

uTools CE

用户开发与使用指南



福建新大陆自动识别技术有限公司

版本号： V1.1
地址： 福州市马尾区儒江大道 1 号新大陆科技园
邮编： 350015
电话： 0591-83979211
传真： 0591-83979208
公司主页： www.nlscan.com
电子邮件： support@mail.nlscan.com

作者： 郑春光
作者单位： 技术部
文档编号：

提交档案

文档名称及存放路径

名称	uTools_CE 用户使用指南.doc
时间	2009 年 3 月 2 日星期一
大小	489,472 字节
存放	\Release\Doc\98x_doc\ uTools_CE 用户使用指南.doc

修改记录

版本	日期	修改内容	姓名
V 1.0	2008/12/25	创建此文件	郑春光
V 1.1	2009/03/02	最新修订版	李志雄

此文档已由以上人员编辑修改。

审阅记录

日期	姓名	职务	备注	签字

此文档已由以上人员审阅通过。

分发派送

日期	部门	姓名	职务

此文档已分发派送以上人员。

目 录

1	uTools CE 介绍	1
2	uTools CE 安装指南	1
3	编写一个应用程序	2
4	如何执行编译好的程序	3
5	如何调试程序	4
	5.1 在线调试	4
	5.2 如何输出调试信息	4
6	如何创建 CAB 安装包与布署	6
	6.1 创建 CAB 安装包	6
	6.2 布署并安装 CAB 安装文件	6
7	常见问题解决方法	7
	7.1 常见编译与调试问题	7
	7.2 如何移植用于其它设备上的 CE 程序	8
	7.3 其它常见问题	8
8	附录	9
	8.1 PT980 设备按键与键值对应表	9
	8.2 重定义 PT98x 键盘按键功能	10

1 uTools CE 介绍

uTools CE 是一款依托于 Visual Studio .net 2005 的 Add-On 扩展插件型软件，是专门用于开发 Newland 公司 CE 平台应用软件的工具。通过 uTools 可以编写出更高效的，适用于我司 CE 平台设备的软件。

uTools CE 只能使用 C++语言开发，它支持标准的 CE 内核编程接口，并集成了 MFC 类库，使用户开发应用程序的界面时更方便快捷，并且减少再学习所需要花费的时间与人力成本，提高效率。

在我们的 CE 设备上使用 C#或 VB 语言开发 .NET 应用程序，如果需要用到读码、拨号、获取系统信息的 API 接口，需要安装 uTools CE .Net，可以从我们公司的网站上得到。

2 uTools CE 安装指南

安装 uTools CE 之前，您必须确保满足以下的条件：

1. Windows 2000 或 Windows XP 以上版本操作系统。
2. Visual Studio 2005，并且选中了“Visual C++”与“Smart Device Programmability”选项，如下图所示。



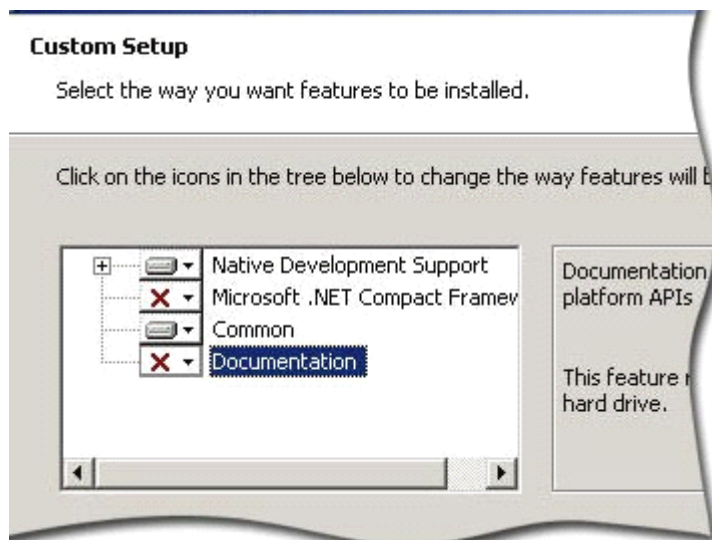
3. Visual Studio .net 2005 Service Pack 1，它包含了一些 Visual Studio .net 2005 的 Bug 修正，提高了兼容性，并且使程序更稳定。
4. Microsoft ActiveSync 4.5，它包含了 Smart Device 设备的 USB 驱动与文件浏览器，如果需要使用在线调试或传输文件的功能就必须安装此软件。

