

Microsoft® Virtual Labs

**Monitoring Resource Leaks Using
the Application Verifier Tool for
Windows Mobile 5.0**

Microsoft®

Table of Contents

Monitoring Resource Leaks Using the Application Verifier Tool for Windows Mobile 5.0 . 1	
Exercise 1 Launching ActiveSync and the Windows Mobile 5.0 Pocket PC Phone Emulator	2
Exercise 2 Copying the Required Program Files from the Desktop Computer to the Emulator by Using ActiveSync.6	
Exercise 3 Launching and Setting Up the Windows CE 5.0 Test Kit.....	7
Exercise 4 Launching the Application Verifier Tool from the Windows CE 5.0 Test Kit and Setting It Up to Run the Test Application	12
Exercise 5 Running the Test Application and Obtaining the Logged Data	16
Exercise 6 Analyzing the Logged Data with the Application Verifier Tool.....	18
Summary.....	23

Monitoring Resource Leaks Using the Application Verifier Tool for Windows Mobile 5.0

Objectives

The objective of this lab is to use the Microsoft Windows CE 5.0 Test Kit and the Application Verifier Tool to monitor possible resource leaks and memory corruptions while running an application on a Windows Mobile 5.0 Pocket PC Phone Emulator. You will then use the Application Verifier Tool to examine the generated logs.

In this lab, you will perform the following exercises:

- Launching ActiveSync and the Windows Mobile 5.0 Pocket PC Phone Emulator
- Copying the required program files from the desktop computer to the emulator by using ActiveSync Etc
- Launching and setting up the Windows CE 5.0 Test Kit
- Launching the Application Verifier Tool from the Windows CE 5.0 Test Kit and setting it up to run the test application
- Running the test application and obtaining the logged data
- Analyzing the logged data with the Application Verifier Tool

Scenario

This lab uses ActiveSync to make a connection between the Pocket PC Phone Emulator and the development workstation. If this connection is lost, be sure it is restored. For more information about setting up an ActiveSync connection, see “Launch and set up Microsoft ActiveSync” in Exercise 1. Also, be sure that the Pocket PC Phone Emulator is running and is cradled at all times in this lab.

Estimated Time to Complete This Lab

45 Minutes

Computer used in this Lab



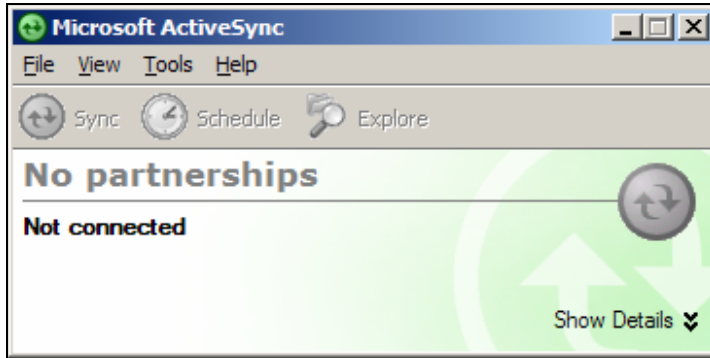
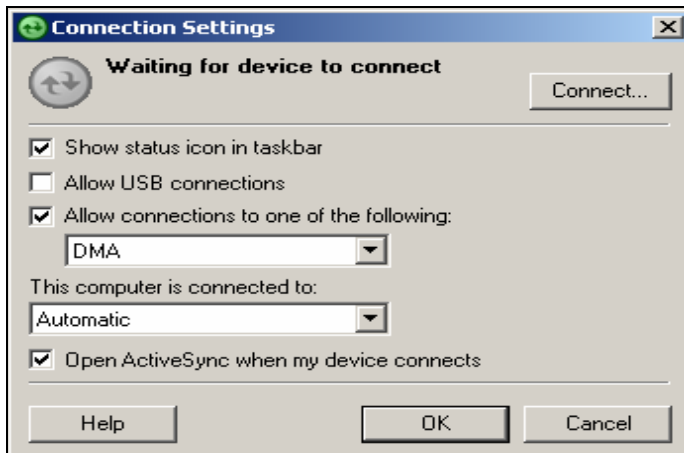
MEDC

Exercise 1

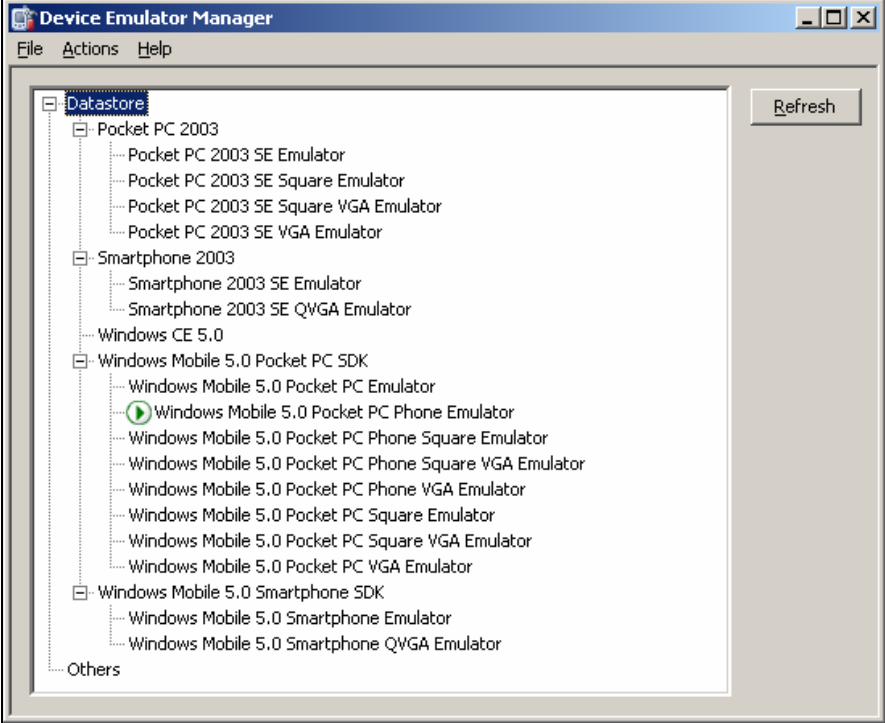
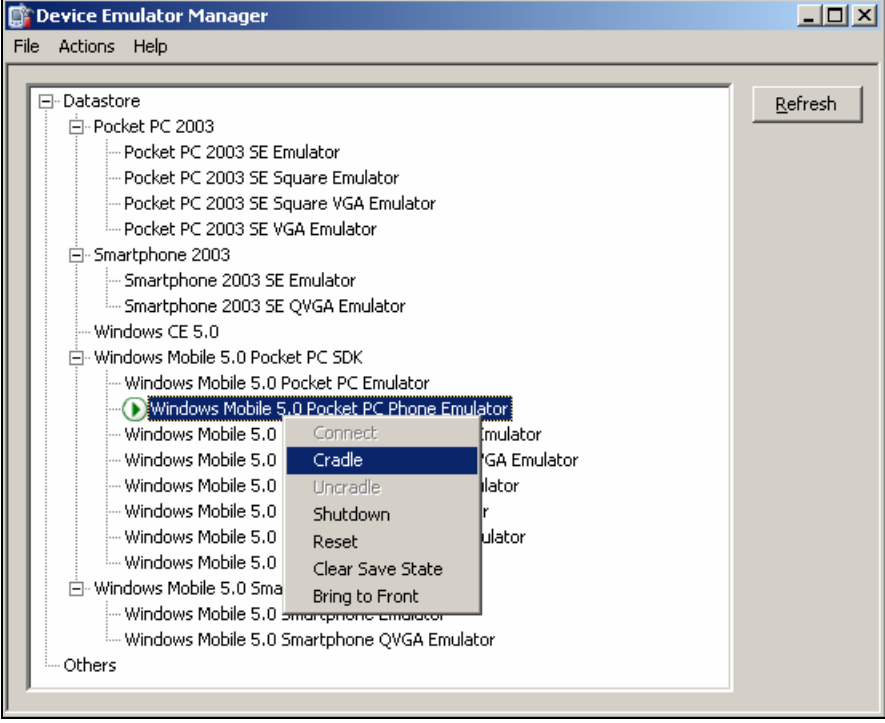
Launching ActiveSync and the Windows Mobile 5.0 Pocket PC Phone Emulator

