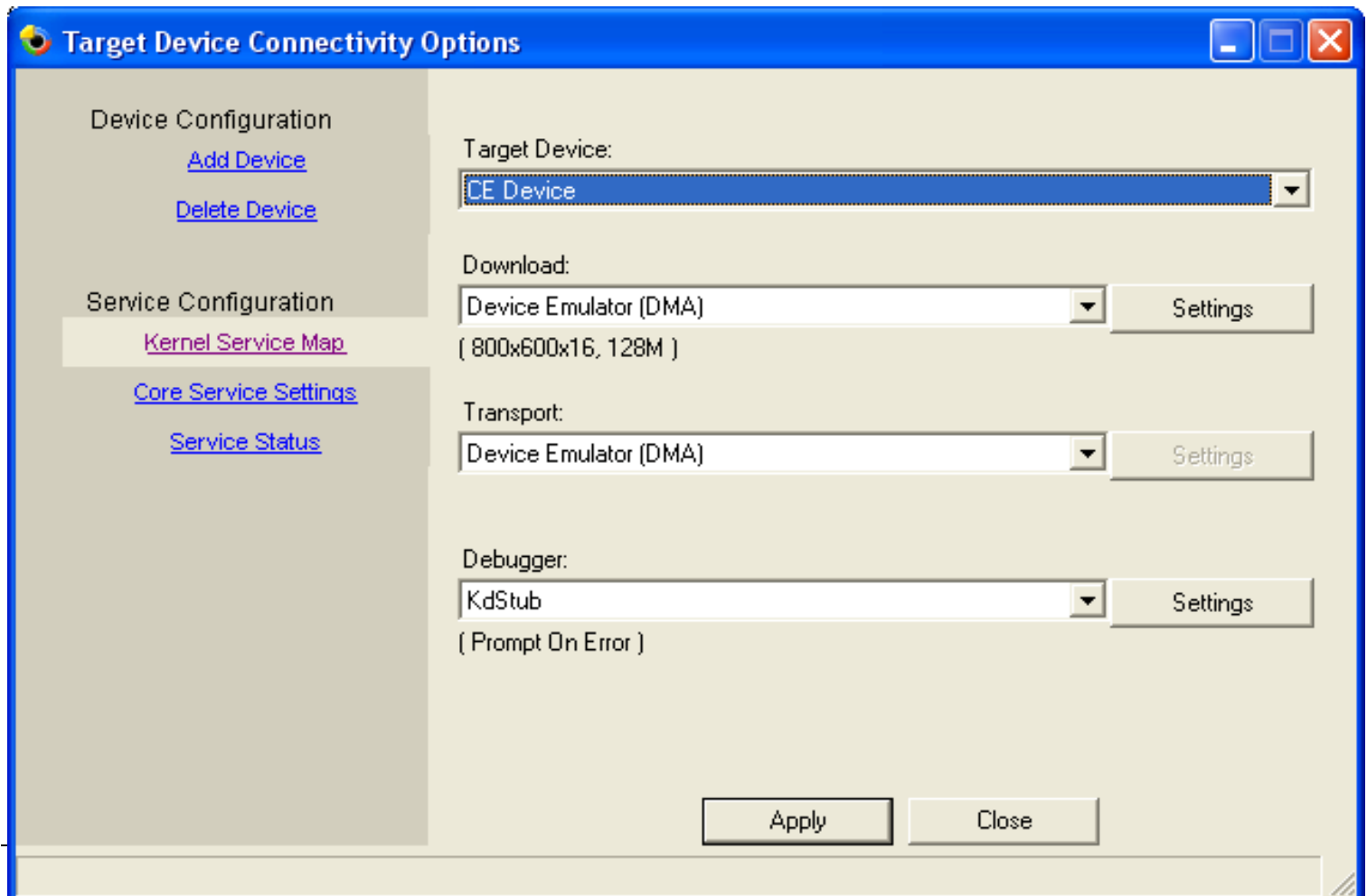
On the right, a table lists the files and their properties:

Name	Size	Type
afd.dll	288256	应用程序扩展
appdata.ini	69	配置设置
asterisk.wav	3116	波形声音
asynctac.dll	59904	应用程序扩展
audevman.dll	82944	应用程序扩展
autoras.dll	19456	应用程序扩展
aygshell.dll	97280	应用程序扩展
backlight.dll	8704	应用程序扩展
battdrv.dll	22528	应用程序扩展
boot.hv	36864	HV 文件
btagsvc.dll	155136	应用程序扩展
btd.dll	955392	应用程序扩展
btdrt.dll	98304	应用程序扩展
bthamb.dll	18432	应用程序扩展
bthcsr.dll	60416	应用程序扩展
bthpnl.cpl	32256	控制面板扩展
bthsc.dll	17408	应用程序扩展
bthuart.dll	17408	应用程序扩展
bthuniv.dll	28160	应用程序扩展
bthusb.dll	20480	应用程序扩展

在模拟器上运行WinCE 6

► 配置 connectivity options

选择菜单Target / Connectivity Options... Target Device, 照图配置。



在模拟器上运行WinCE 6

▶ 启动目标设备

选择菜单 Target / Attach Device, Visual Studio 会启动 Device Emulator 并且自动下载构建的操作系统映像。

▶ 断开目标设备

▶ 重启目标设备

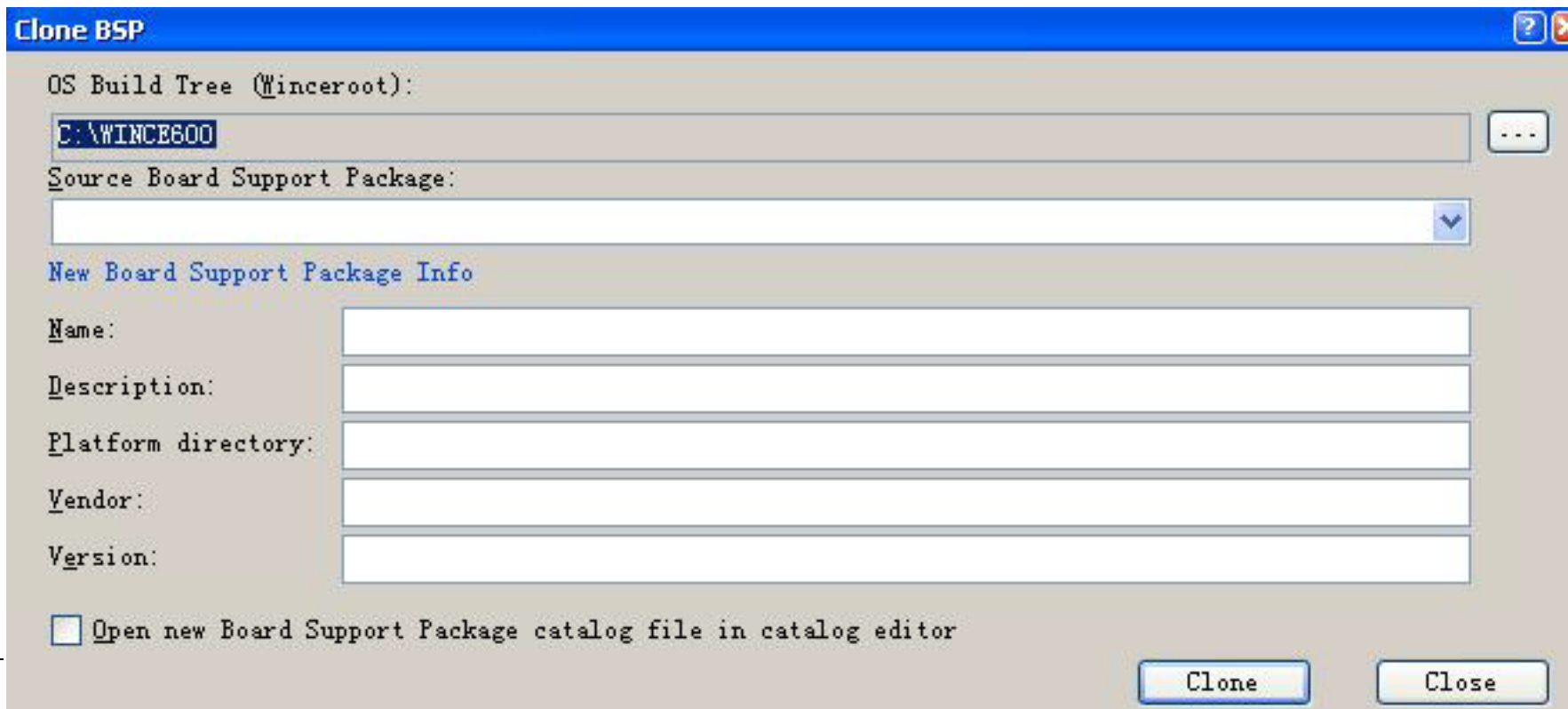


构建自定义的WinCE

- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ 配置
- ▶ 其他

克隆BSP

- ▶ PB可以克隆BSP，并将它加在你自己设计的OS中。
- ▶ 方法：
选择 Tools | Platform Builder for CE6.0 | Clone BSP 菜单来打开 Clone BSP 对话框；



克隆BSP

在Source Board Support Package下拉菜单中选择要被克隆的BSP;

在New Board Support Package Info 区域中为新的BSP填写相关信息;

最后, 点击 **Clone** 按钮. Clone BSP 工具会基于你所选的板级支持包来克隆一个新的板级支持包。新的BSP可用在你自己设计的OS中

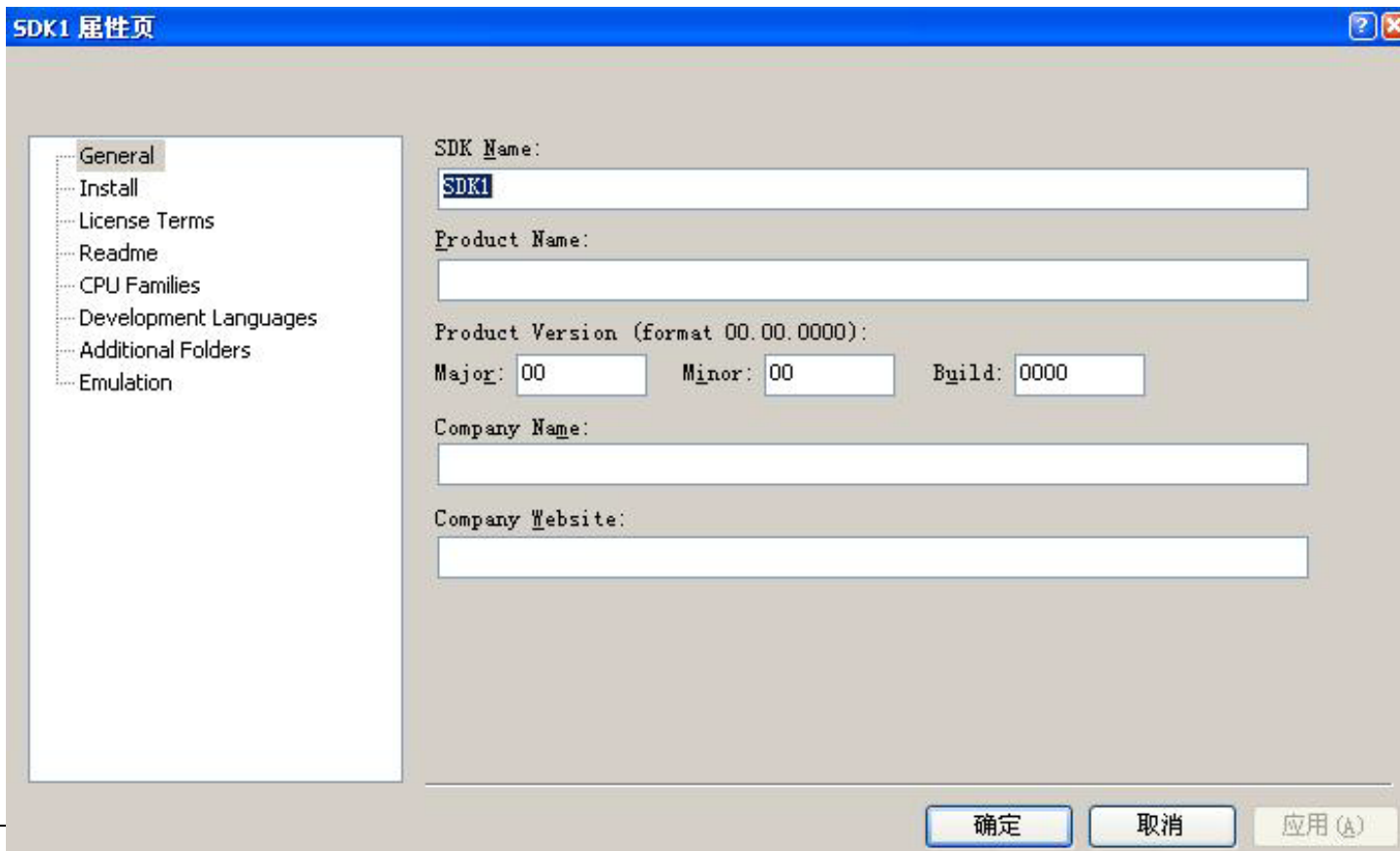
导出SDK

- ▶ PB可以基于你生成的OS，创建一个新的SDK。
- ▶ 一般，SDK包含以下内容：
 - ▶ 头文件
 - ▶ 库文件
 - ▶ 连接支持文件（支持开发机与目标机间的通信）
 - ▶ 平台管理器
 - ▶ 运行时文件
 - ▶ 文档
 - ▶ 模拟器（可选）

导出SDK

▶ 方法：

选择“项目”下的“Add New SDK”，为预创建的SDK输入必要的信息，然后点击“确定”；



SDK1 属性页

General

Install

License Terms

Readme

CPU Families

Development Languages

Additional Folders

Emulation

SDK Name: SDK1

Product Name:

Product Version (format 00.00.0000):
Major: 00 Minor: 00 Build: 0000

Company Name:

Company Website:

确定 取消 应用 (A)

导出SDK

▶ 方法：(续)

在项目解决方案资源管理器中展开“SDK”，选择新建的SDK右键单击“Build”；

成功生成过后，在目标目录下，有相应SDK的安装包。

创建子项目

▶ 可在PB下，在你设计的OS下，创建项目（驱动或应用等）。

▶ 方法：

在资源管理器窗口中，定位到Subprojects结点。

鼠标右键单击Subprojects结点，然后选择Add New Subproject，Windows Embedded CE Subproject 向导将会出现，根据向导，创建子项目。

▶ 配置子项目中与操作系统映像相关的信息

接下来配置子项目的设置，这样可以简单的调试它而无需重新构建整个操作系统映像。这是一个很好的节省开发时间的方法。

方法见下页

创建子项目

▶ 方法：

- ▶ 选择项目 Properties.
- ▶ 展开 Configuration Properties 结点，然后选择 Subproject Image Settings.
- ▶ 双击 Project settings in run-time image 列表中的子项目。Edit Run-Time Image Settings 对话框会出现。
- ▶ 选择 Exclude from image 与 Always build and link as debug 复选框，然后单击 OK.