以上几个安装完了即可安装 uTools CE。

注意：如果您是用 **Visual Studio 2008** 开发，**uTools CE**，那么在安装时必须把“**.Net Compact Framework**”与“**Document**”选项去除，否则安装过程将会出错。如下所示：



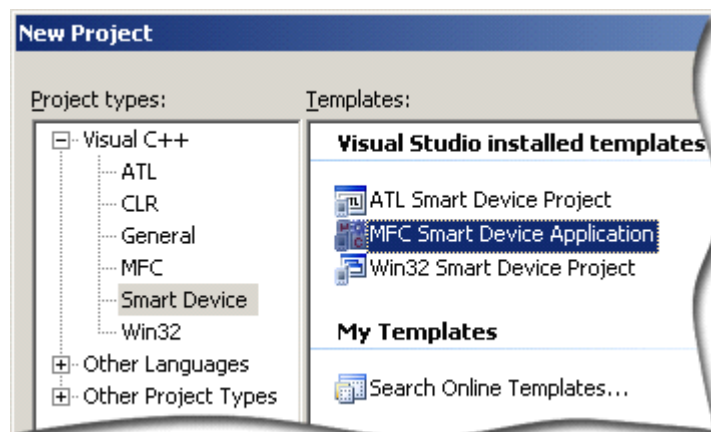
安装时，点 Custom，弹出如下界面：



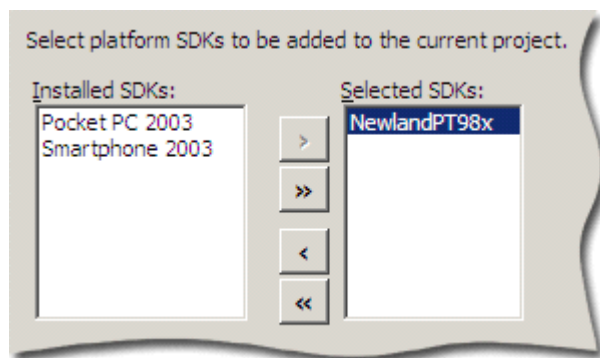
把“.Net Framework”与“Document”选项去除（Visual Studio 2008 环境下）

3 编写一个应用程序

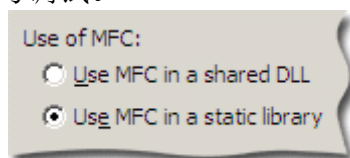
3.1 新建一个 Visual C++ 工程，把工程类型设置为 Smart Device，模板选择 MFC Smart Device Application，如下图所示：



- 3.2 在确定 Platform 时，选择 NewlandPT98x，并且把其它的设备去掉，如下图所示：



- 3.3 在选择以何种方式使用 MFC 库时，建议用户选择“Use MFC in a static library”方式来使用 MFC 动态链接库，因为我们的设备端只提供了 Release 版的 MFC 动态链接库，如果用“Use MFC in a shared DLL”选项，将导致 Debug 版本的程序无法在设备上运行与调试。