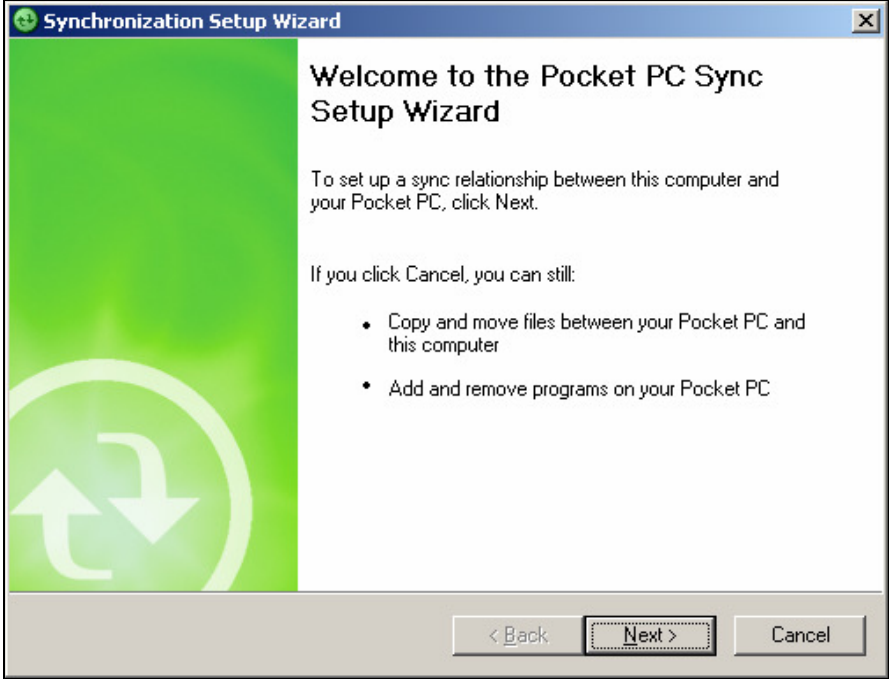
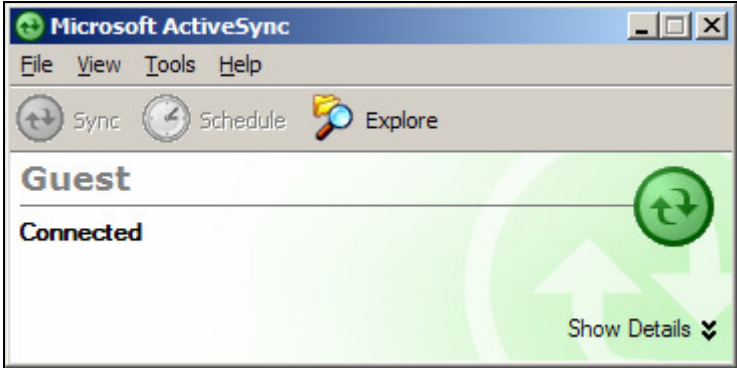
Scenario

In this exercise, you will launch ActiveSync and then launch the Windows Mobile 5.0 Pocket PC Phone Emulator from Visual Studio 2005.

Tasks	Detailed Steps
1. Launch and set up ActiveSync	a. On the desktop computer, click Start All Programs Microsoft ActiveSync. This step launches the Microsoft ActiveSync, as shown in Figure 1.  Figure 1. Microsoft ActiveSync b. In ActiveSync, click File Connection Settings. c. Select the following setting in the Connection Settings dialog box, as shown in Figure 2: - Select Show status icon in taskbar. - In the Select Allow connections to one of the following box, select DMA. - In the This computer is connected to box, select Automatic. - Select Open ActiveSync when my device connects. 

Tasks	Detailed Steps
	Figure 2. Connection Settings dialog box d. Click OK to save the settings. e. Minimize ActiveSync.
2. Launch the Windows Mobile 5.0 Pocket PC Phone Emulator from Visual Studio 2005	a. On the desktop computer, click Start All Programs Microsoft Visual Studio 2005 Microsoft Visual Studio 2005. b. In Visual Studio, click Tools Device Emulator Manager to open the Device Emulator Manager, as shown in Figure 3. Figure 3. Device Emulator Manager c. Minimize Visual Studio. d. In the Device Emulator Manager window, right-click Windows Mobile 5.0 Pocket PC Phone Emulator in the list of emulators, and then click Connect to launch the emulator. The emulator opens. e. Wait for about a minute or so as the emulator boots up completely. If you get a dialog box that says that a connection error has occurred, dismiss and ignore it. After the emulator is booted, you should see the Today screen on the emulator. f. Minimize the Pocket PC Phone Emulator window. g. In the Device Emulator Manager window, verify that there is a green icon next to Windows Mobile 5.0 Pocket PC Phone Emulator, as shown in Figure 4. If the green icon is not present, the emulator may not have fully booted up yet. Wait for a few more moments.

Tasks	Detailed Steps
	 Figure 4. Green icon symbolizing that the emulator is connected h. Right-click Windows Mobile 5.0 Pocket PC Phone Emulator, and then click Cradle to cradle the emulator, as shown in Figure 5.
	 Figure 5. Cradling the emulator i. Wait for a minute or so as ActiveSync connects to the emulator. j. If you get an ActiveSync message, “You will be able to synchronize only files,

Tasks	Detailed Steps
	folders, and other non-Outlook items if you continue”, click OK. <i>Note: The Synchronization Setup Wizard appears, as shown in Figure 6.</i>  Figure 6. Synchronization Setup Wizard k. Click Cancel to close the wizard. <i>Note: The Microsoft ActiveSync window appears.</i> l. Verify that the text Connected appears in the Microsoft ActiveSync window, as shown in Figure 7.  Figure 7. ActiveSync: Connected m. Minimize the Microsoft ActiveSync window, and then minimize the Device Emulator Manager window. <i>Note: If you don't get to this point, be sure your ActiveSync connection settings are as specified in Step 2 of the "Launch and set up Microsoft ActiveSync" procedure.</i>

Exercise 2

Copying the Required Program Files from the Desktop Computer to the Emulator by Using ActiveSync

Scenario

In this exercise, you will copy the required program files from the desktop computer to the emulator.

Tasks	Detailed Steps
1. Copy the required program files from the desktop computer to the emulator	a. On the Desktop, find the Lab Program Files folder, and then double-click it. b. Select leakdemo.exe, and then click Edit Copy. c. On the desktop computer, click Start All Programs Microsoft ActiveSync. d. On the Microsoft ActiveSync window, click the Explore icon to open the File Explorer, as shown in Figure 8. e. Double-click the My Windows Mobile-Based Device folder. f. Paste the copied program file, leakdemo.exe, to this location. If a message appears that says that ActiveSync may need to convert the file, click OK. g. Wait for a few moments as the file is copied onto the emulator. h. Close File Explorer and the Lab Programs Files folder. i. Minimize the Microsoft ActiveSync window.