- ▶ 从 Visual Studio 菜单中选择 Build | Targeted Build Settings
- ▶ 确保 Make Run-Time Image After Building 项没有被选中。

运行子项目

- ▶ 选择菜单 Target | Run Programs...
- ▶ 从 Available Programs 列表中选择要运行的程序, 然后点击 Run。

构建自定义的WinCE

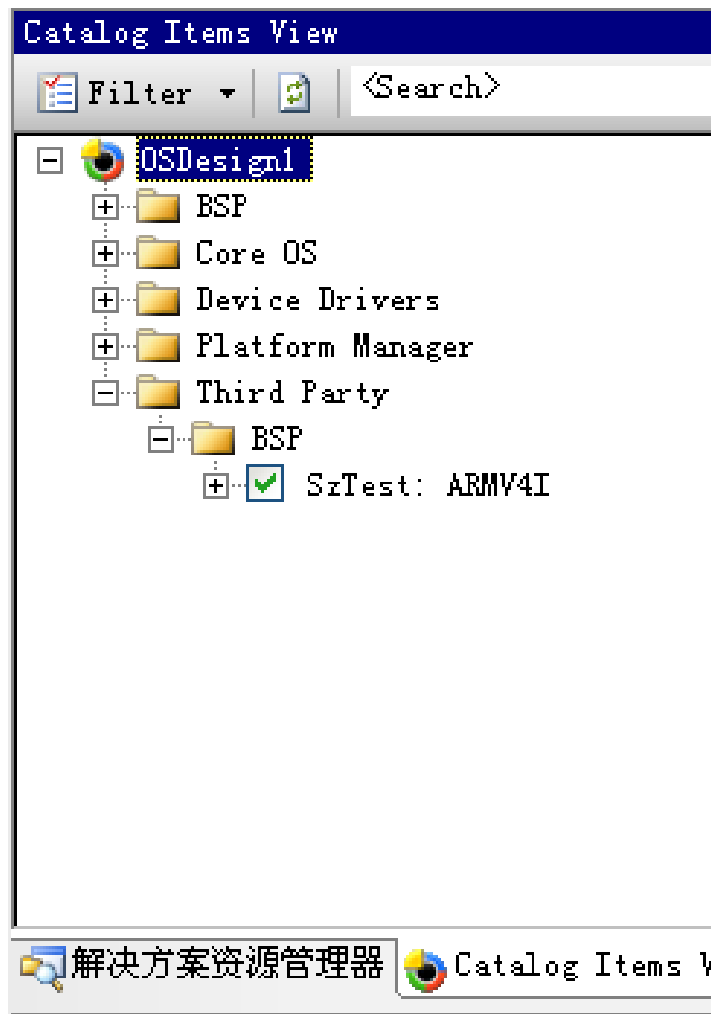
- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ 配置
- ▶ 其他

目录项(Catalog Item)

- ▶ 向导只是为我们的操作系统映像选择了最基本的组件，接下来可通过特征 (Feature) 或叫目录项 (Catalog Item) 进一步改变向导中的组件。
- ▶ WinCE中，一个模块 (逻辑群) 就叫一个特征 (Feature) 或目录项 (Catalog Item)。如下图：

目录项(Catalog Item)

- ▶ 这些Feature被分为：BSPs、内核OS、设备驱动、平台管理器和第三方特征5大类。
- ▶ 目录中的项目包含了设计CE平台所需用到的全部特性。
- ▶ 特性不是孤立存在的，某些特性可能需要和另外一些特性的支持。当开发者为定制操作系统选择了某些特性时，其相关的另外一些特性会自动被加入到开发者定制的操作系統，Platform Builder会自动解析特性或组件之间的相关性。



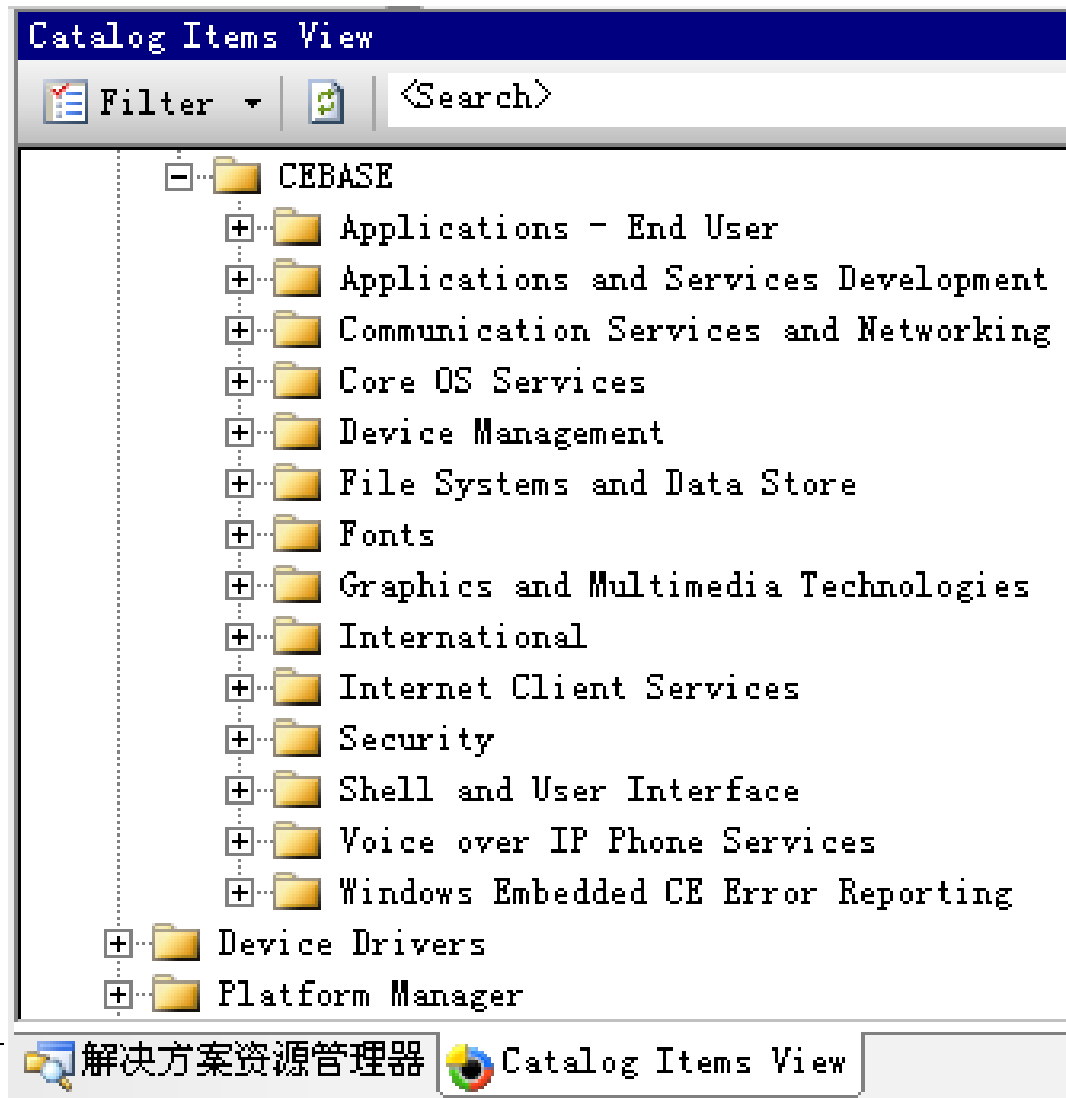
目录项(Catalog Item)

▶ BSPs

列出了所有platform Builder中已安装的BSP，供选用

Core OS 特性

- ▶ Core OS按照不同的功能特性分为十四类：



Core OS 特性

▶ Applications - End User

发布给最终用户的应用程序。包括ActiveSync的CE端、远程桌面连接程序、MSN、CAB文件安装和卸载工具、写字板、游戏等；

▶ Applications and Services Development

CE中用来开发应用程序和服务的库和系统功能。包括.NET Compact Framework、MFC、ATL、COM (DCOM)、MQ、C运行库等；

▶ Communication Services and Networking

网络相关的特性。包括WAN、LAN、PAN上的一些协议的实现等；

▶ Core OS Services

OS的核心特征。包括串、并口、USB口、支持内存映射、电源管理等；

Core OS 特性

▶ Device Management

设备管理。包括snmp和设备管理客户端

▶ File Systems and Data Store

文件系统和数据存储选项。包括注册表、存储管理、文件管理等；

▶ Fonts

各种可选字体

▶ Graphics and Multimedia Technologies

图形和多媒体支持。包括音频、视频等各种多媒体；

▶ International

全球化支持。包括各种语言的输入法和多语言用户界面等；

Core OS 特性

- ▶ Internet Client Services

因特网服务。包括web 浏览器及组件等；

- ▶ Security

安全性支持

- ▶ Shell and User Interface

图形界面组件。各种风格的菜单及XP风格的皮肤等；

- ▶ Voice over IP Phone Services

VOIP相关组件。包括RTC协议的实现等

- ▶ Windows Embedded CE Error Reporting

WinCE的错误报告组件

目录项(Catalog Item)

- ▶ Device Drivers

各种设备的驱动程序



- ▶ Platform Manager

平台管理器，负责Win CE设备与PC机的连接

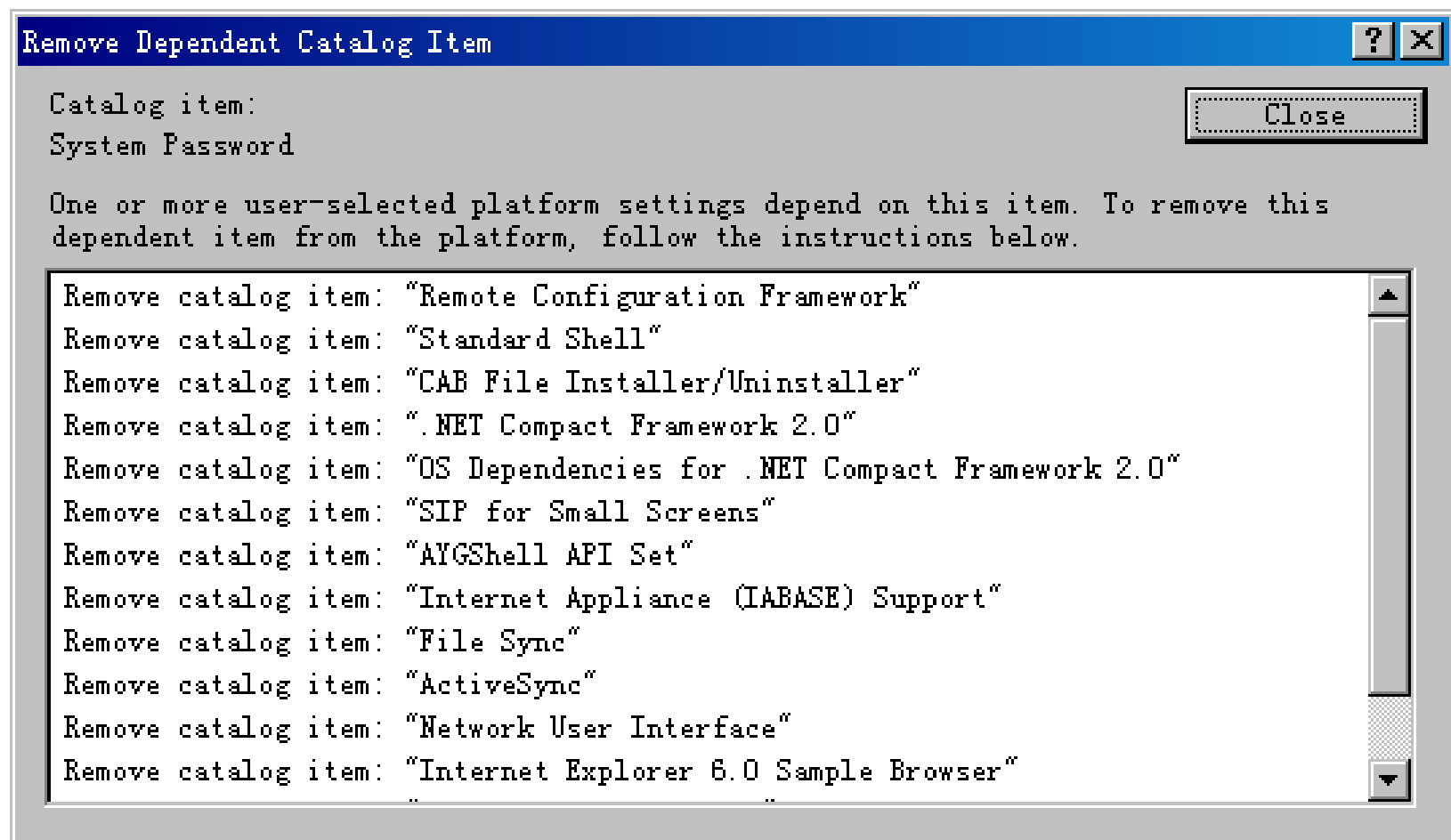
- ▶ Third Party

第三方组件

目录项(Catalog Item)

- ▶ 查看OS Design的组件
 - ▶ 在Catalog Item View中观察组件前面选择框的图标。
 - ▶  符号表示该组件是被用户主动选择进操作系统映像中的。
 - ▶  符号表示该组件是被其它组件关联进操作系统映像中的。
 - ▶ 鼠标右键单击关联组件，选择 “Reasons for Inclusion of Item”，可出现 “Remove Dependent Catalog Item” 对话框，显示该组件是由于添加了哪些组件而被关联进来的。如下页图是关联 “System Password” 组件的组件。

显示组件的依赖关系



目录项(Catalog Item)

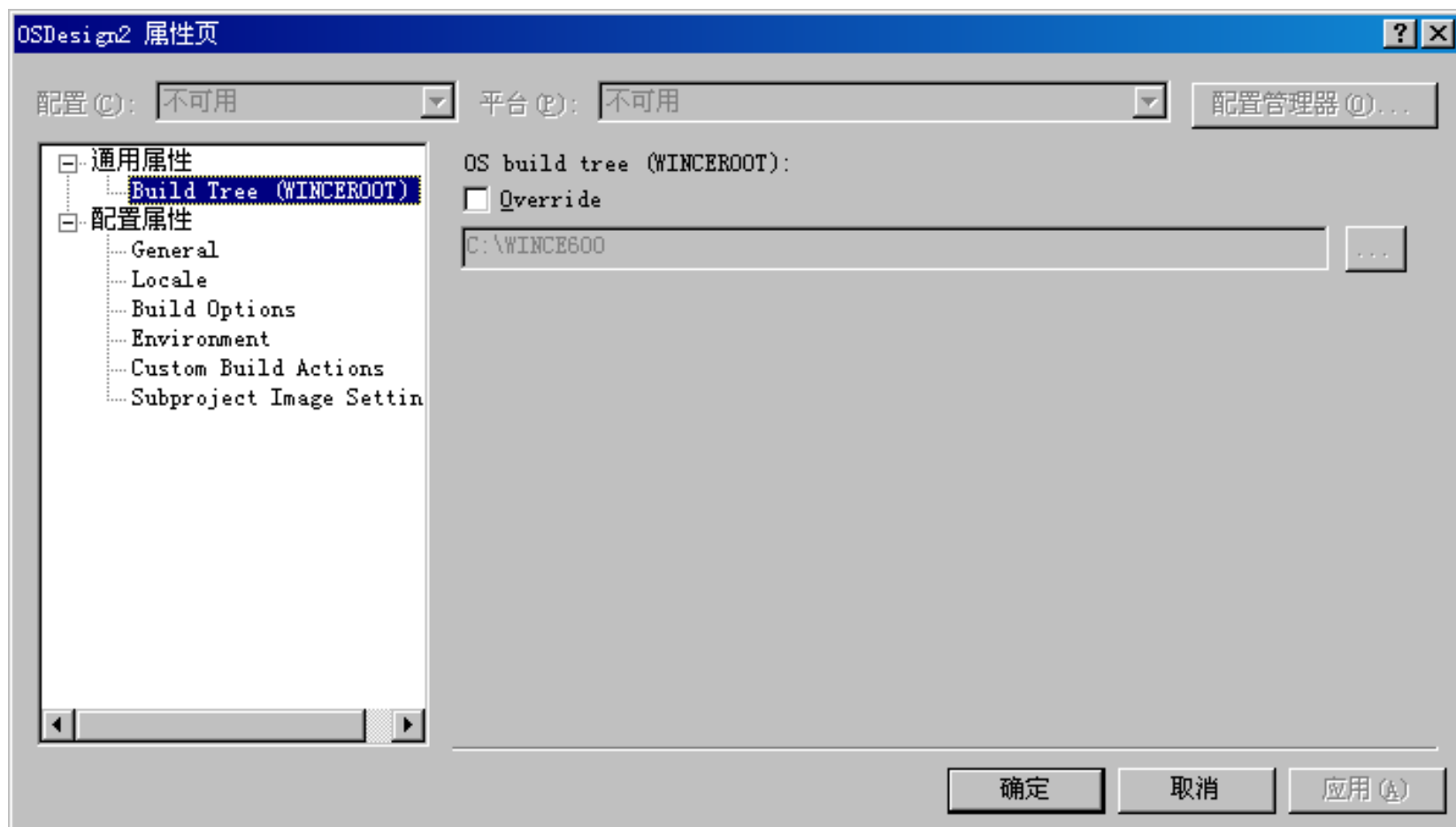
▶ 查看OS Design的组件（续）

▶ 可通过“Display In Solution View” 查看Windows Embedded CE 6.0开放的源代码。

如：Core OS | CEBASE | Applications - End User | Active Sync | Active Sync system cpl, 单击右键选择，视图会自动变到 Solution Explorer 选项卡，并且显示包含Active Sync 组件的子项目。

平台配置

浏览和修改关于整个平台的选项



平台配置

▶ 通用属性—Build Tree(WINCEROOT)

This category enables you to override the default OS build tree (WINCEROOT).