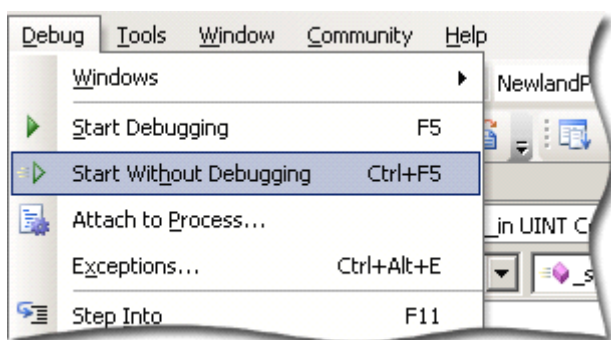
- 3.4 剩下的工程参数可以根据用户自己的需求来确定，没有特别要求。至此，一个应用程序的框架就建立起来了。

4 如何执行编译好的程序

在程序编译完成之后，我们就可以布署程序并运行了。布署程序有两种方式（这里指的是开发与测试过程中的布署）。

一种是把编译生成的目标文件通过 ActiveSync 软件复制到设备的文件系统上，然后从设备的文件浏览器（我的设备）中打开应用程序，这里就可以看到程序运行起来的界面与结果了。

二是通过 Debug 菜单的 Start Without Debugging 选项来运行程序。如下图所示：

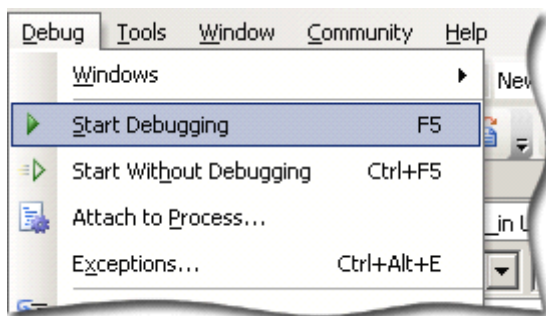


这种方法要求设备必须已经通过 USB 连接到了 PC 上。且 ActiveSync 状态是已连接。运行时，Visual Studio 会先把相应的文件复制到设备的“\Program Files\[Project Name]”目录下，然后再调用程序运行。如果程序有用到额外的数据库、动态库或图片之类的文件，需要手工拷贝到“\Program Files\[Project Name]”目录下。

5 如何调试程序

程序调试有两种手段，一种是直接联机在线调试，通过设置程序运行断点等方法进行单步运行调试，另外一种是不中断程序的运行，而通过程序输出的调试信息来观察参数与定位错误位置。两种模式各有优点与用处，用户可以依据自己的需要选择适当的调试方法。

5.1 在线调试

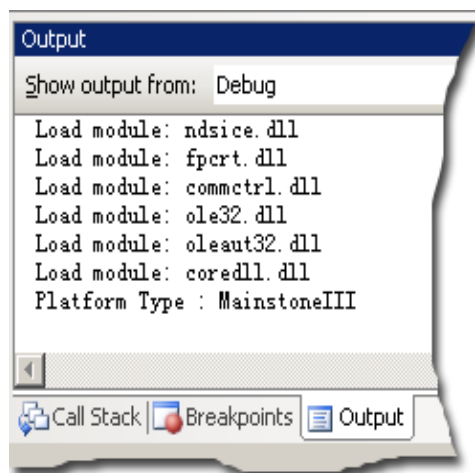


点击运行之后可以在 Output 窗口可以看到如下信息：

“Deploy started: Project: xxx, Configurior xxx”

这表示 Visual Studio 正在向设备布署应用程序，在这过程中，Visual Studio 将把应用程序复制到设备的 “\Program Files\[Project Name]” 目录下，并且把一些应用程序需要的动态链接库传输到设备上，然后就会开始以调试模式运行应用程序。

这一布署过程在第一次调试时所需要的时间比较长，可能需要将近 1 分钟，但之后再行进行调试时，布署速度将会有很大的提高。



5.2 如何输出调试信息

除了联机在线单步调试之外，有时需要输出一些调试信息，当调试信息的量比较小时还可以显示在程序的界面上，但信息量大的时候就需要用其它的解决方法了。

Windows CE 提供了三个接口来输出调试信息，它们的函数原型如下：

void TRACE(LPCWSTR lpFmtStr, ...);