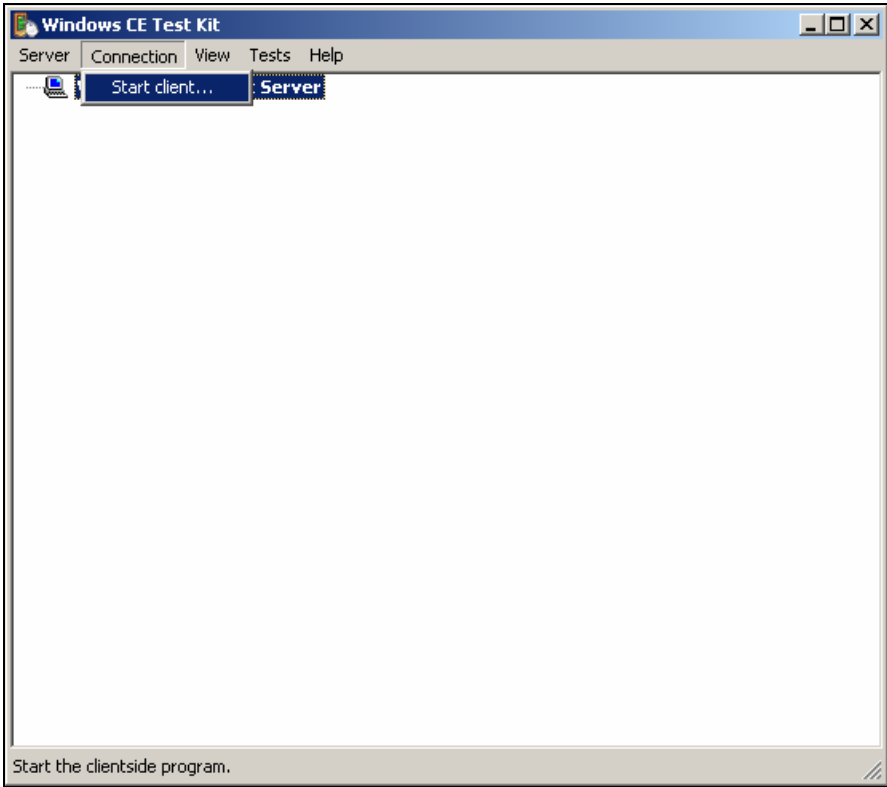
Figure 8. File Explorer on the desktop computer

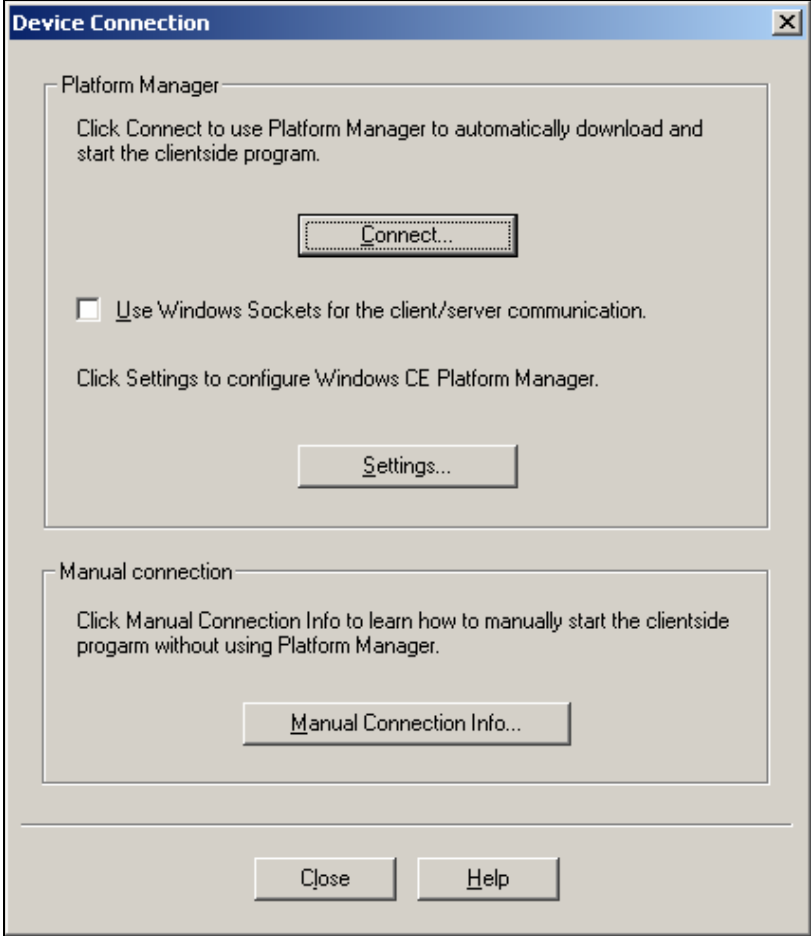
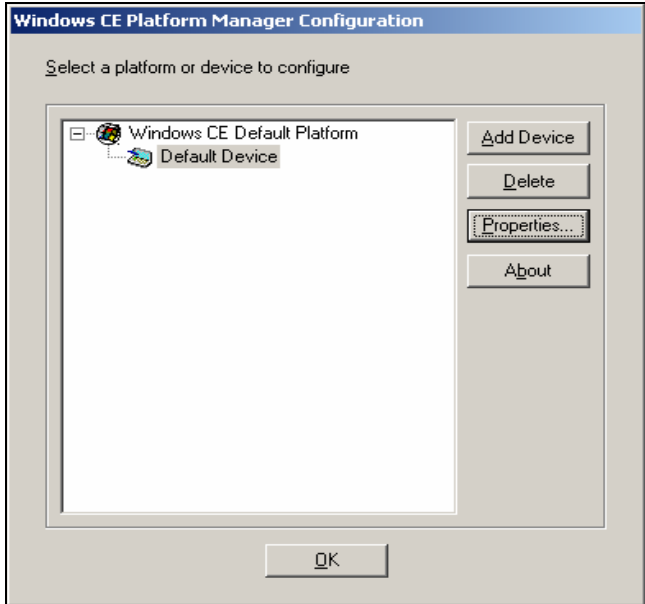
Exercise 3