The WINCEROOT corresponds to the Windows Embedded CE 6.0 installation directory, typically C:\WINCE600.

▶ OS build tree (WINCEROOT)

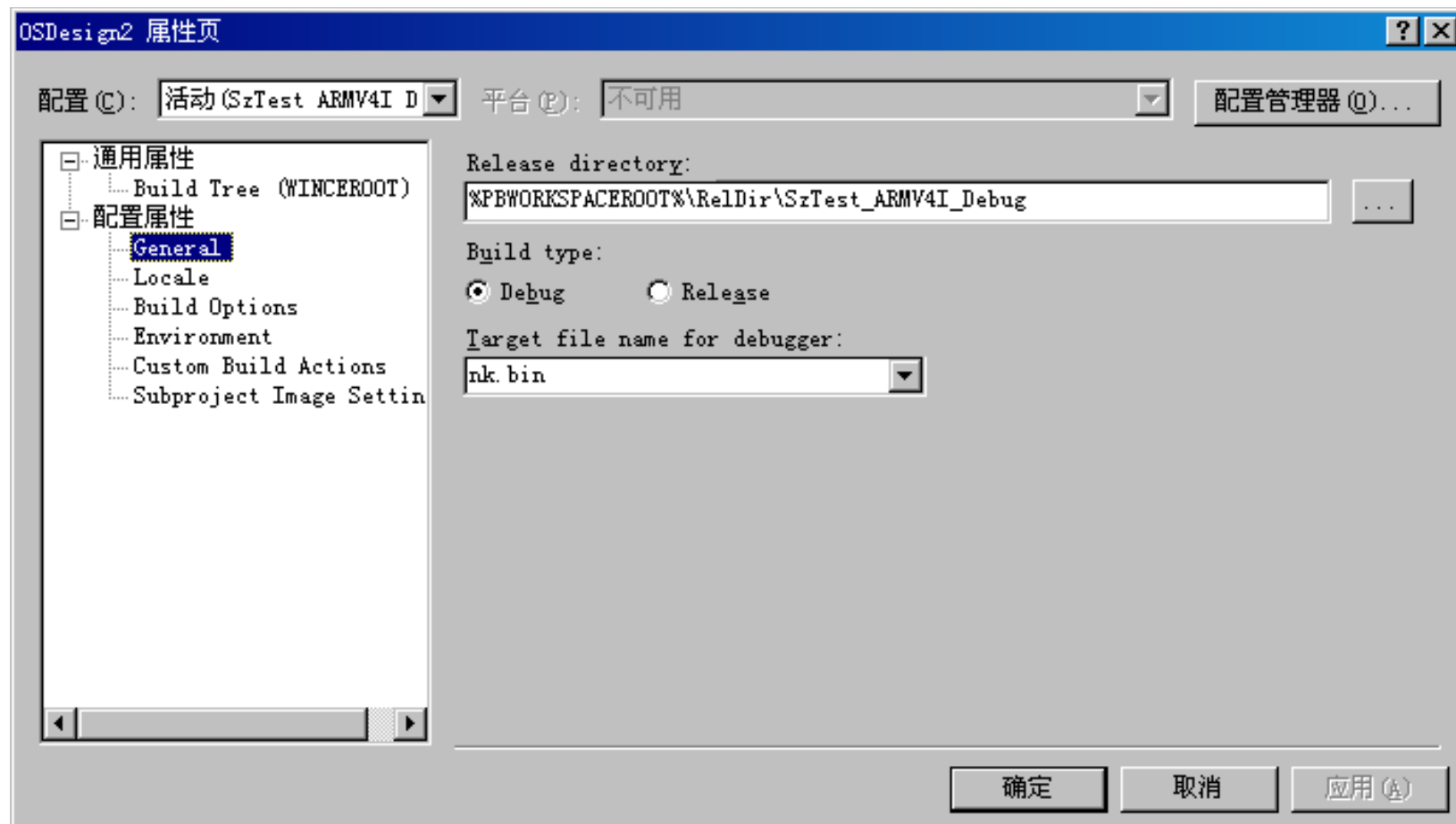
Displays the path to the OS build tree, typically C:\WINCE600.

▶ Override

Select this check box to override the default OS build tree and set the WINCEROOT of your choice.

平台配置

配置属性--General



平台配置

▶ 配置属性—General

This category enables you to configure some general configuration properties for the active OS design.

▶ Release directory

Specifies the release directory for the OS design, where the final run-time image is placed after the build process has completed.

▶ Build type

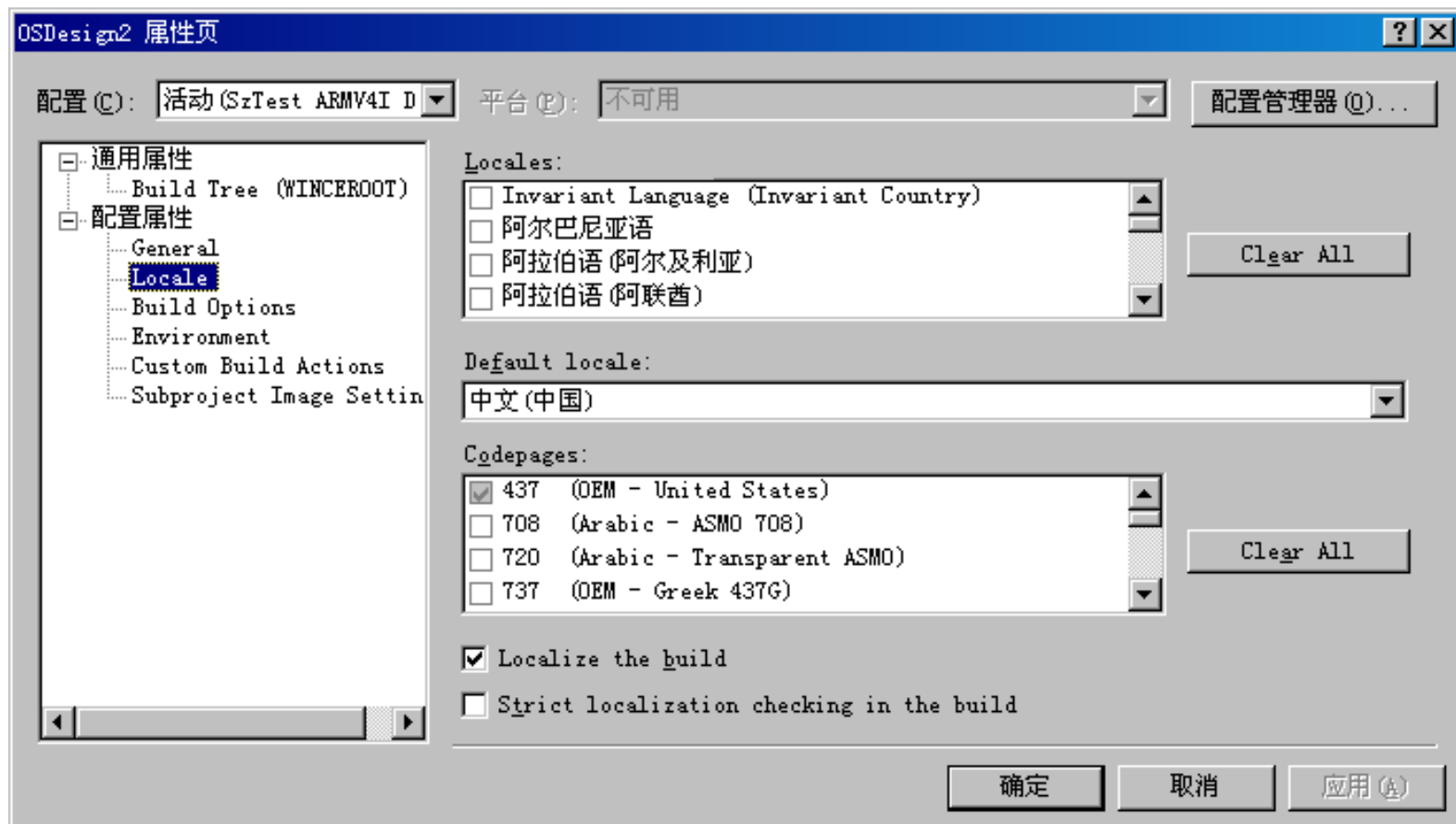
Specifies whether the run-time image that Platform Builder creates is a debug or release run-time image.

▶ Target file name for debugger

Specifies the file name that the debugger will use for the target file. By default, this is Nk.bin.

平台配置

► 配置属性--Locale



平台配置

▶ 配置属性—Locale

This options to configure your OS design to create a locale-specific run-time image.

▶ **Locales**

Displays the names of the valid locales that you can choose to localize your OS design in.

▶ **Default locale**

Displays the languages currently selected from the **Locales** list.

▶ **Codepages**

Displays the list of supported ANSI and OEM code pages. By default, selected locales are associated with the default ANSI and OEM code pages.

▶ **Localize the build**

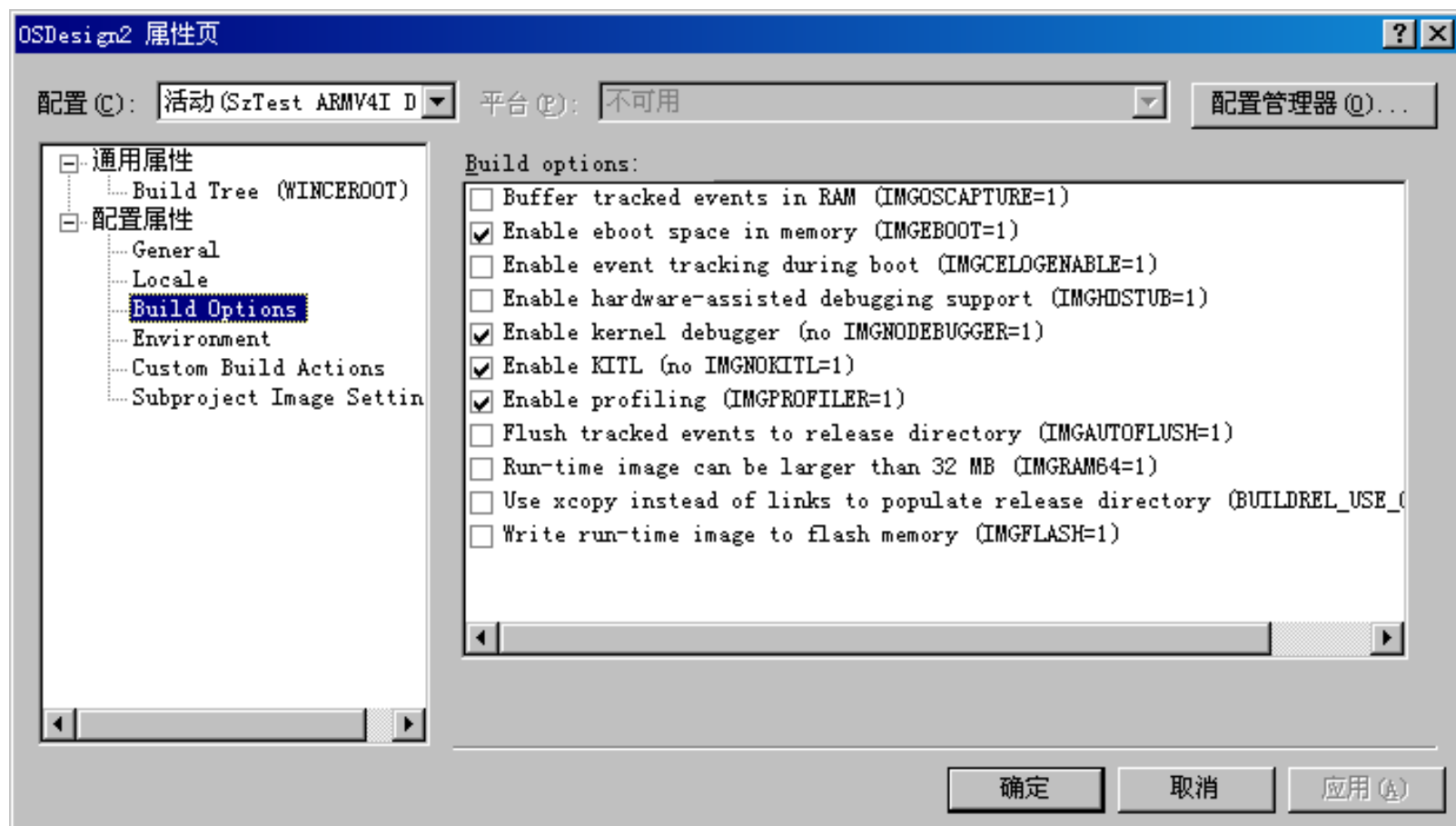
Select to ensure localization of the run-time image.

▶ **Strict localization checking in the build**

Select to choose strict localization, which requires that all localizable files be available during the **Make Run-Time Image** phase. If this option is selected and a localized module is not found, the build process fails to generate a run-time image.