要使用 TRACE 接口，必须加入 MFC 库的支持，否则无法使用。实际上 TRACE 接口是一个宏定义，它只在 Debug 版时起作用，生成 Release 版时将不会有任何操作与信息输出。通常情况下，我们推荐用户使用这个接口。

void NKDbgPrintfW(LPCWSTR lpFmtStr, ...);

NKDbgPrintfW 接口是一个真实存在的内核接口，一量调用它就将输出指定的

信息，而不管是 Debug 版还是 Release 版。如果用户的应用程序不想加入 MFC 库的支持但又想输出调试信息的话，可以使用这个函数。

void OutputDebugString(LPCWSTR lpDbgStr);

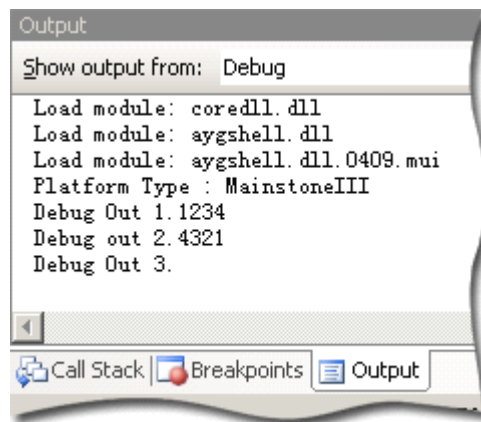
OutputDebugString 接口也是一个真实存在的内核接口，一旦调用将输出指定的信息，而不管 Debug 版还是 Release 版。

这三个接口中，TRACE 与 NKDbgPrintfW 都是可变参数接口，用法与 printf 函数相似，解释命令也与 printf 的接口相一致，方便用户输出某些参数的当前值，OutputDebugString 则没有这个功能，只能输出不经解析的字符串。

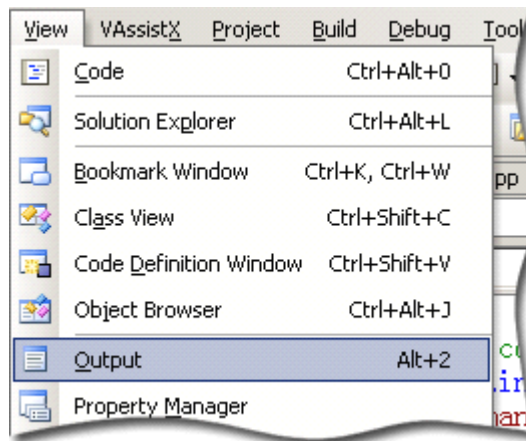
以下是这三个调试信息输出函数的示例代码与输出结果。

示例代码：

```
TRACE(_T("Debug Out 1.%d\r\n"), 1234);  
NKDbgPrintfW(_T("Debug out 2.%d\r\n"), 4321);  
OutputDebugString(_T("Debug Out 3.\r\n"));  
最终显示出来的调试信息：
```



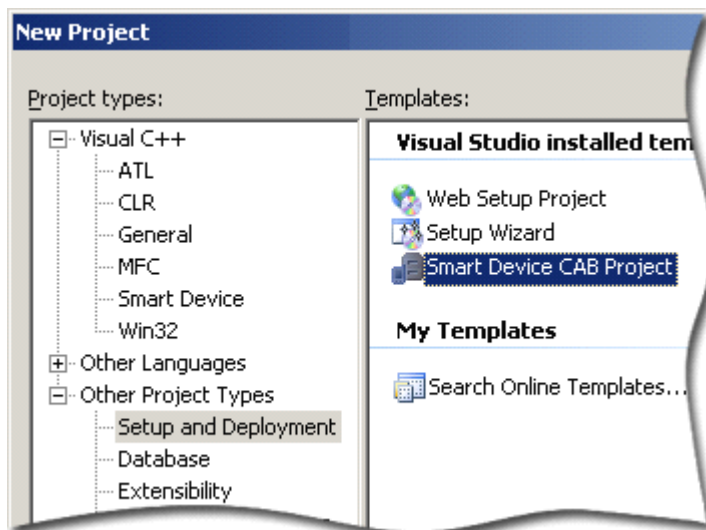
如果看不到 Debug 的 Output 窗口，可以从 View->Output 菜单调出，如下图所示：