Launching and Setting Up the Windows CE 5.0 Test Kit

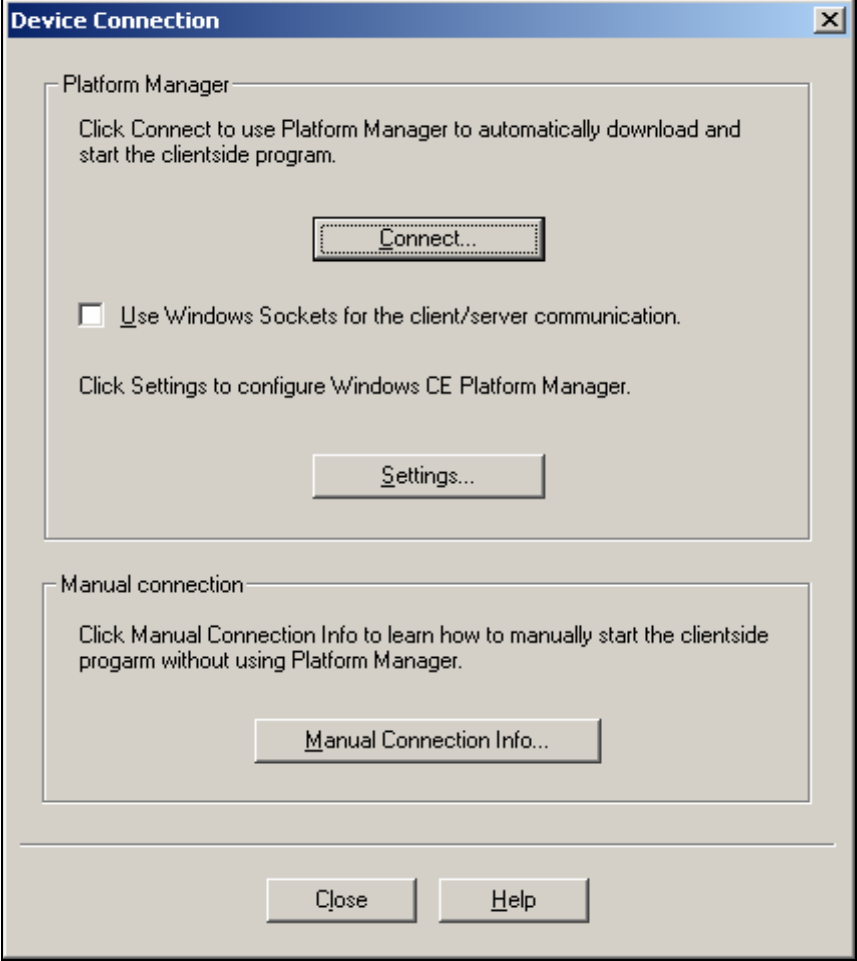
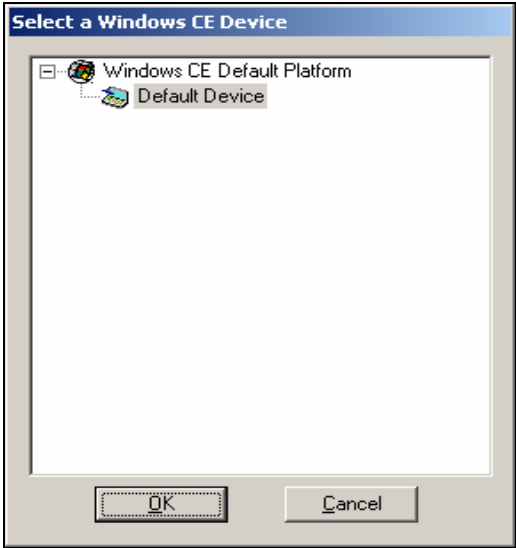
Scenario

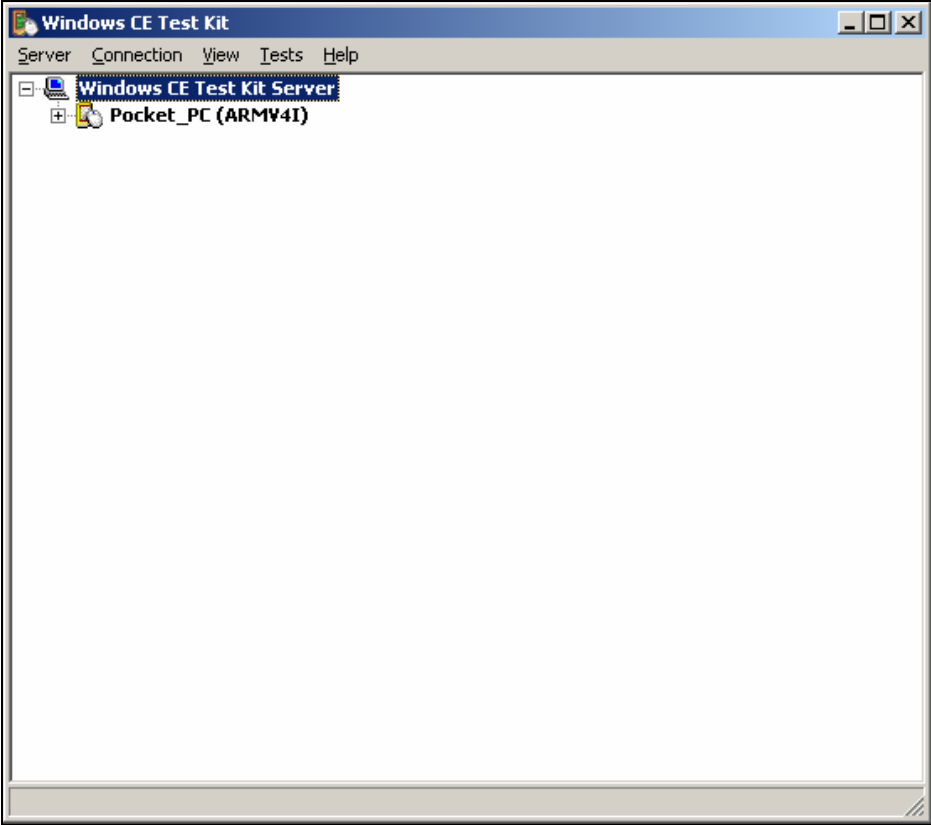
In this exercise, you will launch and set up the Windows CE 5.0 Test Kit (CETK).

Tasks	Detailed Steps
1. Launch and set up the CETK	a. On the desktop computer, click Start All Programs Microsoft Windows CE 5.0 Microsoft Windows CE 5.0 Test Kit. b. If you get a Windows Security Warning dialog box with the warning “Windows Firewall has blocked some features of this program,” click the Unblock button to enable all features of the Test Kit. <i>Note: If the Windows Firewall security warning appears, it may not appear in the foreground. You may need to use the Windows taskbar to activate the window to unblock the program. You should be aware of this note as you continue through the following steps.</i> c. In the Windows CE Test Kit window, click Connection Start Client, as shown in Figure 9.  Figure 9. Windows CE Test Kit d. The Device Connection dialog box appears, as shown in Figure 10.

Tasks	Detailed Steps
	The 'Device Connection' dialog box has a title bar with a close button. It contains two main sections. The 'Platform Manager' section has a text box explaining that clicking 'Connect' will use Platform Manager to download and start the program. Below this is a 'Connect...' button. There is an unchecked checkbox labeled 'Use Windows Sockets for the client/server communication.' and a 'Settings...' button. The 'Manual connection' section has a text box explaining that clicking 'Manual Connection Info' will provide instructions on how to start the program manually. Below this is a 'Manual Connection Info...' button. At the bottom are 'Close' and 'Help' buttons. Figure 10. Device Connected dialog box e. Click Settings. The Windows CE Platform Manager Configuration dialog box appears, as shown in Figure 11. The 'Windows CE Platform Manager Configuration' dialog box has a title bar. Below the title bar is a label 'Select a platform or device to configure'. It features a list box containing 'Windows CE Default Platform' and 'Default Device'. To the right of the list box are four buttons: 'Add Device', 'Delete', 'Properties...', and 'About'. An 'OK' button is located at the bottom center of the dialog.

Tasks	Detailed Steps
	Figure 11. Platform Manager Configuration dialog box f. Under Windows CE Default Platform, select Default Device. g. Click Properties. The Device Properties dialog box appears. h. In the Transport box, select Microsoft ActiveSync. i. In the Startup Server box, select Microsoft ActiveSync, as shown in Figure 12. Figure 12. Device Properties dialog box j. Click OK to save the settings. k. If a dialog box appears with the message, “You have not tested with the current transport and/or server,” click Yes to dismiss the message. l. On the Windows CE Platform Manager Configuration dialog box, click OK.
2. Connect the CETK to the emulator	a. On the Device Connection dialog box, be sure the Use Windows Sockets for the client/server communication check box is not selected, as shown in Figure 13.