平台配置

► 配置属性 -- Build Options



平台配置

▶ 配置属性 -- Build Options

- ▶ Buffer tracked events in RAM (IMGOSCAPTURE=1)

使用OSCapture.exe 来记录日志，日志放在RAM中。

- ▶ Enable Eboot Space in Memory (IMGEBOOT=1)

为Eboot在内存中预留Config.bib文件指定的空间，这样Eboot可在此空间写入信息，操作系统可以读取Eboot存储的数据。

Eboot: BOOTLOADER通过Ethernet下载操作系统映像。

- ▶ Enable Event Tracking during Boot (IMGCELOGENABLE=1)

选该项可开启事件跟踪子系统(event-tracking subsystem)。

- ▶ Enable hardware-assisted debugging support (IMGHDSTUB=1)

开启硬件辅助调试。

平台配置

▶ 配置属性 -- Build Options (续)

- ▶ Enable Kernel Debugger (no IMGNODEBUGGER=1)

启用内核调试器。

- ▶ Enable KITL (no IMGNOKITL=1)

要建立宿主机和目标机间的通讯就必须选择该选项，取消该项也会同时取消被选定的“Enable CE Target Control Support”选项。

KITL：内核独立传输层，负责在目标机和开发机间传递各种信息。

- ▶ Enable Profiling (IMGPROFILER=1)

可将Windows CE的有关内核信息以日志的形式装入平台镜像中，以便分析、优化

- ▶ Flush Events to Release Directory (IMGAUTOFLUSH=1)

选择该选项将事件放入release目录，同时开启事件跟踪

平台配置

▶ 配置属性 -- Build Options(续)

- ▶ Run-time Image Can be Larger than 32 MB (IMGRAM64=1)

Select to enable a run-time image that requires 32 MB or more of RAM in order to successfully download to the target device after the run-time image is built.

- ▶ Use XCOPY instead of links to populate release directory (BUILDREL_USE_COPY=1)

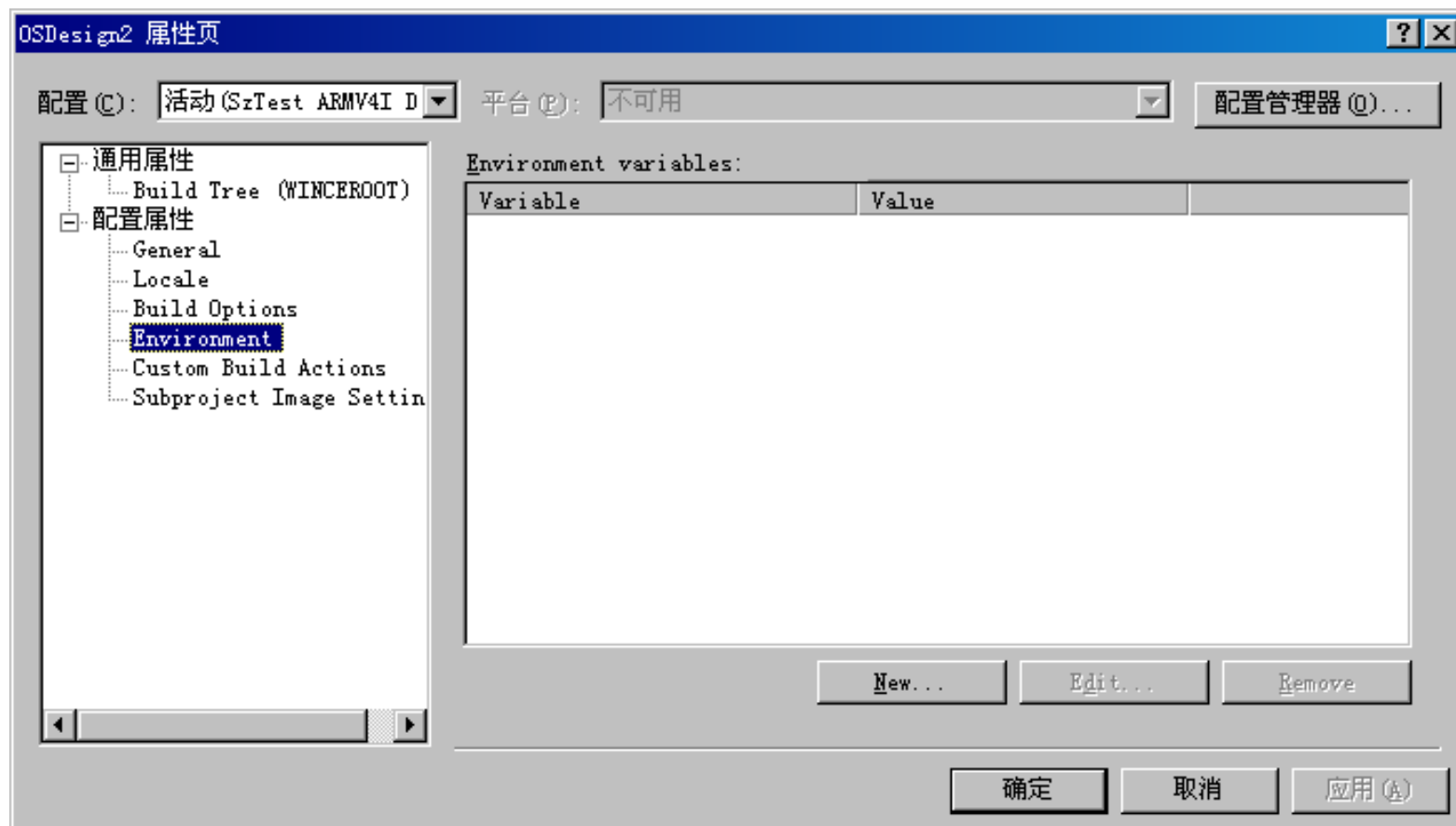
Select to directly copy files to the release directory.

- ▶ Write Run-time Image to Flash Memory (IMGFLASH=1)

Select to allow the run-time image to be written to flash memory after download.

平台配置

► 配置属性 -- Environment



平台配置

▶ 配置属性 -- Environment

This category enables you to enter the environment variables that you want to set in the active OS design.

▶ **Variable**

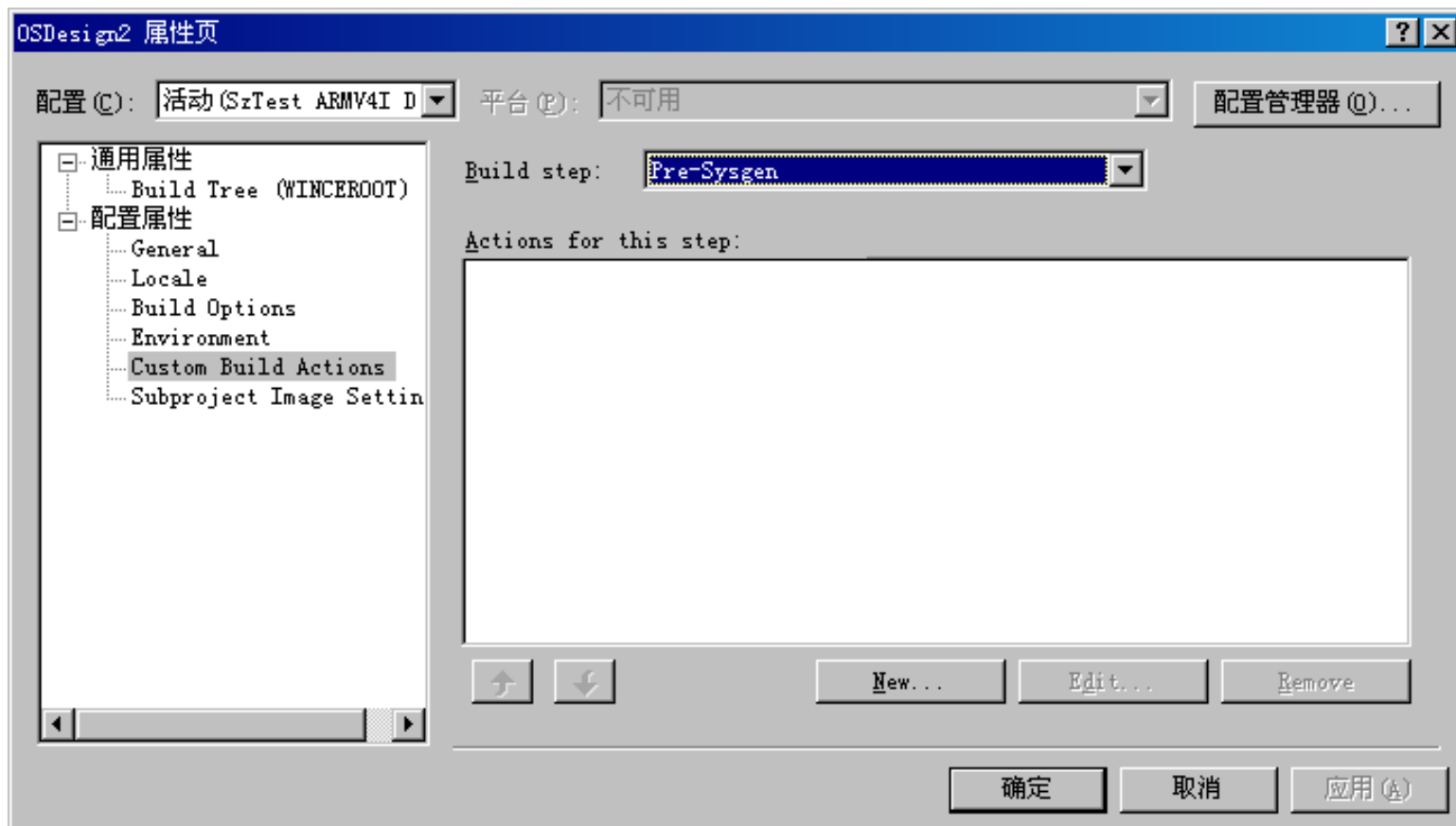
Specifies the name of the environment variable that you want to set.

▶ **Value**

Specifies the value of the associated environment variable.

平台配置

► 配置属性 -- Custom Build Actions



平台配置

▶ 配置属性 -- Custom Build Actions

This category enables you to customize the build process by specifying the custom build actions that you want the build system to perform during the build process.

▶ Build step

Specifies the point in the build process when a custom build action will be performed. The available steps are:

Pre-Sysgen, Post-Sysgen, Pre-Make Image, Post-Make Image

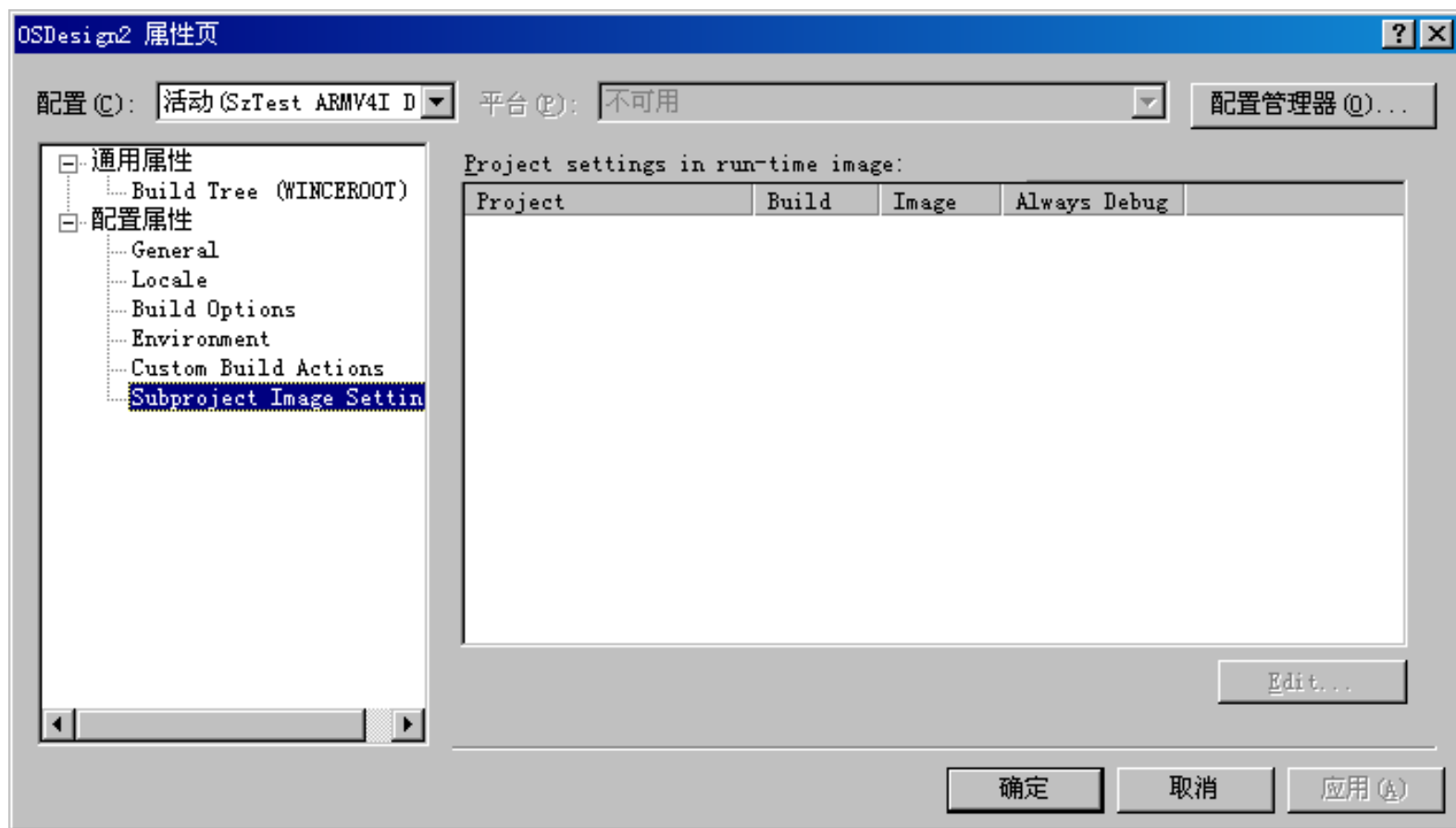
▶ Actions for this step

Enter the custom build actions that you want to perform. You can also perform the following tasks with existing custom build actions.

- ▶ Edit an existing custom build action
- ▶ Remove an existing custom build action
- ▶ Rearrange the order in which custom build actions are performed

平台配置

► 配置属性 -- Subproject Image Settings



平台配置

▶ 配置属性 -- Subproject Image Settings

This category enables you to determine the settings for the active subprojects in the currently active OS design.

▶ **Project settings in run-time image**

Displays the currently loaded subprojects in the active OS design, and their associated settings.

▶ **Project**

Displays the name of the subproject.

▶ **Build**

Specifies whether the subproject is excluded from the build process. By default, this is not selected.

▶ **Image**

Specifies whether the subproject is excluded from the final run-time image. By default, this is not selected.

▶ **Always debug**

Specifies that the subproject is always built and linked as a debug configuration.