6 如何创建 CAB 安装包与布署

6.1 创建 CAB 安装包

程序开发完成之后，可以把所需要的文件打包在一起，生成一个 CAB 安装文件，用于在 CE 设备上布署。Visual Studio 里已经提供了这个工具。要创建一个 CAB 安装包，你需要新建一个项目，如下图所示，项目类型选择 Setup and Deployment，模板选择 Smart Device CAB Project，就可以创建一个安装包的工程了。关于更具体的配置与安装包生成操作，请参考微软的 Visual Studio 开发手册。



6.2 布署并安装 CAB 安装文件

安装包生成之后就可以进行布署了。布署的方法是把 CAB 安装文件通过 ActiveSync 软件发送到设备上，然后在设备上运行这个 CAB 安装包，如果不需要更改默认的安装路径的话，直接按回车就可以完成安装过程。

注意：

CAB 文件在安装完成之后会被系统自动删除。

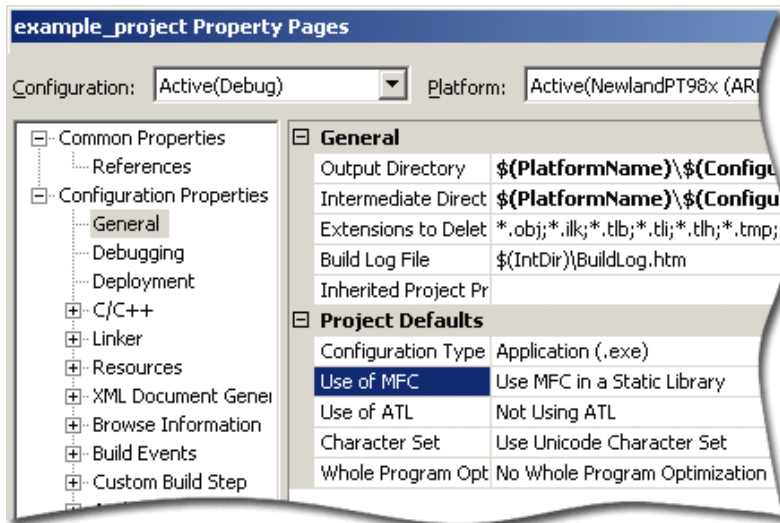
7 常见问题解决方法

7.1 常见编译与调试问题

7.1.1 为什么编译程序都没问题，但只有 **Release** 版的程序可以运行，**Debug** 版的无法运行且没有任何提示？

如果你在建立新工程的时候，选择 MFC 运行时刻库的链接与使用方式时选择了“Use MFC in a shared DLL”，那么就会出现这个问题。这是因为我们的设备上只提供了 Release 版的 MFC 运行时刻库，所以导致 Debug 版的程序无法运行。

解决方法是调出工程的属性设置对话框，然后选中 Configuration Properties 下的 General 选项，然后找到右边的 Use of MFC 选项，并把它修改为“Use MFC in a Static Library”，如下图所示，最后保存设置，再编译一遍代码就可以解决这个问题了。