Tasks	Detailed Steps
	 Figure 13. Device Connection dialog box b. Click Connect to make a connection between the Test Kit and the emulator. <i>Note: The Select a Windows CE Device dialog box appears, as shown in Figure 14.</i>  Figure 14. Select a Windows CE Device for connection

Tasks	Detailed Steps
	c. Select Default Device, and then click OK to initiate the connection. d. Wait for a few moments as the connection is made. e. If you get a Windows Security Warning dialog box with the warning, “Windows Firewall has blocked some features of this program,” click Unblock. <i>Note: The Pocket PC emulator should now be listed in the Windows CE Test Kit window, as shown in Figure 15.</i>  The screenshot shows a window titled 'Windows CE Test Kit' with a menu bar containing 'Server', 'Connection', 'View', 'Tests', and 'Help'. Below the menu bar is a tree view with a plus sign icon and the text 'Windows CE Test Kit Server'. Underneath this, there is a plus sign icon and the text 'Pocket_PC (ARMV4I)'. Figure 15. The emulator appears in the Windows CE Test Kit window <i>Note: Although the Pocket PC Phone Edition device emulator may be minimized, it displays a Client Side window with continually updating status messages.</i>

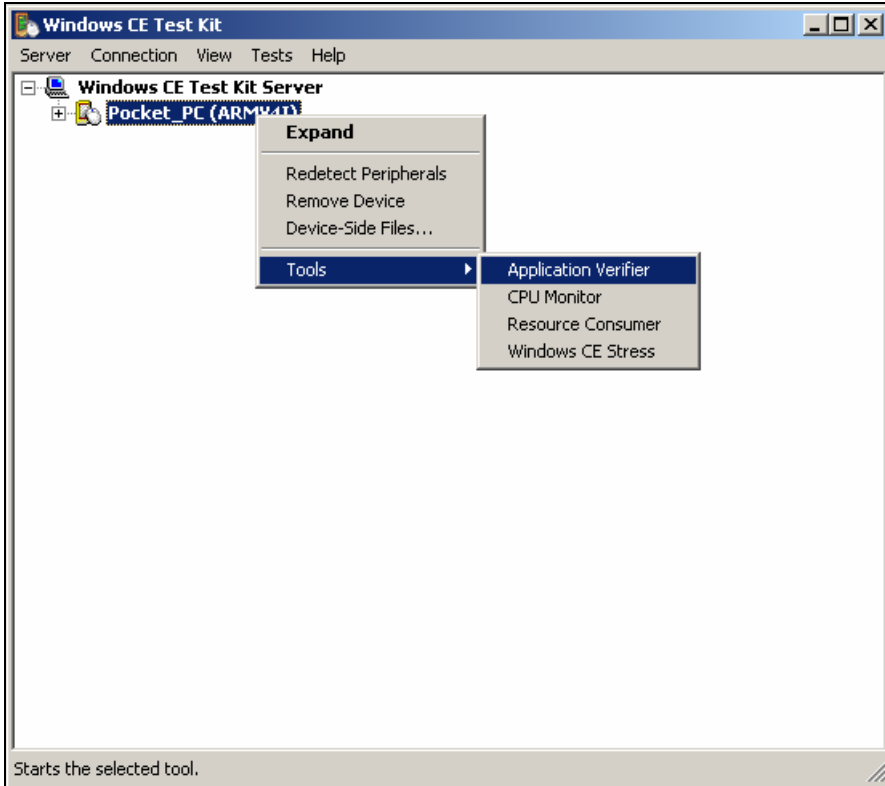
Exercise 4

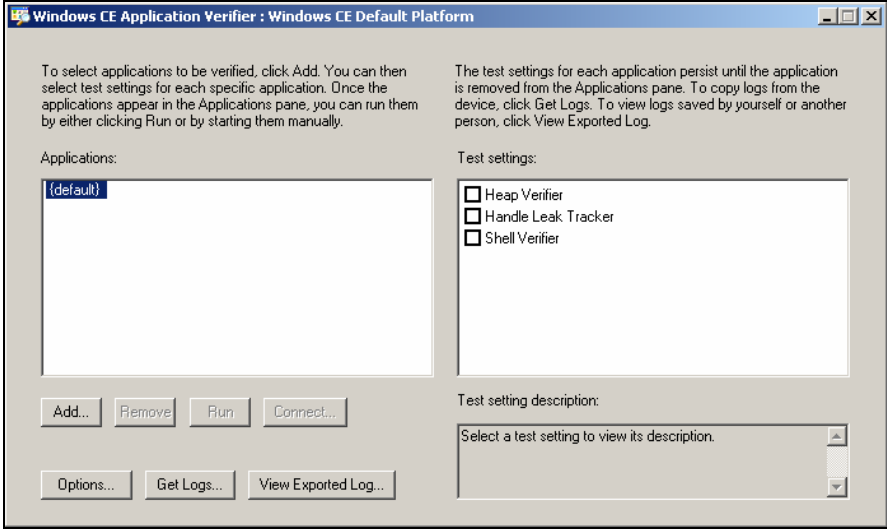
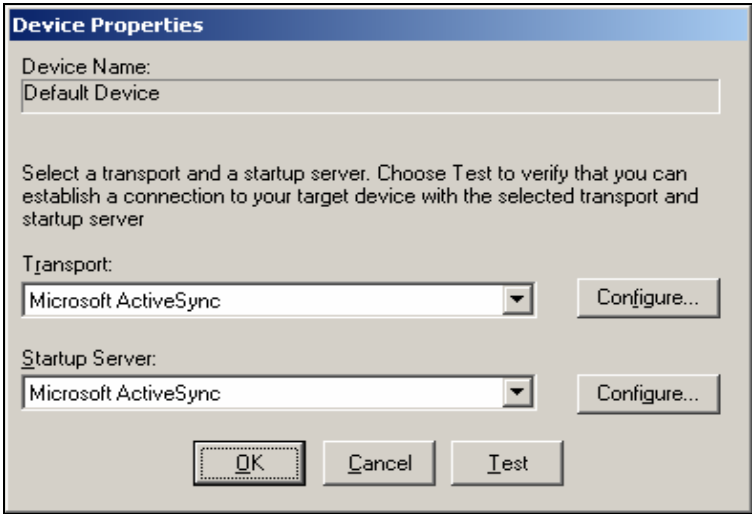
Launching the Application Verifier Tool from the Windows CE 5.0 Test Kit and Setting It Up to Run the Test Application

Scenario

In this exercise you will launch the Application Verifier Tool, and then set it up to run the test application.

Note: The Application Verifier Tool is shipped as part of the Windows CE 5.0 Test Kit. You can also download the Application Verifier Tool by itself and run it without the Test Kit. However, for the stand-alone Application Verifier Tool to work, it may require some connection components that ship with the Test Kit or Windows CE 5.0 (the Platform Builder toolset). For this lab, you will launch the Application Verifier Tool from the Test Kit.

Tasks	Detailed Steps
1. Launch the Application Verifier Tool	a. In the Windows CE Test Kit window, right-click Pocket_PC (ARMV4I), and then click Tools Application Verifier, as shown in Figure 16.  The screenshot shows the 'Windows CE Test Kit' application window. The menu bar includes 'Server', 'Connection', 'View', 'Tests', and 'Help'. The main pane displays a tree view with 'Windows CE Test Kit Server' expanded, showing a sub-item 'Pocket_PC (ARMV4I)'. A right-click context menu is open over 'Pocket_PC (ARMV4I)', with options: 'Expand', 'Redetect Peripherals', 'Remove Device', 'Device-Side Files...', 'Tools', and 'Application Verifier'. The 'Tools' menu item is highlighted, and its sub-menu is open, showing 'Application Verifier', 'CPU Monitor', 'Resource Consumer', and 'Windows CE Stress'. The 'Application Verifier' option is selected. At the bottom of the window, a status bar reads 'Starts the selected tool.' Figure 16. Launching Application Verifier b. Wait a few moments as the Application Verifier Tool opens. Note: The Application Verifier Tool attempts to initiate a connection to the emulator. The Device Connection dialog box appears.