构建自定义的WinCE

- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ 配置
- ▶ 其他

Build的几个阶段

当在IDE的Build菜单下选择 Build Platform 命令时，系统创建平台依次经历了以下4个过程：

- ▶ Sysgen 阶段（生成阶段）
- ▶ Feature Build 阶段（编译阶段）
- ▶ Release Copy阶段（Release文件夹复制阶段）
- ▶ Make Image阶段（镜像打包阶段）

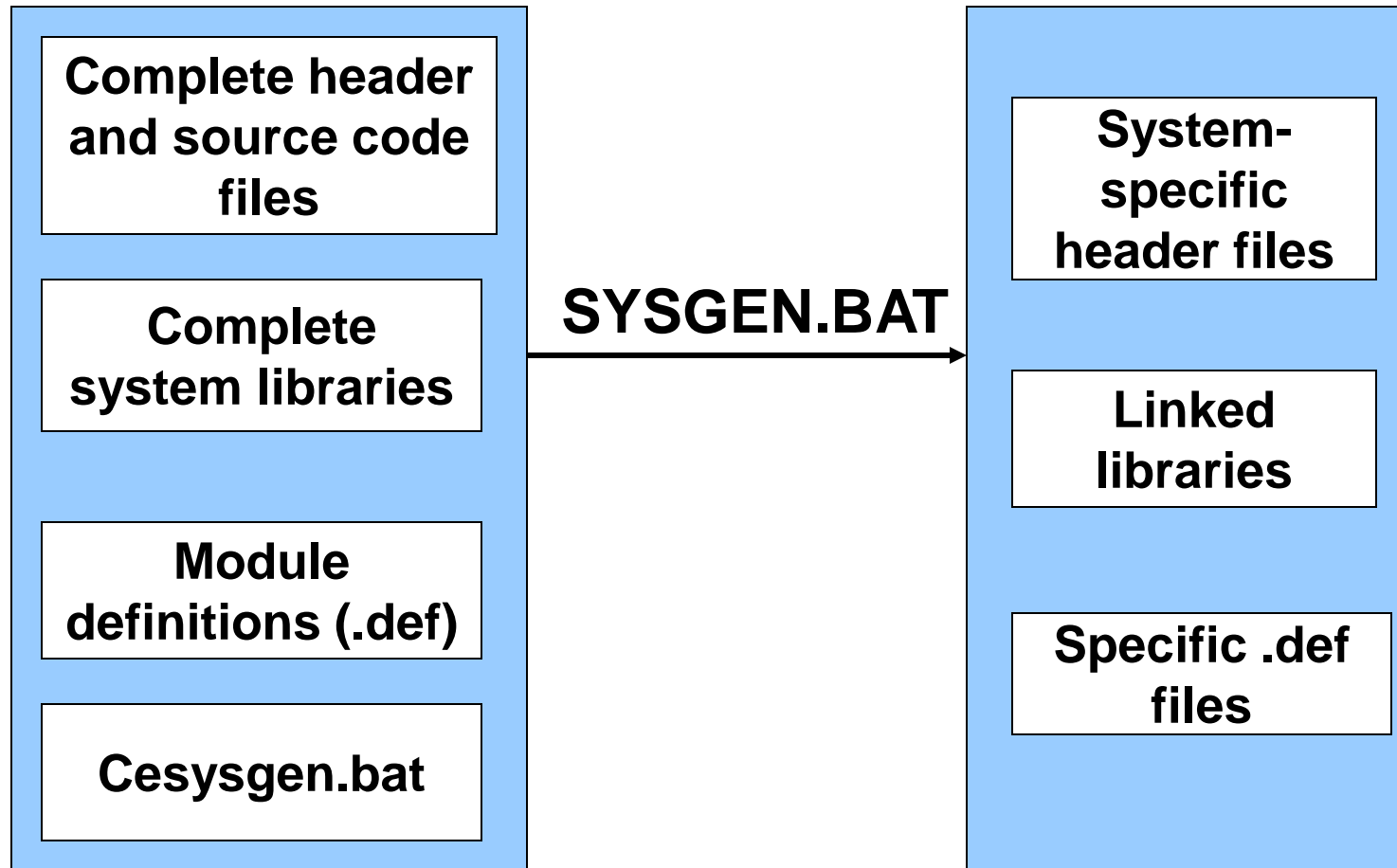
Sysgen 阶段任务

- ▶ 根据用户选择的组件，生成相应的子集，子集包括定制平台需要的头文件、库及执行文件。
- ▶ 链接选择的静态库到模块
- ▶ Make SDK（可用于再开发）
 - ▶ 根据用户选择过滤系统头文件，产生只包含为定制平台导出的函数的头文件。
 - ▶ 为系统模块产生输入库
 - ▶ 构建板级支持包 (BSP)

此阶段的输出在：

Wince600\OSDesigns\%projectname%\wince600\%cpu_type%_x86\cesysgen。

System Generation



Feature Build 阶段任务

▶ 对上步产生的源代码进行编译链接。

包括：选择的OS模块，BSP源代码，用户应用程序。

如源代码未改动，不编译。

整个build阶段都是在围绕着dirs文件和sources、Makefile文件来进行，前者决定了哪些文件夹要被编译，后者决定了哪些文件要被如何编译。

Release Copy 阶段任务

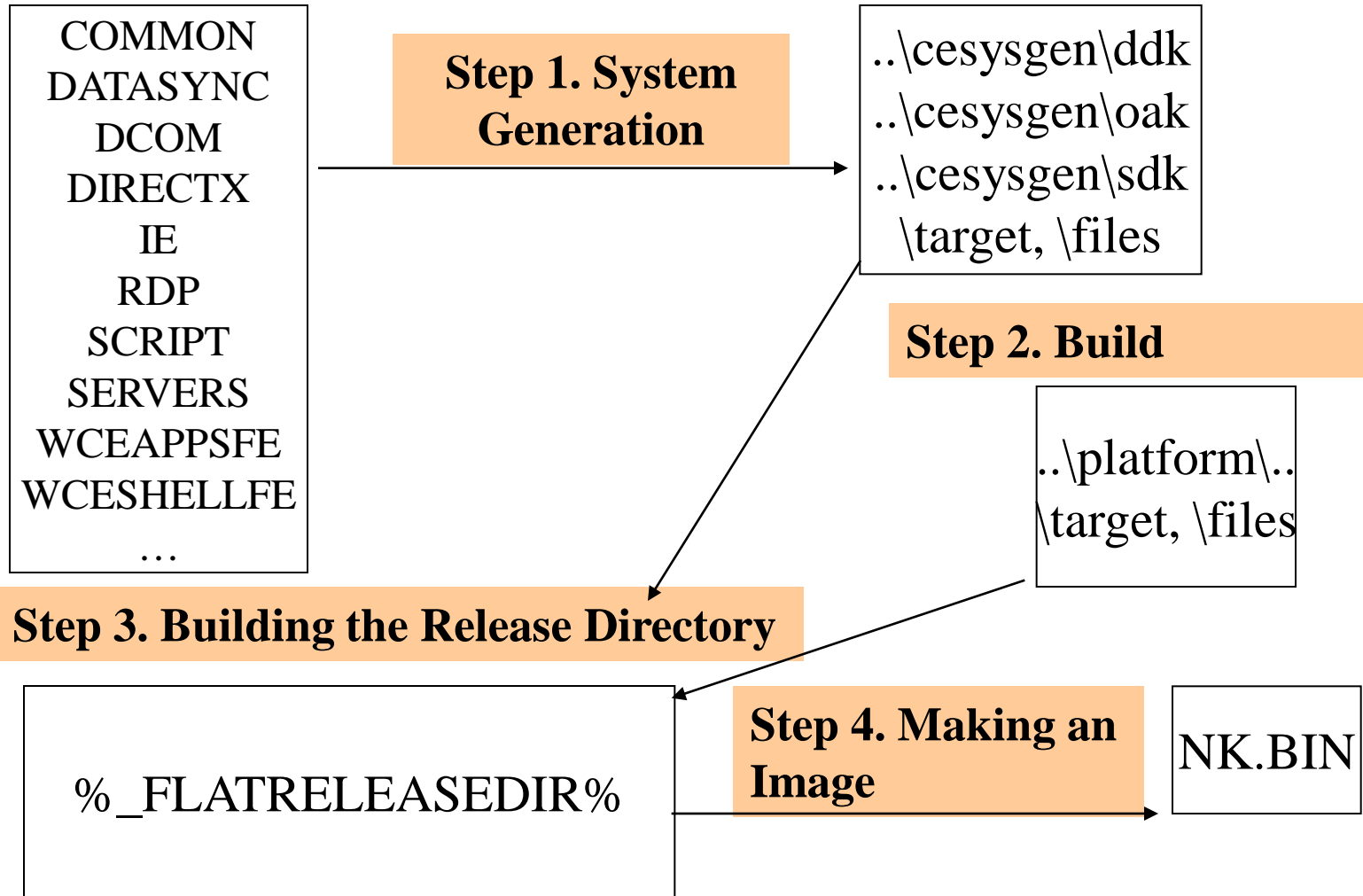
- ▶ 拷贝所有用户生成OS镜像所需的文件到release目录下。

在Sysgen阶段生成的模块和文件首先被拷贝，接着是Feature Build阶段生成的模块和文件。

Make Image 阶段任务

- ▶ Rrelease目录中的文件被打包生成二进制镜像文件： Nk.bin。
- ▶ 此阶段包括：文件合并、注册表压缩、资源文件替换和打包四个子过程
 1. 文件合并：重点整合配置文件：
 - 所有的 .bib 文件到CE.BIB
 - 所有的 .reg 文件到REGINIT.INI
 - 所有的 .dat 文件到INITOBJ.DAT
 - 所有的 .db 文件到INITDB.INI
 2. 注册表压缩：
 - 将REGINIT.INT文件压缩成二进制注册文件(DEFAULT.FDF)
 3. 资源文件替换：
 - 将EXE或DLL中的资源替换成本地语言，如简体中文。
 4. 打包生成镜像 (NK.BIN)

Build 过程



Build 结果

- ▶ nk.bin :
 - ▶ Windows CE 二进制镜像数据格式(压缩)
 - ▶ 运行时必须首先装载到内存中.
 - ▶ 由数据段和代码段组成
- ▶ nk.nb0 :
 - ▶ 二进制数据文件格式(没压缩)
 - ▶ 能够被转载到Flash中
 - ▶ 可以以 **XIP** (Execute In Place) 模式运行

当前位置

- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ **配置**
- ▶ 其他

平台配置文件

通过修改这些配置文件可以裁剪优化Windows CE。用户根据需要可以创建自己的配置文件。

Platform Builder 使用两类配置文件：

- 源码配置文件(在Build阶段处理)

Build工具根据源码配置文件去处理源代码。

- 镜像配置文件(在Make Image阶段处理)

Make Binary Image工具调用相应的工具，这些工具使用镜像配置文件来产生OS镜像。

原码配置文件

- ▶ 源码配置文件用于编译源码时使用，如驱动程序、应用程序等。源码配置文件给build工具(Build.exe)提供如下信息：
 - ▶ 需遍历的目录
 - ▶ 需编译的C和VC++文件
 - ▶ 需build的二进制文件类型
- ▶ PB在编译源码文件时会寻找当前目录下存放的源码配置文件，根据配置文件的信息来编译和链接，产生EXE、DLL、LIB文件。CE的源码文件所在的目录中都包含了相应的配置文件，这些配置文件只对当前目录或者子目录的源码有效，具体分为三种：

参考：ms-

[help://MS.VSCC.v80/MS.MSDN.v80/MS.WINCE.v50.en/wceosdev5/html/wce50sourcecodeconfigurationfiles.htm](http://MS.VSCC.v80/MS.MSDN.v80/MS.WINCE.v50.en/wceosdev5/html/wce50sourcecodeconfigurationfiles.htm)

原码配置文件

▶ DIRS 文件：

描述包含源代码的子目录。条目如下：

- ▶ DIRS：指定哪个目录下的所有源码要被编译
- ▶ DIRS_CE：只有为CE编写的源码才被编译
- ▶ OPTIONAL_DIRS：指定可选的目录（很少使用这个选项），可以只编译指定目录而不是全部编译。

如：`dirs_ce= \
ceddk \
regenum \
pcibus \
"`

“\
”是继行符，等价

`dirs_ce= ceddk regenum pcibus`

原码配置文件

- ▶ SOURCES文件：包含build阶段所需的宏变量。Build.exe用这些宏变量来确定如何编译和链接源代码。常用的有：
 - ▶ TARGETNAME：指定编译链接产生的主文件名
 - ▶ TARGETTYPE：指定编译链接产生的文件的类型（决定了扩展名）。文件共分三种：.lib（LIBRARY）和.dll（DYNLINK）和.exe（PROGRAM）。
 - ▶ TARGETLIBS：指定.lib链接文件。动态链接。函数的地址被链接过来，系统执行时会加载该库。
 - ▶ SOURCELIBS：指定链接的lib文件。静态链接，函数的实体被链接过来，生成一份拷贝。
 - ▶ SOURCES：源码文件。包含扩展名为*.c或*.h或*.cpp的文件。
 - ▶ EXEENTRY：.exe文件的执行代码入口点。
 - ▶ INCLUDES：指定额外的要搜索的头文件的路径。
 - ▶ sources.cmn文件是通用的SOURCES文件。在这个文件中可以指定作用于所有源码文件的配置选项。
- ▶ 在同一目录下，DIRS文件与SOURCES文件不能同时存在。

原码配置文件

- ▶ **MAKEFILE文件：**包含编译和链接源码时所需的变量。
 - ▶ SOURCES文件会被MAKEFILE和nmake使用，来编译项目源代码；
 - ▶ 当Build工具定位一个源文件时，它调用Nmake工具，做下列之一：
 - 编译制定的C或C++源文件
 - 根据包含在makefile中的链接规则链接目标模块
- ▶ 整个编译和链接过程：build.exe通过MAKEFILE文件收集编译和链接需要的数据（源码文件、链接文件、编译和链接选项）产生一系列的内部环境变量，然后调用nmake.exe，nmake.exe根据内部环境变量执行编译、链接，最后产生最终文件（*.lib *.exe *.dll）；
- ▶ MAKEFILE中对所有项目都通用的部分在makefile.def中，makefile.def超过3000行。一般makefile仅包含makefile.def。
- ▶ 如目录包含SOURCES文件，那么一般还会有MAKEFILE文件。

原码配置文件

▶ 如

假设目录/文件布局如下:

```
hello/ --+-- dirs
      |
      +-- code/ --+-- hello.c
                  |
                  +-- sources
                    |
                    +-- makefile
```

▶ dirs 文件

DIRS = code

原码配置文件

- ▶ 编写sources文件

```
TARGETNAME = hello
```

```
TARGETTYPE = PROGRAM
```

```
SOURCES=hello.c
```

- ▶ 编写makefile文件

```
!INCLUDE $(_MAKEENVROOT)\makefile.def
```

`_MAKEENVROOT`是一个环境变量，指向makefile.def所在目录，一般是%_PublicRoot%\COMMON\OAK\MISC下。

镜像配置文件

- ▶ 镜像配置文件用于在制作CE镜像文件时使用。CE的镜像文件扩展名为.bin。
- ▶ 镜像配置文件包括4种文件类型：二进制映像生成文件.BIB、注册表文件.REG、目录和文件分配表文件.DAT、数据库文件.DB。通过修改这些配置文件可以裁剪优化Windows CE。用户根据需要可以创建自己的配置文件，PB提供的主要配置文件如表所列。

ms-

[help://MS.VSCC.v80/MS.MSDN.v80/MS.WINCE.v50.en/wceosdev5/html/wce50conimageconfigurationfiles.htm](http://MS.VSCC.v80/MS.MSDN.v80/MS.WINCE.v50.en/wceosdev5/html/wce50conimageconfigurationfiles.htm)

镜像配置文件

.BIB文件	.REG文件	.DAT文件	.DB文件
Common.bib	Common.reg	Common.dat	Common.db
Dcom.bib	Dcom.reg	Servers.dat	
Msmq.bib	Msmq.reg		
Platform.bib	Platform.reg	Platform.dat	Platform.db
Project.bib	Project.reg	Project.dat	
Wceshellfe.bib	Wceshellfe.reg	Wceshellfe.dat	Wceshellfe.db
Wceappsfe.bib	Wceappsfe.reg	Wceappsfe.dat	Wceappsfe.db
Config.bib	Script reg		