7.1.2 可以用 **Visual Studio** 的 **Pocket PC** 平台开发软件吗？

如果你的应用程序并不需要进行平台相关的操作，如控制识读头读码等操作，那么你就可以在 Pocket PC 平台下开发软件，最后再布署到我们的 CE 终端设备上。

7.1.3 用 **uTools CE** 平台编译的软件可以在其它公司的 **CE** 平台，如 **Pocket PC** 上运行吗？

不保证能在其它公司的 CE 平台上运行，除非用相应平台的 SDK 重新编译过后，因为我们的设备可以兼容 Pocket PC 的二进制代码，但 Pocket PC 或其它设备则不一定能兼容 uTools CE 生成的二进制代码。

7.1.4 为什么编译时总会出现“warning C4996”警告信息？

在使用 uTools CE 编译程序时，会出现如下所示的多个类似警告信息，

warning C4996: 'wcsncpy' was declared deprecated

Message: 'This function or variable may be unsafe. Consider using wcsncpy_s instead. To disable deprecation, use _CRT_SECURE_NO_DEPRECATED. See online help for details.'

这是微软公司编译器与头文件导致的问题。因为微软公司不推荐用户使用这些函数，但实际上这些函数本身是没问题的，而且简化了需要的参数，只是使用时要求程序员要把握好它们的特性，以免写出不够安全与健壮的程序。

7.1.5 如何禁止 “warning C4996” 警告信息？

繁多的警告信息让我们感觉编译信息太多，可能会忽略掉需要关注的其它警告或出错信息，所以用户有时需要屏蔽掉这些警告信息。屏蔽的方法有两种，列出如下。

1. 在项目的 stdafx.h 中寻找 #pragma once 语句，在它的后面加入如下所示的宏定义：`#define _CRT_SECURE_NO_DEPRECATED`

2. 在项目的 stdafx.h 中寻找 #pragma once 语句，在它的后面加入如下所示的编译控制语句：`#pragma warning(disable:4996)`

7.1.6 为什么我的程序已经编译通过，但无法在 PC 上运行？

用 uTools CE SDK 平台生成的应用程序只能在指定的设备上运行，而不能在 PC 上直接运行。

7.1.7 为什么我无法在 Visual Studio 环境下调试程序？

要想在 Visual Studio 环境下调试程序，你必须确认已经安装了 Microsoft ActiveSync 4.5 以上版本的软件，并且设备已经通过 USB 接口正确连接到 PC 上。否则 Output 窗口将会出现如下出错信息：

```
Device is not ready. If target is an ActiveSync device, verify whether
ActiveSync connection is running.
Connect to device failed.
```

7.2 如何移植用于其它设备上的 CE 程序

不同的 CE 设备平台，在系统接口与图形界面上的接口基本兼容，因此，移植的重点要放在有重大区别的条码数据采集部分。我司的设备使用一套自定义的扩展接口 NDSI API 来支持数据采集与识读控制，接口的具体定义与使用请参考 uTools CE 开发手册的识读头控制部分。

在进行程序移植时不建议直接在原来的工程中进行修改，建议新建一个 uTools CE 的工程项目，然后再把原来工程的代码与资源添加进来，然后在这个新的工程基础上进行修改与编译，以避免因不同平台带来的兼容性问题导致的程序出错。

7.3 其它常见问题

7.3.1 代码在 PC 上编译通过，但复制到设备上去之后却无法运行，也看不到任何提示。

这很可能是由于设备上缺少某些程序所需要的动态链接库所引起的。

由于没有任何提示，你可能需要认真检查你的程序是否有依赖于什么系统没提供的动态链接库或用户自己编写的动态链接库。或者你可以使用 Visual Studio 提供的 Depends.exe 程序来查看你的应用程序所需要的动态链接库，在 Visual Studio 的安装路径下可以搜索到它。

7.3.2 如何在 PC 上编辑设备上的注册表信息？

因为设备的屏幕通常较小，而且按键不丰富，所以在设备上进行注册表编辑操作很烦琐。针对这个问题，微软在 Visual Studio 2005 软件中提供了一个工具，名字为 Remote Registry Editor，用这个工具软件我们可以方便地在 PC 上修改设备的注册表内容，使用方法与 Regedit 基本相同，只是初始化时要求选择一个设备来进行操作。