Tasks	Detailed Steps
	c. Click Connect to connect to the Application Verifier Tool to the emulator. d. In the Select a Windows CE Device dialog box, select Default Device, and then click OK. e. Wait (this could take up to a minute) as the connection is made. <i>Note: After the connection is made, the Device Connection dialog box automatically closes, and the Application Verifier Tool appears, as shown in Figure 17.</i>  Figure 17. The Application Verifier Tool f. Do one of the following: - If the Application Verifier Tool appears, minimize the Windows CE Test Kit window. Proceed to the next task: Load the test application in the Application Verifier Tool. - If the Application Verifier Tool does not appear, repeat this procedure from Step a. When you get to Step c, click the Settings button, select the Default Device, and then click the Properties button. Be sure that the Transport and Startup Server list values are both set to Microsoft ActiveSync, as shown in Figure 18, and then click OK to save the settings. On the Windows CE Platform Manager Configuration dialog box, click OK, and then continue with Step d. 

Tasks	Detailed Steps
	Figure 18. Device Properties dialog box
2. Load the test application in the Application Verifier Tool	a. In the Application Verifier Tool, click Add. This step opens the Add Application dialog box. b. Select the test application, Leakdemo.exe, as shown in Figure 19, and then click Open. Figure 19. Adding the test application to the Application Verifier Tool c. The Application Verifier Tool now shows leakdemo.exe in the list of applications to monitor. d. In the Test Settings box, select Heap Verifier, Handle Leak Tracker, and Shell Verifier, as shown in Figure 20.

Tasks	Detailed Steps
	Figure 20. Test application loaded in Application Verifier Note: You are now ready to run the application on the emulator. Note: The following information may be useful when you are using the Application Verifier Tool outside this Virtual Lab. For the best results, enable as many test settings as possible. Additional test settings are available via the Options button. Note that the test settings are stored in the registry of the device, so they persist until you perform a clean reboot. You can monitor more than one application at the same time. Simply add all of the applications that you are interested in by clicking the Add button. To stop monitoring an application, remove it from the list of applications by clicking the Remove button.

Exercise 5

Running the Test Application and Obtaining the Logged Data

Scenario

In this exercise, you will run the test application on the emulator, and then obtain the logged data.

Tasks	Detailed Steps
1. Run the test application on the emulator	a. In the Application Verifier Tool, click Run. <i>Note: When running the Application Verifier Tool outside this lab, in addition to using the Run button, you can start an application directly from the device. For example, you can start an application from a file browser, a shortcut, by clicking Start Programs (for Pocket PC), or by clicking Start (for Smartphone). The Run button simply causes a CreateProcess("Application Name") call, which only works if your application is on the PATH. Also, remember to load your applications into the Application Verifier Tool (as explained in the "Load the test application in the Application Verifier Tool" task), and then choose your test settings before you begin to run the application. If your application is already loaded into memory before you choose the Application Verifier Tool settings, those settings will not take effect until the next time the application is reloaded into memory.</i> b. On the Windows taskbar, restore the Pocket PC Phone Emulator window. c. Wait until the test application is running in the foreground of the emulator with the title Minimal Windows App. Allow enough time (about 45 seconds for this lab) for the application to be fully loaded. d. Click the X button at the top right of the test application once to close the application. e. Wait for about 15 seconds as the logs are generated. <i>Note: The Application Verifier Tool automatically generates logs on process exit. However, it is still possible to get the logs at other times too. The logs are useful when you are testing drivers or other modules that don't get unloaded. On Windows Mobile 5.0-based Pocket PCs, you can find out which programs are running from the memory control panel in the Settings control panel (Click Start Settings. Click the System tab, and then click Memory. Click the Running Programs tab). From this location, you can also stop running applications.</i> f. Minimize the emulator's window.
2. Transfer the logged data from the emulator onto your development workstation	a. In the Application Verifier Tool, click the Get Logs button. This step opens the File Browser. b. Browse to the Desktop Logs folder, and then click OK. c. Wait until the Get Logs message appears, which notifies you that the logs have been successfully copied. d. Click OK to close the message box. <i>Note: Although the Application Verifier Tool stores the logged data in text files, it is much more convenient to use the viewer on the Application Verifier Tool. The log files are named according to the following convention: AppVerifier_ProcessName_ProcessIdentifier.log.</i>
3. View the log files by using the Application	a. In the Application Verifier Tool, click the View Exported Log button. This step opens the Windows file browser.

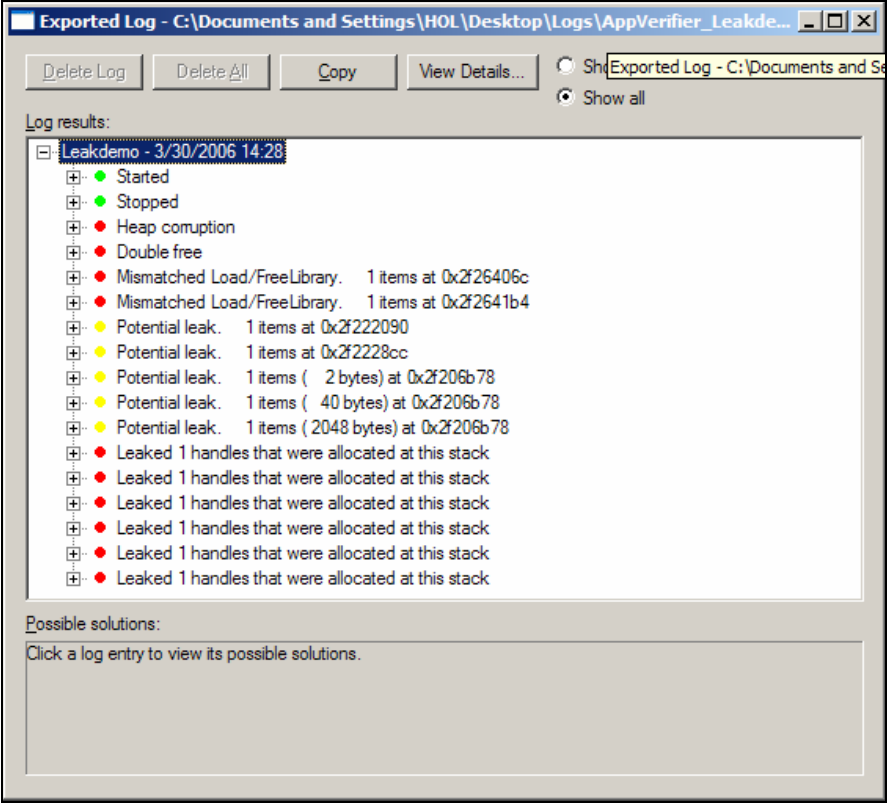
Tasks	Detailed Steps
Verifier log viewer	b. Browse to the Desktop Logs folder, and then select the log file you previously transferred. <i>Note: Placing your cursor over a file in the file browser tells you the modification date and time. You can use this information to distinguish between multiple log files in the same directory.</i> c. Click Open to open the log files. This step opens the Browse for Folder dialog box, where you can select the folder that contains the .map file that corresponds to the test application, as shown in Figure 21. Figure 21. Browse to the folder that contains the .map files d. Select Desktop Lab Program Files, and then click OK. <i>Note: When monitoring an application outside this lab, you may get a message that says, “Unresolved callstacks” while opening your logged data. If the application has corresponding .map files, use the file browser that appears to select the folder that contains the .map files. If you have no such files, click OK to continue. The .map files contain extra debug information, such as function names, that can help you track potential resource leaks faster and better. To generate map files when developing an application with Visual Studio 2005, you may have to set the appropriate Linker options. To do so, right-click the project in Solution Explorer, and then click Properties. Under Configuration Properties, click Linker Debugging, and then set the Generate Map File option to Yes.</i>