配置文件：.REG文件

- ▶ 与桌面Windows一样，Windows CE也使用注册表（Registry）来保存应用程序、驱动程序和用户的设定以及其他一些配置信息。
- ▶ 定义默认的注册设置。
 - ▶ COMMON.REG, IE.REG, WCEAPPS.REG, WCESHELL.REG: 为Windows CE模块定义注册设置；
 - ▶ PLATFORM.REG: 与平台相关的注册设置，如驱动入口等；
 - ▶ PROJECT.REG: 定义与你的工程相关的注册设置；

如：

```
[HKEY_LOCAL_MACHINE\init]
```

```
“Launch10” = “shell.exe”
```

```
“Launch20” = “device.exe”
```

上述注册表入口设置规定内核在启动时必须自动运行shell.EXE和device.exe模块。

配置文件: .DAT文件

▶ 定义镜像的文件夹结构

冷启动时, 如何初始化RAM文件结构

- ▶ COMMON.DAT, IE.DAT, WCEAPPS.DAT, WCESHELL.DAT
- ▶ PLATFORM.DAT
- ▶ PROJECT.DAT

注意, Windows CE不支持像Windows XX一样的盘符驱动器, 而且没有当前目录的概念。

如:

```
Root:-Directory( "Program Files" )
```

```
Directory( "\Program Files" ):-Directory( "My Projects" )
```

```
Root:-Directory( "My Documents" )
```

```
Directory( "\My Documents" ):-File( "MyFile.doc", "\  
Windows\Myfile.doc" )
```

生成2个根目录: Program Files和MyDocuments。Program Files有1个子目录: My Projects。文件MyFile.DOC位于My Document目录中。

配置文件：.DB 文件

- ▶ 定义默认的镜像数据库

- ▶ COMMON.DB, WCEAPPS.DB, WCESHELL.DB

- 这些 .DB 文件是与 Common, Wceapps, and Wceshell 模块相关的数据库

- ▶ PLATFORM.DB

- 该文件包含特定平台的默认数据库

- ▶ PROJECT.DB

- 该文件特定工程的默认数据库

配置文件: .BIB 文件

- ▶ 表明应该被包含在镜像中的模块、组件和加载位置
- ▶ 包含4个部分(MEMORY、FILES、MODULES和CONFIG)的文本文件:

配置文件: .BIB 文件

▶ MEMORY 部分

定义有效的物理内存块。

格式: 名称 首地址 大小 内存类型

名称: 内存区的唯一名

首地址: 内存区的首地址 (十六进制表示)

大小: 内存区的大小 (十六进制表示)

内存类型: 分成多种类型:

- ▶RAM: 指定此段是RAM, 可分配给正在运行的进程;
- ▶RAMIMG: 运行时映像所在段, 可为RAM, 通常为NOR Flash;
- ▶RESERVED: 为保留的内存空间, 内核不能使用这部分内存, 但是应用程序可以访问。
- ▶FIXUPVAR: 可改变内核模块中全局变量值
- ▶NANDIMAGE: 闪存类型

配置文件: .BIB 文件

如:

```
NK 80200000 00600000 RAMIMAGE
```

```
RAM 80800000 00800000 RAM
```

在Windows CE系统中，NK的起始地址是80200000，它占有6MB内存空间，用来存放运行时映像区。RAM在80800000开始，占有8MB的内存空间，用于程序存储区。

配置文件: .BIB 文件

▶ FILES 部分:

为静态数据文件预留的内存。数据文件包括字体文件 (.TTF)、文本文件 (.TXT)、位图文件 (.BMP) 和声音文件 (.WAV) 等。

格式: 模块名称 路径 内存块 类型

模块名称: 在Windows CE系统中的文件名。

路径: 文件的位置。

内存块: 文件所属的内存区。内存块的定义由MEMORY部分指定;

类型: 定义文件属性。

Windows CE中的文件有以下文件属性:

S为系统文件、H为隐含文件、U为非压缩文件、R为压缩资源文件、C为压缩文件。1个文件可以同时具有几种属性。

配置文件: .BIB 文件

▶ 如:

```
Tahoma.ttf $(_FLATRELEASEDIR)\Tahoma.ttf NK SHU
```

上例说明NK内存区包含\$(_FLATRELEASEDIR)\Tahoma.ttf文件。在Windows CE系统中, tahoma.ttf的文件属性是系统隐含的非压缩文件。

配置文件: .BIB 文件

▶ MODULES 部分

指定被装载到内存中的目标模块和组件，包括所有的可执行文件.EXE和动态链接库.DLL。

格式： 模块名称 路径 内存块 类型

模块名称一般为模块的真实名称；

路径为当前文件所处的位置（路径中指定的文件名和前面模块名称最好一致）；

内存块是指这个模块将被存放到哪个内存块中。内存块的定义由MEMORY部分指定；

类型指定这个模块将被存放的属性，具体类型如下：

配置文件: .BIB 文件

- S: 系统文件 H: 隐藏文件
- R: 只压缩模块的资源部分 (默认模块是不压缩的)
- C: 压缩模块所有部分 D: 禁止调试 N: 标志模块是非信任的
- P: 忽略CPU类型
- K: 指定romimage.exe修正模块 (仅用于调试或者内核跟踪)
- X: 指定romimage.exe对此模块验证签名
- M: 运行时加载整个模块, 而不是按需分页
- L: 不分离DLL在进程地址空间和Slot 1

如:

```
device.dll  $(_FLATRELEASEDIR)\device.dll  NK  SH
```

device模块加载到内存区。Windows CE系统中device.dll具有系统和隐含属性。

配置文件: .BIB 文件

▶ CONFIG 部分

这部分不是必需的。类似环境变量，PB预设置了一些配置变量。常用的配置及说明如下：

▶ AUTOSIZE

格式：AUTOSIZE = OFF | ON 默认值为OFF。

在config.bib中的MEMORY部分定义了有效的内存区域，其中两部分RAM、RAMIMAGE分别用于进程使用区域和保存镜像区域。如果为ON，romimage.exe在创建nk.bin时将RAM和RAMIMAGE两部分合并成一个部分，然后从最低地址开始保留RAMIMAGE大小的内存，其余都作为RAM使用。

▶ BOOTJUMP

格式：BOOTJUMP = address | NONE 默认值为NONE。

每次重新启动CE内核，默认执行的代码从RAMIMAGE的首地址开始。如果在BOOTJUMP指定一个地址（必须在RAMIMAGE范围内），那么将从指定的地址开始执行。

配置文件: .BIB 文件

▶ OUTPUT

格式: OUTPUT = path

指定romimage.exe将创建完成的内核文件nk.bin放置到的路径。一般放置到%_FLATRELEASEDIR%下。

▶ COMPRESSION

格式: COMPRESSION = OFF | ON 默认值为ON。

romimage.exe在创建内核时默认压缩所有可写部分。对于文件，默认全部压缩。对于模块(.exe、.dll)，默认压缩可写部分。模块的可写部分包括数据段，也就是在模块运行时一定加载到内存中的部分。如果模块在.bib中定义时具有C属性（表明压缩模块所有部分），那么当前这个选项就忽略了。

▶ KERNELFIXUPS

格式: KERNELFIXUPS = OFF | ON 默认值为ON。

如果为ON，romimage.exe创建内核前重定位内核到RAM的开始位置。

配置文件：.BIB 文件

▶ ROMFLAGS

格式：ROMFLAGS = Flags

设置内核选项的位掩码，多个位掩码可以组合使用。

- 0x0001 禁止按需分页：EXE和DLL默认是按需分页的。
- 0x0002 禁止全内核模式。
- 0x00000010 只信任来自ROM的模块（DLL、EXE）。默认ROM中的模块和所有文件系统的模块都是内核信任的。OEM能够在OAL层实现对所有运行模块的检查，这个标志将忽略对来自ROM保存的模块的检查。
- 0x00000020 停止刷新TLB。这个标志仅用于运行在x86CPU上的内核。TLB（Translation Look-aside Buffer），有人译成变换索引缓冲区，它的作用是在虚拟地址和物理地址之间转换。对于具有实时性的内核，这个标志应该设置。
- 0x00000040 按照/base链接选项中的地址加载DLL。这样内核将不采用重定位加载DLL。不建议采用。

配置文件: .BIB 文件

▶ ROMSIZE

格式: ROMSIZE = size

指定ROM映像（内核镜像）的大小。

要产生NK.NBx文件，你必须设置此参数。如ROMSIZE = 02000000，设置NK.NB0的大小为32M，若Image的大小超过32M，则还会产生NK.NB1，NK.NB2，...

▶ ROMSTART

格式: ROMSTART = address

指定内核镜像的首地址

▶ ROMWIDTH

格式: ROMWIDTH = width

指定数据宽度，一般为32位

▶ ROMOFFSET

格式: ROMOFFSET = address

指定偏移地址。

注: config中绝大多数【CONFIG】选项不需要修改。凡是配置文件都可以使用IF/ENDIF条件语句。

当前位置

- ▶ Windows Embedded CE操作系统设计概述
- ▶ 平台创建流程
- ▶ 克隆BSP、导出SDK及创建子项目
- ▶ 特性(Feature)和平台配置
- ▶ Build的几个阶段
- ▶ 配置
- ▶ 其他

其他

- ▶ 远程工具
- ▶ ActiveSync
- ▶ 模拟器概述

远程工具

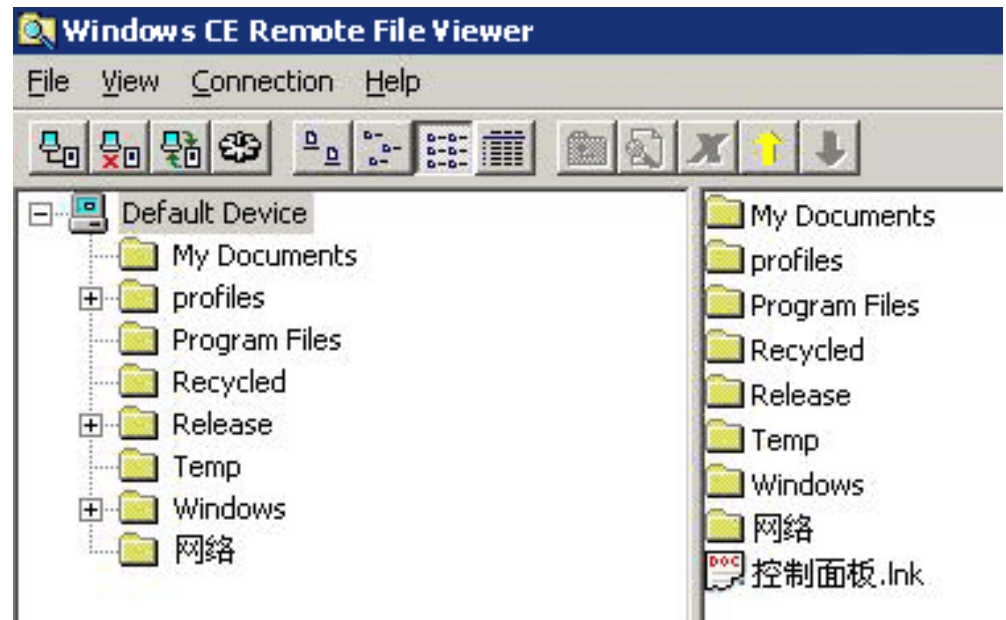
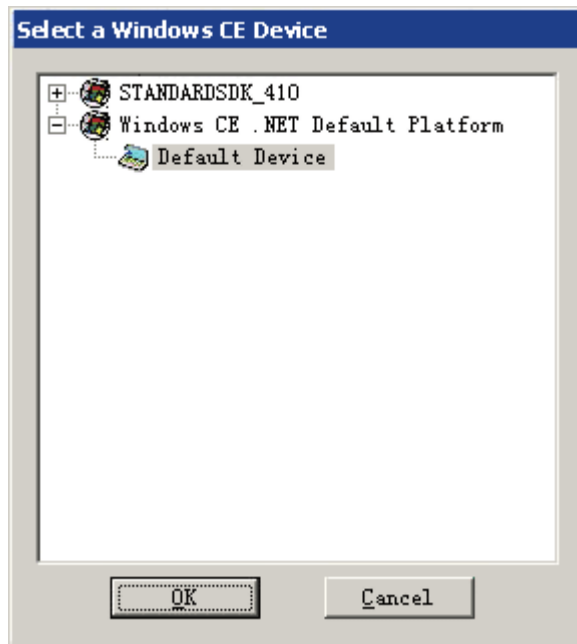
- ▶ 你可以在宿主机上远程执行位于目标机上的各种程序
- ▶ 不同的平台支持不同的远程工具
- ▶ 位于PB、EVC和VS的Tools 菜单下
- ▶ 开发者可以使用相应的接口和APIs产生新的远程工具
- ▶ 远程工具可分为3类
 - ▶ 调试：Heap Walker, Process Viewer, Spy
 - ▶ 监测：Call Profiler, Kernel Tracker, Performance Monitor
 - ▶ 信息管理：File Viewer, Registry Editor, System Information, Zoom-in

远程调试工具的使用

- ▶ 远程文件浏览程序 (Remote File Viewer)
- ▶ 远程堆查看程序 (Remote Heap Walker)
- ▶ 远程内核跟踪程序 (Remote Kernel Tracker)
- ▶ 远程性能监视程序 (Remote Performance Monitor)
- ▶ 远程进程浏览程序 (Remote Process Viewer)
- ▶ 远程注册表编辑程序 (Remote Registry Editor)
- ▶ 远程消息监视程序 (Remote Spy)
- ▶ 远程系统信息 (Remote System Information)
- ▶ 远程屏幕截图程序 (Remote Zoom-in)
- ▶ 远程调用评测程序 (Remote Call Profiler)