7.3.3 如何在截取设备上的屏幕显示？

有时需要为编写好的软件制作一份用户手册，这时就需要有一个手段来获取设备屏幕上的显示内容了。针对这个问题，微软的 Visual Studio 2005 提供了一个名为 Remote Zoom In 的工具软件，它可以方便地截取设备上的显示信息，并保存为 BMP 图片，方便用户的文档编写与制作。

8 附录

8.1 PT980 设备按键与键值对应表

PT98x Keyboard	PC Keyboard	Windows Define	Hex Value
1	1	' 1'	0x31
2	2	' 2'	0x32
3	3	' 3'	0x33
4	4	' 4'	0x34
5	5	' 5'	0x35
6	6	' 6'	0x36
7	7	' 7'	0x37
8	8	' 8'	0x38
9	9	' 9'	0x39
0	0	' 0'	0x30
*	F7	VK_F7	0x76
#	F8	VK_F8	0x77
Up	上	VK_UP	0x26
Down	下	VK_DOWN	0x28
Left	左	VK_LEFT	0x25
Right	右	VK_RIGHT	0x27
OK	Enter	K_RETURN	0x0d
C	Esc	VK_ESCAPE	0x1b
BackSpace	BackSpace	VK_BACK	0x08
Space	Space	VK_SPACE	0x20
Fn	F16	VK_F16	0x7f
Scan	F15	VK_F15	0x7e
Left Side Key	F13	VK_F13	0x7c
Right Side Key	F14	VK_F14	0x7d

8.2 重定义 PT98x 键盘按键功能

功能描述：通过修改注册表键值实现重定义 PT98x 系列产品键盘按键(除电源键外)对应的键值的功能（例如：要把键盘上其中的某一个按键定义成"."）。该功能需重启设备才能生效。

适用版本：PT980x V1.3.4 Build080725 及以后版本。

打开设备的注册表有如下两种方式：

1. 可以通过电脑端来打开并修改设备上的注册表。使用 USB 线连接 PT98x 设备和 PC 机，并使用 Visual Studio 2005 的远程注册表编辑器工具（如："开始 -> 所有程序 -> Microsoft Visual Studio 2005 -> Visual Studio Remote Tools -> Remote Registry Editor"），打开 PT98x 设备的注册表。

2. 点设备的"开始->运行"，敲入"regedit"按回车即可在设备上打开注册表。

使用说明：

首先打开设备上的注册表，在注册表项中找到 HKLM\PDA\KeyMap\VKey 目录。如果不存在，则创建此目录。

然后在该目录下新建/修改注册表数据。数据类型为 DWORD,数据名称如表 1-1 所示。数据内容为按键的虚拟键值。

键盘按键	数据名称
左侧按键	"LeftSide"
右侧按键	"RightSide"
OK 键	"OK"
取消键	"Cancel"
向左键	"Left"
向右键	"Right"
向上键	"Up"
向下键	"Down"
SCAN 键	"Scan"
删除键	"BackSpace"
空格键	"Space"
Fn 键	"Fn"
号键	""
#号键	"#"
按键 "1"	"1"
按键 "2"	"2"
按键 "3"	"3"
按键 "4"	"4"
按键 "5"	"5"
按键 "6"	"6"
按键 "7"	"7"
按键 "8"	"8"
按键 "9"	"9"
按键 "0"	"0"

表 1-1 PT98x 设备键盘按键对应注册表数据名称

例子：

要把键盘上的“*”键定义为“.”则在该目录下新建或修改一个名称为“*”的注册表项，注册表项的值为190，类型为DWORD。（190为“.”按键的虚拟键值。每个按键对应的虚拟键值可通过MSDN查得。）