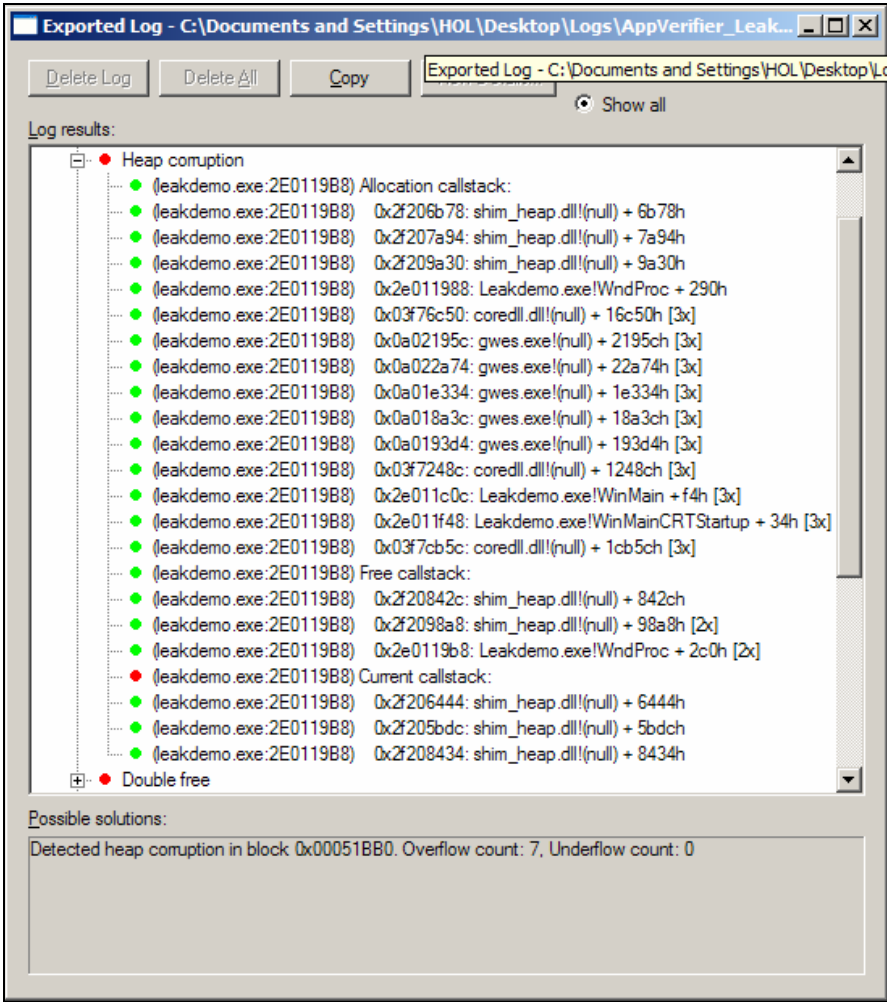
Exercise 6

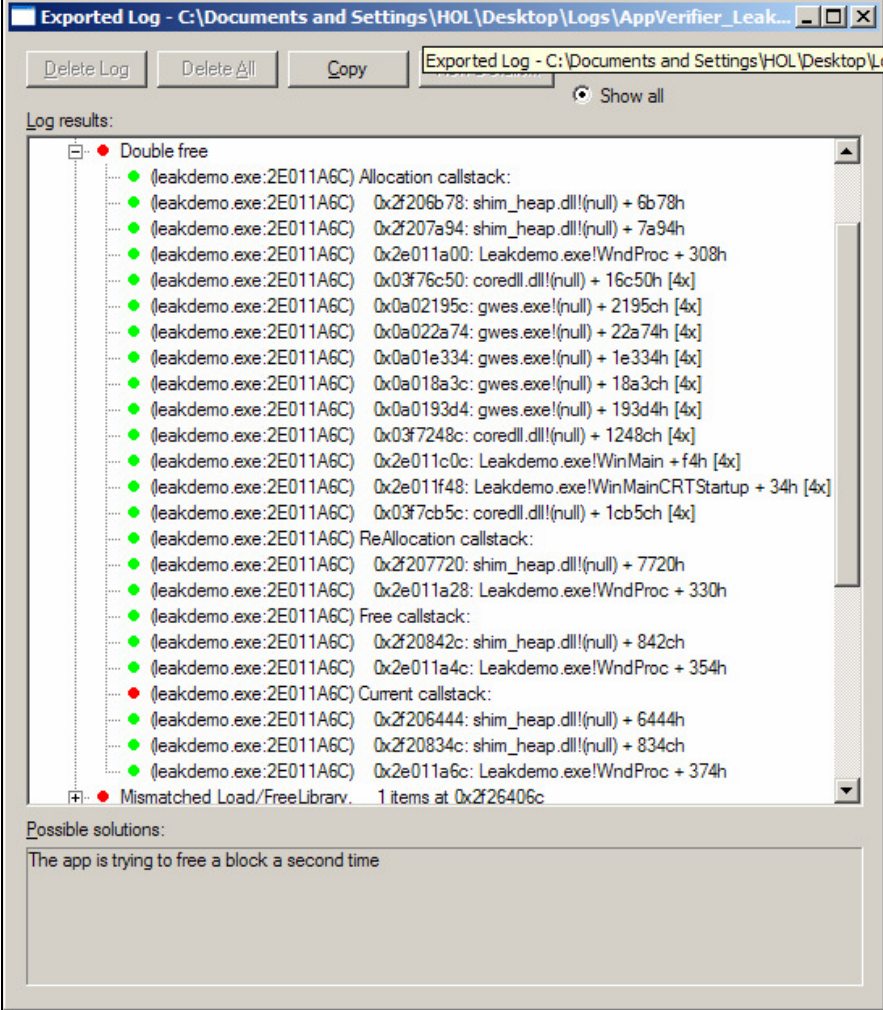
Analyzing the Logged Data with the Application Verifier Tool