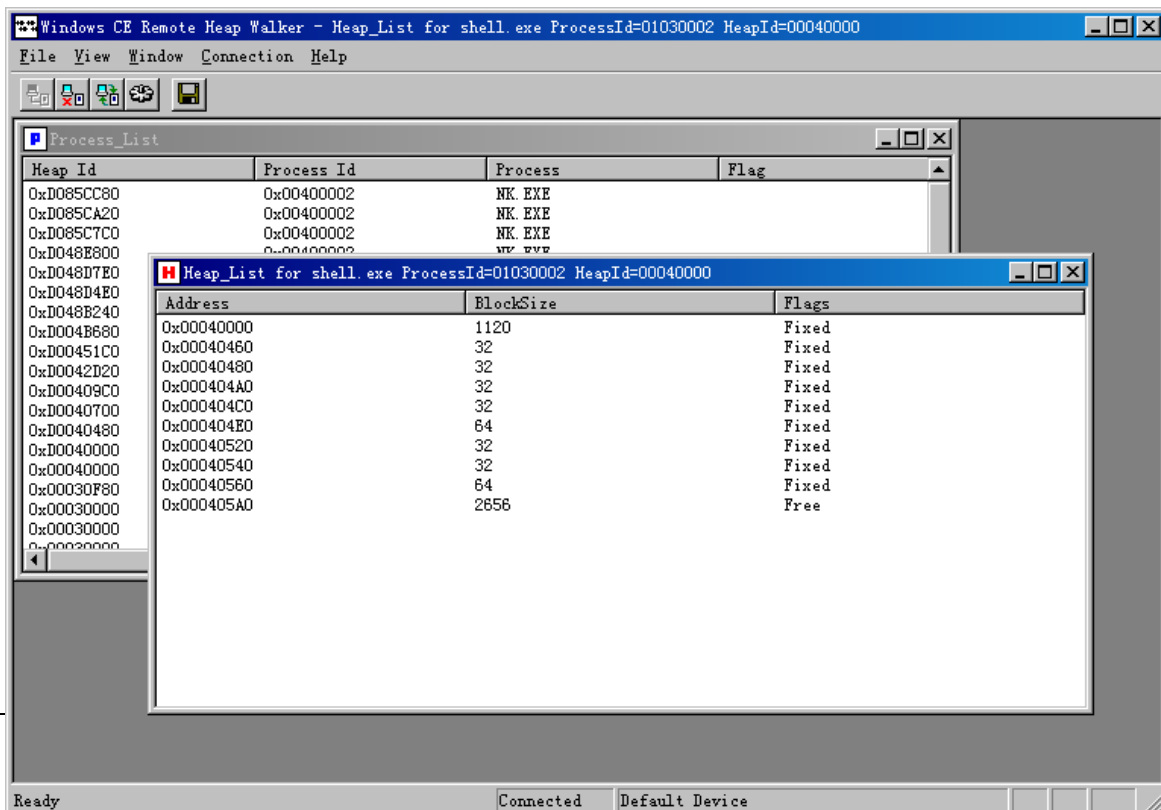
远程文件浏览程序（Remote File Viewer）

- ▶ 向实际平台导入或导出文件
- ▶ 选择访问的平台



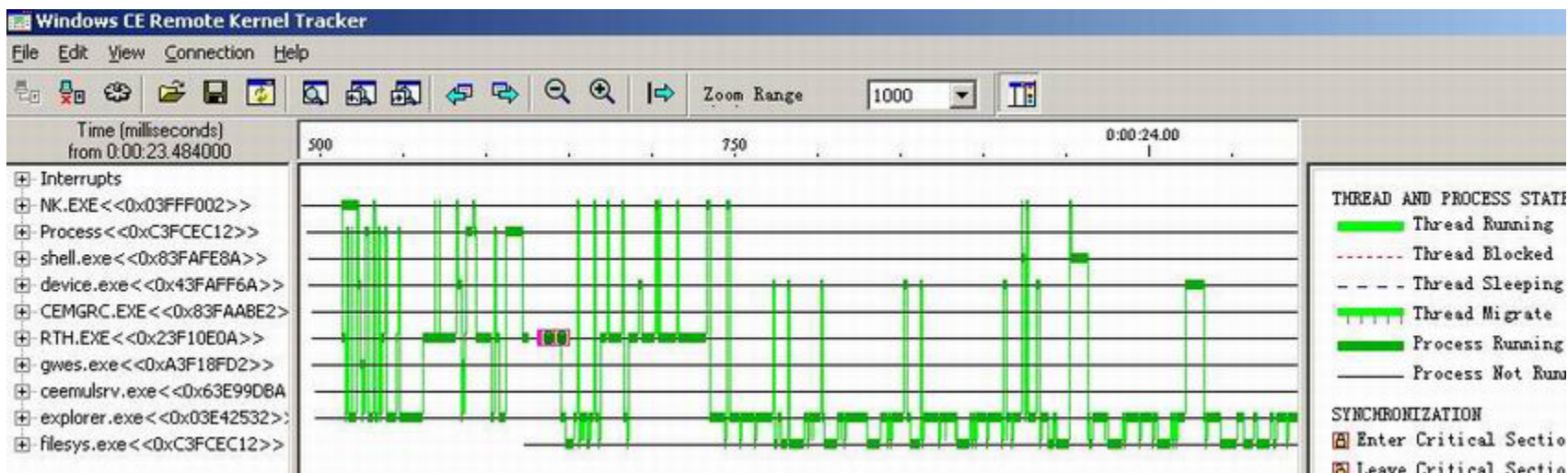
远程堆查看程序（Remote Heap Walker）

- ▶ 查看操作系统中每个进程使用的堆的情况
- ▶ 可以查看
 - ▶ 内核中正运行的进程的名字、ID、及使用的所有的堆的ID
 - ▶ 进程的一个堆中的首地址、结束地址、及标志（Fixed、Free）
 - ▶ 堆中每块（Block）的实际内容



远程内核跟踪程序（Remote Kernel Tracker）

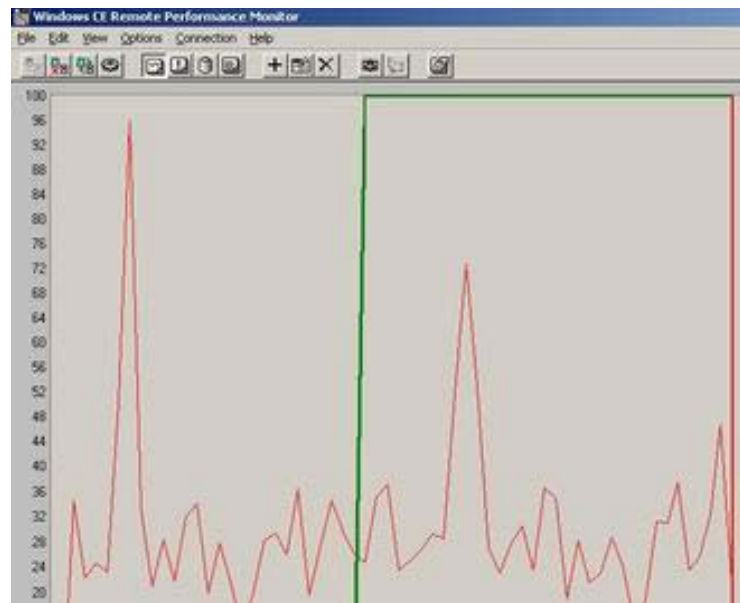
- ▶ 跟踪实际平台内核的执行情况
- ▶ 从开始运行起跟踪，跟踪结束的时间由数据接收缓冲区的大小决定
 - ▶ 缓冲区的大小从1MB到100MB，由用户自己设定
- ▶ 在Build选项中选择
 - ▶ “Enable Event Tracking During Boot”
 - ▶ “Enable Profiling”



远程性能监视程序（Remote Performance Monitor）

▶ 监视项目

- ▶ Remote Access Server (RAS)
 - ▶ Internet Control Message Protocol (ICMP)
 - ▶ TCP/IP
 - ▶ User Datagram Protocol (UDP)
 - ▶ Memory
 - ▶ Battery
 - ▶ System
 - ▶ Process
 - ▶ Thread。
- ▶ 启动后在Edit菜单的Add to Chart中选择监视项



远程进程浏览程序（Remote Process Viewer）

▶ 三个窗口

- ▶ 当前内核中所有进程
- ▶ 进程中的线程
- ▶ 进程中所有加载的DLL

▶ 进程的窗口中显示

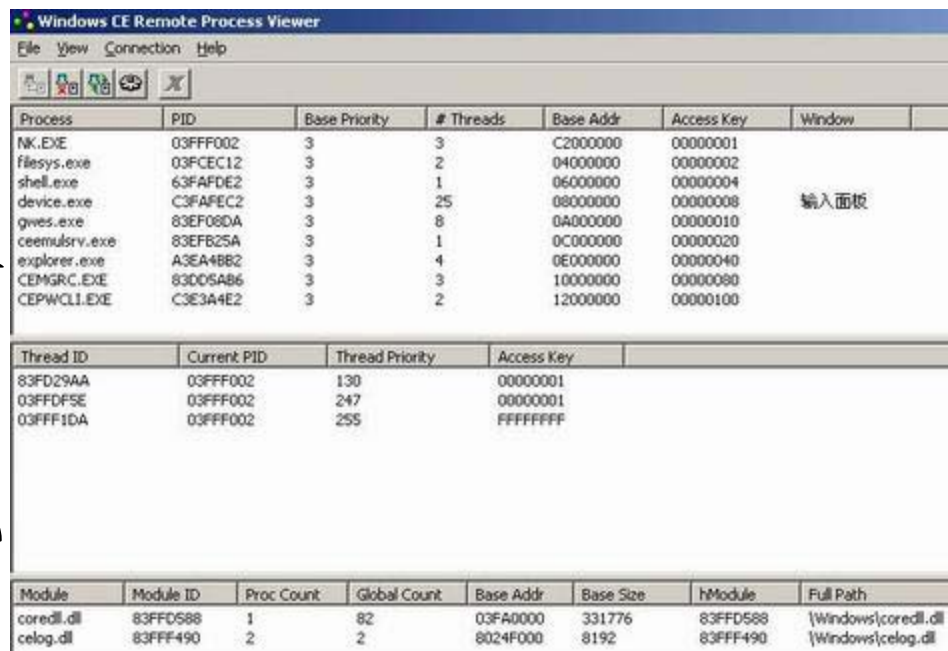
- ▶ 进程名、进程ID、基本优先级级别、拥有的线程总数、基地址、访问键值、主窗口名

▶ 线程的窗口中显示

- ▶ 线程ID、当前进程ID、线程优先级、访问键

▶ DLL模块的窗口中显示

- ▶ 模块名、模块ID、当前进程使用计数、全局使用计数、基地址、大小、模块句柄、路径



The screenshot shows the 'Windows CE Remote Process Viewer' application. It has a menu bar with 'File', 'View', 'Connection', and 'Help'. Below the menu bar are several icons. The main area contains three tables:

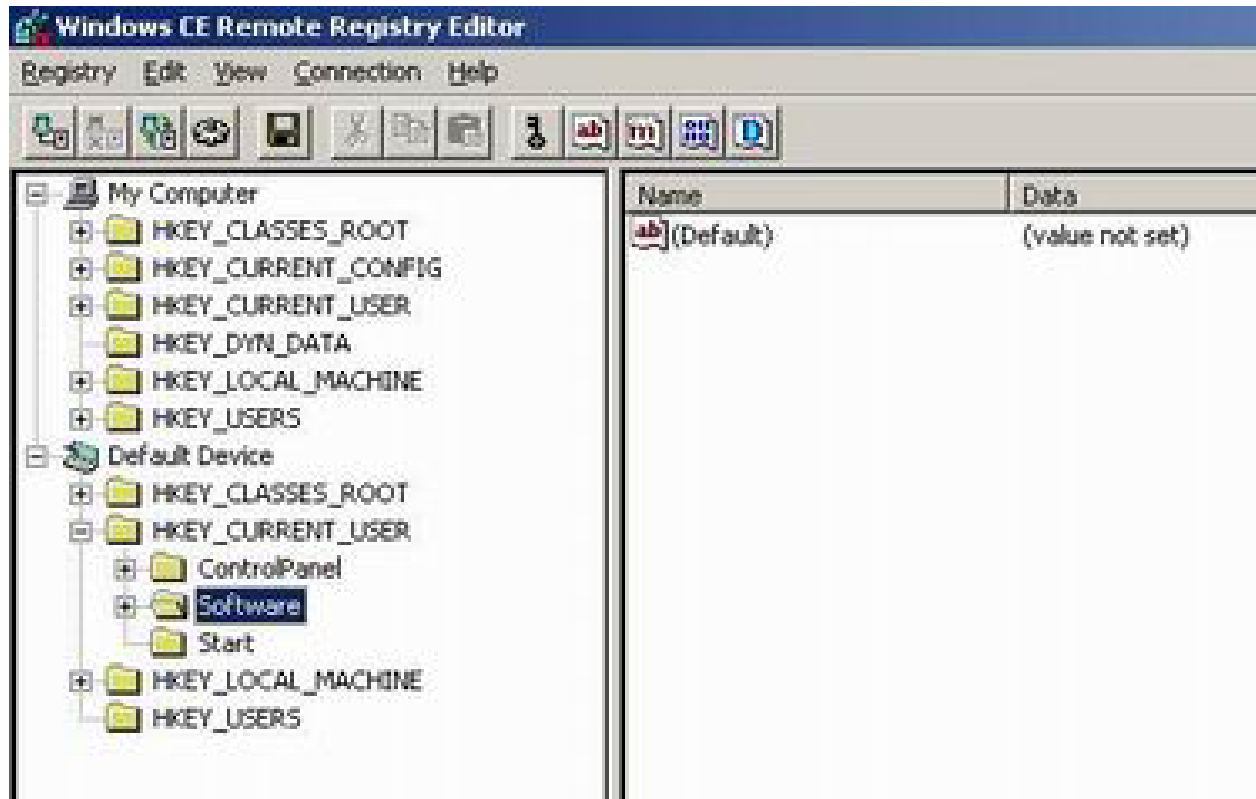
Process	PID	Base Priority	# Threads	Base Addr	Access Key	Window
NK.EXE	03FFF002	3	3	C2000000	00000001	
filesys.exe	03FCEC12	3	2	04000000	00000002	
shell.exe	63FAFDE2	3	1	06000000	00000004	
device.exe	C3FAFEC2	3	25	08000000	00000008	输入面板
gives.exe	83EF08DA	3	8	0A000000	00000010	
ceemulsvr.exe	83EFB25A	3	1	0C000000	00000020	
explorer.exe	A3EA4BB2	3	4	0E000000	00000040	
CEMGR.CEXE	83DD5A86	3	3	10000000	00000080	
CEPWCL1.EXE	C3E3A4E2	3	2	12000000	00000100	

Thread ID	Current PID	Thread Priority	Access Key
83FD29AA	03FFF002	130	00000001
03FFDF5E	03FFF002	247	00000001
03FFF1DA	03FFF002	255	FFFFFFFF

Module	Module ID	Proc Count	Global Count	Base Addr	Base Size	hModule	Full Path
coredll.dll	83FFD588	1	82	03FA0000	331776	83FFD588	\\Windows\coredll.dll
celog.dll	83FFF490	2	2	8024F000	8192	83FFF490	\\Windows\celog.dll