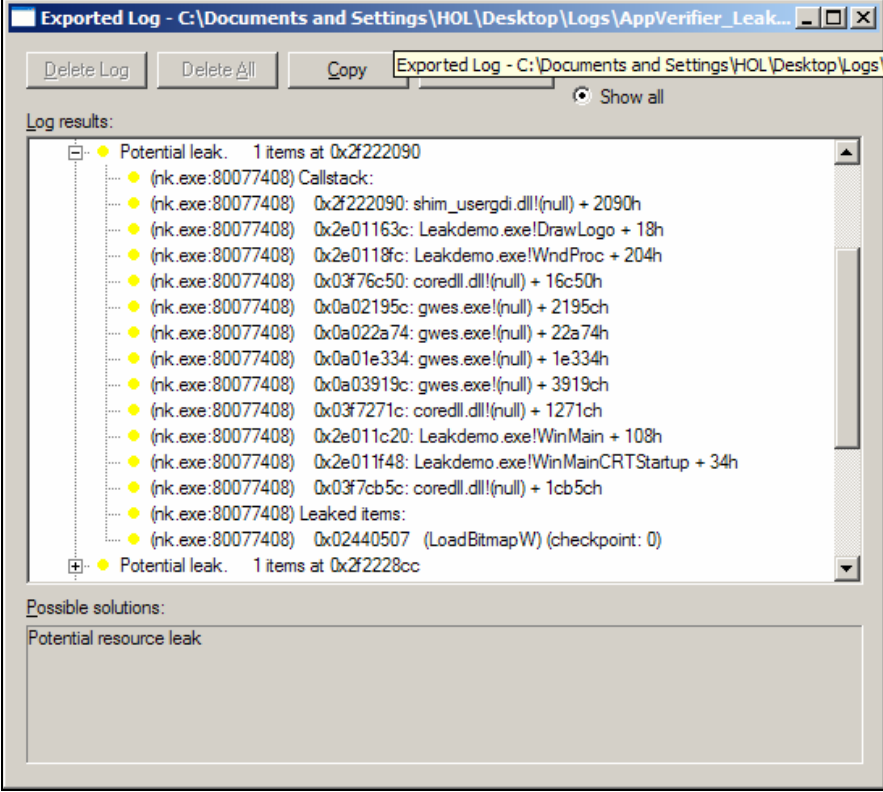
Scenario

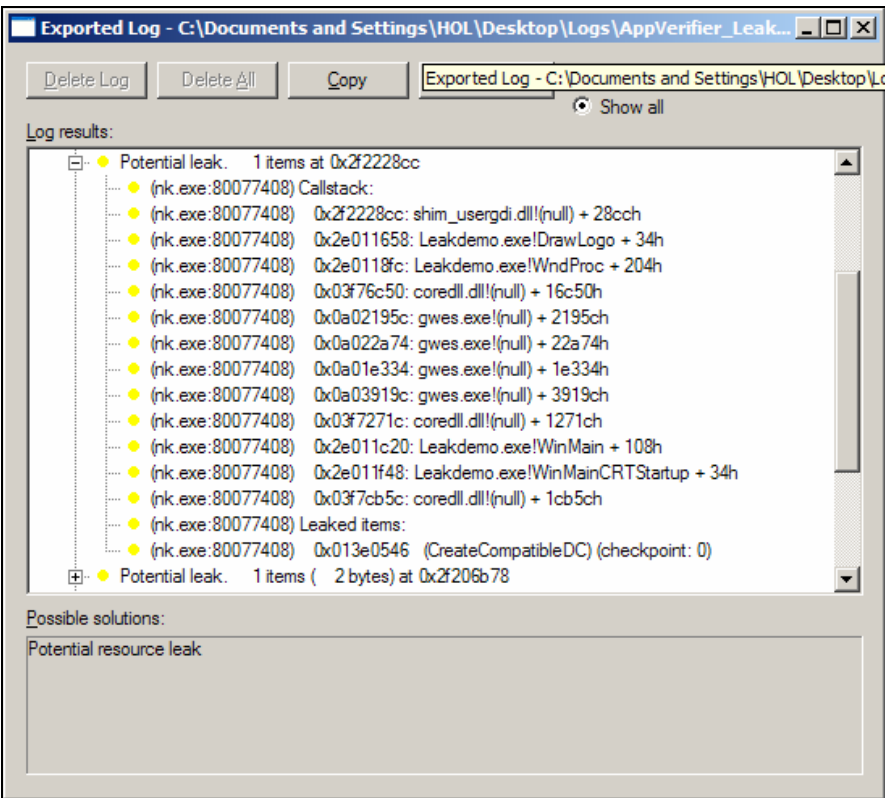
In this exercise, you will analyze the logged data by using the Application Verifier Tool.

Tasks	Detailed Steps
1. Analyze your data	a. Expand Leakdemo in the viewer. The data displays in the format as shown in Figure 22.  Figure 22. Logged data b. Browse to various sections to investigate potential problems. Note: The Application Verifier Tool collects data that you can use as a guide when troubleshooting resource leaks and memory corruptions. The following is brief breakdown of each section of the log: The Started section tells you when the application was started. The Stopped section tells you when the application was stopped. The Heap corruption section reports potential heap corruption problems in your application. The Double free section reports potential double freeing of allocated memory. The Mismatched Load/FreeLibrary section reports LoadLibrary calls that do not

Tasks	Detailed Steps
	have corresponding FreeLibrary calls. These alerts may be the result of intended delay-loads for dynamic-linked libraries. The Potential leak section indicates possible memory leaks. The Leaked handles sections refer to resource handles that were possibly leaked c. Expand each section to reveal the call stack for the potential problem. Note: For example, expanding the Heap corruption bullet shows the stack trace for a potential heap corruption that occurs in the WndProc function of Leakdemo.exe, as shown in Figure 23. You can resize the viewer as needed. Notice the Allocation, Free, and Current call stacks leading to this  Figure 23. Heap Corruption log results d. Expanding the Double free bullet illustrates a double free in the WndProc function of Leakdemo.exe. The Allocation, Reallocation, Free, and Current call stacks leading to this are shown in Figure 24.

Tasks	Detailed Steps
	 Figure 24. Double free log results e. Next, drill down into a Potential leak section, as shown in Figure 25.

Tasks	Detailed Steps
	 Figure 25. Memory leak in DrawLogo - LoadBitmap f. This stack trace shows a memory leak. It identifies the leaked item that results from a LoadBitmap call in the function DrawLogo (the second line in call stack). The following code is the relevant section of the source code for Leakdemo.exe. <pre> void DrawLogo (PAINTSTRUCT *ps) { HBITMAP hBmp; HBITMAP hBmpOld; BITMAP bmp; HDC hDCTemp; int iXPos; int iYPos; hBmp = LoadBitmap (ghInstance, MAKEINTRESOURCE(IDB_BITMAP)); hDCTemp = CreateCompatibleDC (ps->hdc); GetObject (hBmp, sizeof (bmp), & bmp); hBmpOld = (HBITMAP) SelectObject (hDCTemp, hBmp); iXPos = (ps->rcPaint.right-bmp.bmWidth) / 2; iYPos = (ps->rcPaint.bottom-bmp.bmHeight) / 2; BitBlt (ps->hdc, iXPos, iYPos, bmp.bmWidth, bmp.bmHeight, hDCTemp, 0, 0, SRCCOPY); </pre>

Tasks	Detailed Steps
	<pre data-bbox="544 220 1209 346"> // BUG - Most common GDI programming error // DeleteDC (hDCTemp); // DeleteObject (hBmp); } </pre> g. In the previous code, you did not delete the bitmap handle returned by the call to LoadBitmap after using it. This is responsible for the above leaked item, as shown in the log (Figure 25). h. The same block of code is responsible for the next leak violation identified by the Application Verifier Tool, as shown in Figure 26. Notice that, in the DrawLogo function, you did not delete the memory device context handle returned by the call to CreateCompatibleDC after using it.  Figure 26. Memory leak in DrawLogo – CreateCompatibleDC i. The Application Verifier Tool identifies the leak that results from the CreateCompatibleDC call in the DrawLogo function (the second line in the call stack). Note: You can copy the call stack for an identified potential problem onto the Clipboard. To do so, select a bullet, and then click the Copy button in the viewer.

Summary

In this lab, you performed the following exercises:

- Launching ActiveSync and the Windows Mobile 5.0 Pocket PC Phone Emulator
- Copying the required program files from the desktop computer to the emulator by using ActiveSync
- Launching and setting up the Windows CE 5.0 Test Kit
- Launching the Application Verifier Tool from the Windows CE 5.0 Test Kit and setting it up to run the test application
- Running the test application and obtaining the logged data
- Analyzing the logged data with the Application Verifier Tool