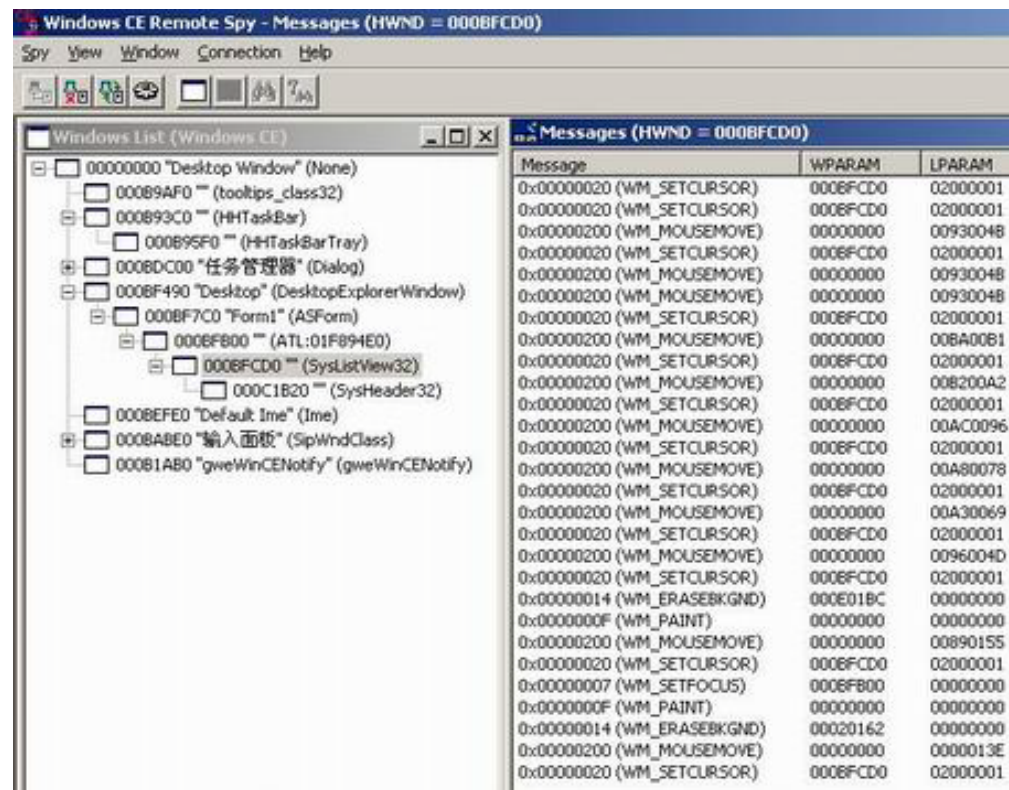
远程注册表编辑程序（Remote Registry Editor）

- ▶ 显示、编辑开发平台下的注册表和实际平台下的注册表



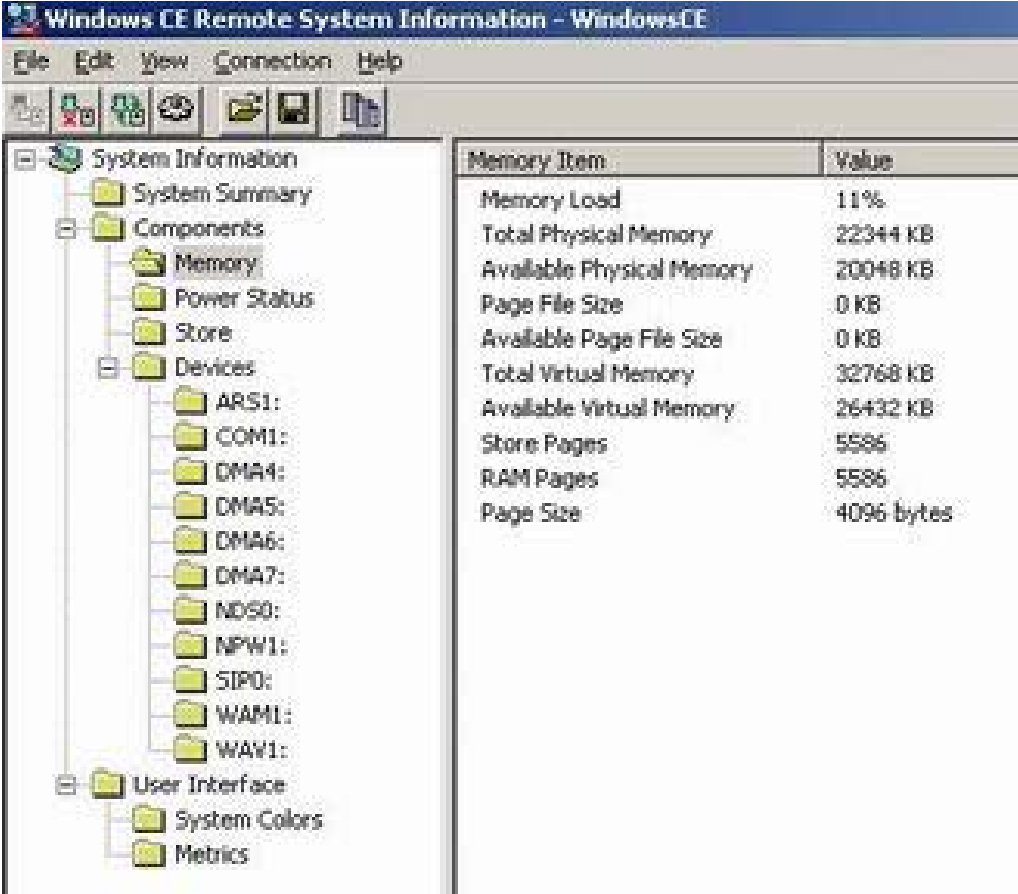
远程消息监视程序（Remote Spy）

- ▶ 和VC下附带的工具spy非常相似
- ▶ 列出平台下的窗口和窗口消息



远程系统信息（Remote System Information）

- ▶ 查看平台的系统信息
 - ▶ 硬件信息
 - ▶ 软件信息



The screenshot displays the 'Windows CE Remote System Information - WindowsCE' application. The interface is divided into two main sections. On the left, a tree view under 'System Information' shows a hierarchy: 'System Summary', 'Components', 'Memory' (selected), 'Power Status', 'Store', 'Devices' (expanded to show ARS1:, COM1:, DMA4:, DMA5:, DMA6:, DMA7:, NDS0:, NPW1:, SIPO:, WAMI:, and WAV1:), and 'User Interface' (expanded to show 'System Colors' and 'Metrics'). On the right, a table lists memory-related metrics and their values.

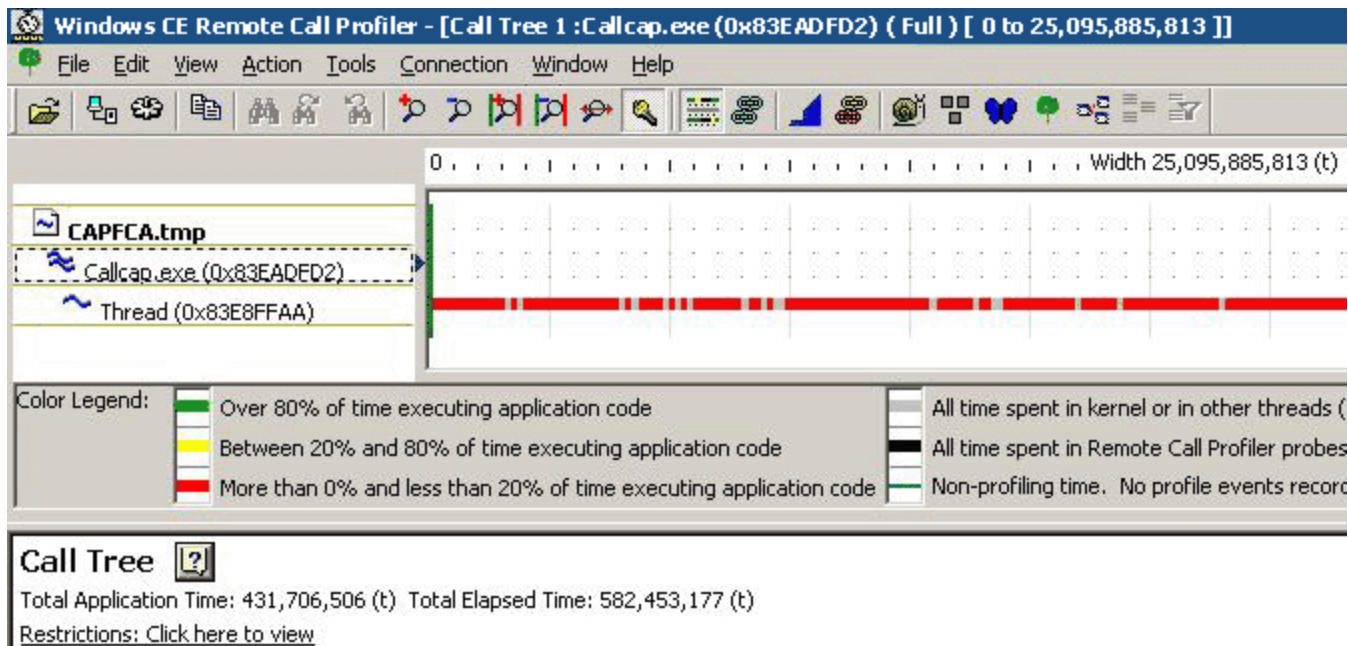
Memory Item	Value
Memory Load	11%
Total Physical Memory	22344 KB
Available Physical Memory	20048 KB
Page File Size	0 KB
Available Page File Size	0 KB
Total Virtual Memory	32768 KB
Available Virtual Memory	26432 KB
Store Pages	5586
RAM Pages	5586
Page Size	4096 bytes

远程屏幕截图程序（Remote Zoom-in）

- ▶ 截取实际平台屏幕图像
- ▶ 适合写说明书或者演示程序
- ▶ 截图单击“File” – “New Bitmap”

远程调用评测程序（Remote Call Profiler）

- ▶ 包含评测和分析工具
- ▶ 利用图形图像显示出你的代码中算法的效率或者占用资源情况
- ▶ 在需要评测的代码中调用提供的接口函数



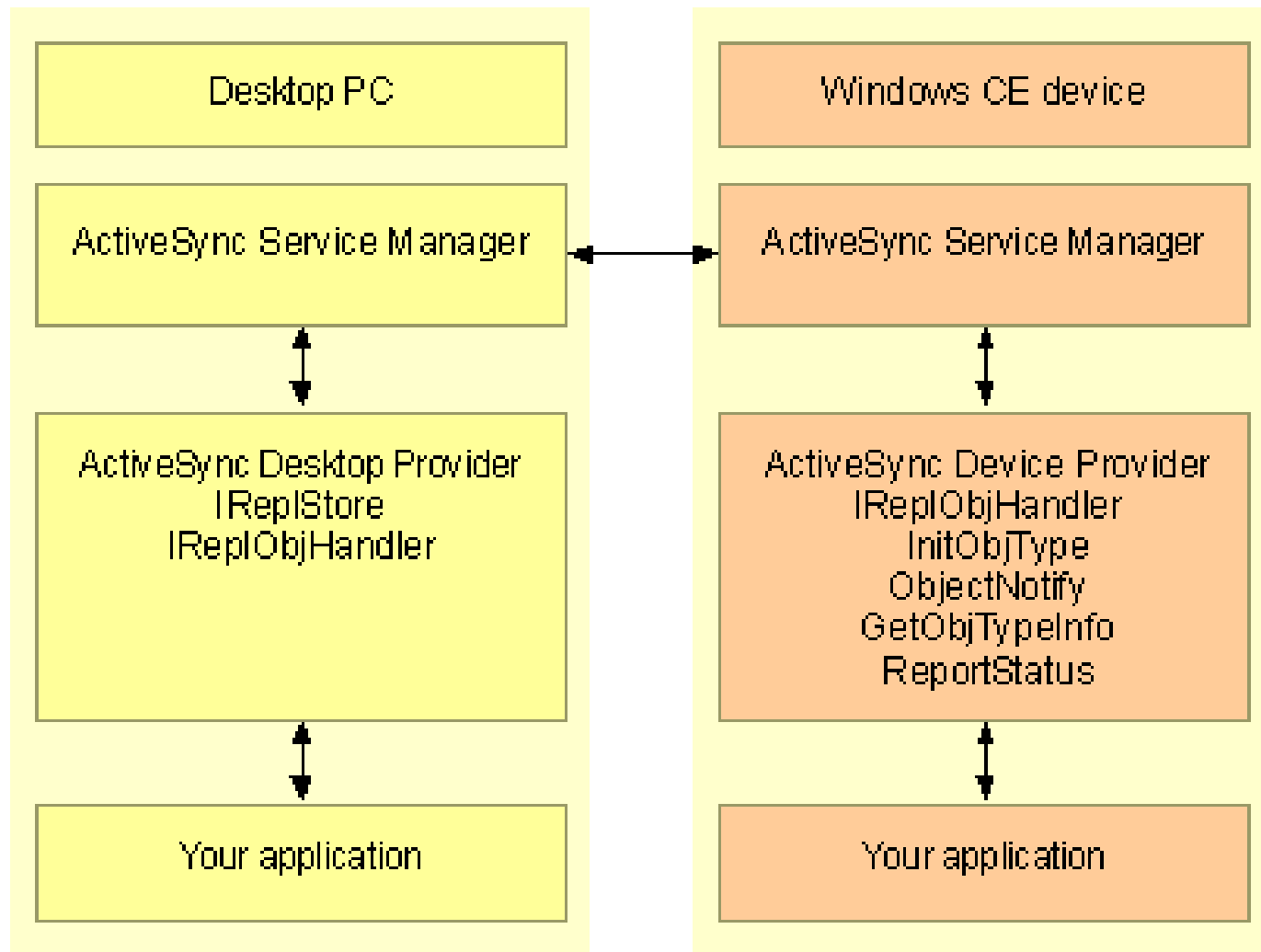
介绍ActiveSync

ActiveSync提供了一种支持，用来同步基于桌面Windows系统和基于CE系统间的数据，它使用时间标记(stamps)和用户参数来跟踪两个系统间的数据改变，并传送相应的数据使得每个系统上的数据都是最新的，过时的或者冗余的数据会被删除。

ActiveSync支持USB, 红外, 调制解调器, 以及 Ethernet 等多种连接。

ActiveSync 不支持CE设备间的数据同步, 也不支持CE设备和服务器间的数据同步。

ActiveSync工作模式



使用ActiveSync调试

- ▶ 安装ActiveSync(从Microsoft网站获得)
- ▶ 宿主机和目标机间提供至少以下一种连接
 - ▶ **Serial**
 - ▶ **USB**
 - ▶ **Infrared**
 - ▶ **Modem**
 - ▶ **Ethernet**
- ▶ 在VS下选择，调试运行在目标机上的程序

仿真器概述

仿真器是一个用来模拟支持Windows CE平台的硬件行为的工具。

使用仿真器，你不需要硬件就可以设计、建立和测试一个**CE**平台，它是用软件的形式来实现对硬件平台的模拟。

在模拟器上映射一个目录

使用模拟器，你可以将一个宿主机上的目录映射到仿真器上，就好像这个目录是安装在模拟器上的一个驱动设备。在模拟器的文件里，这个映射到模拟器上的目录现在处于根目录下，名字是**Storage Card**，模拟器一次只能映射一个目录。

在模拟器上映射一个目录

▶ 映射一个目录到模拟器上

1. 在模拟器窗口里，从模拟器菜单中选择” File”/”Configure”/”Shared Folder”
2. 选择共享文件夹
3. 选择一个目录共享，然后点OK

▶ 删除映射目录

1. 在模拟器窗口里，从模拟器菜单中选择文件夹共享
2. 在共享文件夹中删除目录的路径

模拟器的限制—通讯和网络

- ▶ 不支持拨号连接
- ▶ 不能建立 Remote Desktop Protocol (RDP)
- ▶ 不支持:
 - ▶ IrDA
 - ▶ Bluetooth
 - ▶ 802.11

模拟器的限制—硬件限制

- ▶ Universal serial bus (USB)设备
- ▶ PC Card 设备
- ▶ 存储设备, 包括CF卡和硬盘
- ▶ 文件系统驱动, 包括file allocation table (FAT), compact disc (CD), and digital video disc (DVD) drivers
- ▶ IEEE 1394 devices

仿真器的限制—多媒体限制

- ▶ 不支持对多媒体的硬件加速
 - ▶ DirectX
 - ▶ Windows Media
- ▶ DVD-Video API
- ▶ Digital rights management (DRM)
- ▶ Voice over IP (VoIP)
- ▶ Multiple-screen support

CE6仿真器

▶ 电话模拟器 Cellular Emulator v1

在开发电话应用程序时，开发测试人员可以向电脑里的模拟器拨打电话、收发短信等，就像它就是一台实际的智能手机一样；

▶ 伪GPS (FakeGPS & GPS Settings)

它虽然不能像真正的GPS那样接收真正的数据，但可以读取文本文件里的NMEA数据，还可以将GPSSettings.exe打包在你的GPS应用程序里，让Standard或Professional用户设置其GPS配置