Hands-on Lab

Windows Mobile Device Testing

***Abstract*:** This lab session will go through several different exercises to show you how to use the Windows Hardware Lab Kit (Windows HLK) to perform camera tuning and touch panel tests. The Windows HLK provides test collateral that you can use to validate and assess the ship readiness of your drivers and includes an infrastructure that is designed to meet the at-scale needs of a lab environment. You will learn how to use this framework to schedule tests and view results. For camera tuning, you will use Qualcomm tools to check the camera’s picture quality and adjust parameters to provide better use experiences. Touch is a critical interaction between devices and people. The touch panel test robot will use a Qualcomm template and scripts to imitate different touch operations to test the quality of a touch screen, including touch accuracy, immunity, and response time.

Copyright (c) 2015 Microsoft. All rights reserved.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Contents

[**Lab objective** 3](#_Toc445393333)

[Lab setup 3](#_Toc445393334)

[**Exercise 1: Windows HLK hands-on** 4](#_Toc445393335)

[Section 1: Install the Windows HLK Controller (Done in advance) 4](#_Toc445393336)

[Local path \\172.24.220.210\ak-hlk-mtbf\TH2\_RELEASE.10586.13053.20151029-1700\_MTBF 5](#_Toc445393337)

[Section 2: Install the Windows HLK client on Windows Phone 5](#_Toc445393338)

[Section 3: Create a machine pool 6](#_Toc445393339)

[Section 4: Create a new project 7](#_Toc445393340)

[Section 5: Select the phone to test 8](#_Toc445393341)

[Section 6: Schedule jobs 8](#_Toc445393342)

[Section 7: View results 8](#_Toc445393343)

[Section 8: Investigate failures 9](#_Toc445393344)

[Section 9: Package results 9](#_Toc445393345)

[Section 10: Clean up 10](#_Toc445393346)

[**Appendix A. Installing Windows HLK** 11](#_Toc445393347)

**Lab objective**

This self-paced lab will cover a few different exercises for Windows Phone testing: installing the Windows Hardware Lab Kit (Windows HLK) for a mobile device, scheduling camera tuning and touch panel tests, running them on your device, and most importantly, analyzing the results and investigating failures.

# Lab setup

This lab includes three different tests. Each test will use different hardware and software.

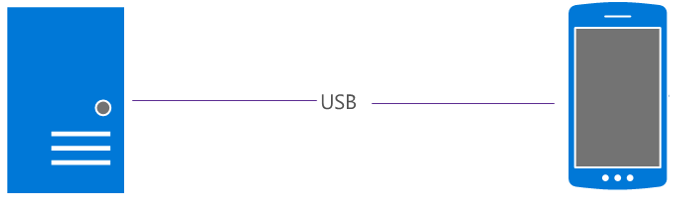
The mobile test environment requires a minimum of a **test server** and a **phone.** The test server must be a machine capable of running Windows Server 2008 R2 or higher, while the test device must be a Windows Phone. In this lab, we will use a direct USB connection.

**Test Server**

Windows Server 2012 R2

**Test Device**

Windows 10 Technical Preview for phones Client



Camera tuning will mainly use chromatix series tools to do tuning. The environment has been setup in advance because of time restrictions.

The touch panel test will use a robot test system. We only have one system, which is already set up for the demo.

The following table lists the software and hardware requirements for this lab:

|  |  |  |
| --- | --- | --- |
| Exercises | Hardware | Software |
| Windows HLK | PC with network with DHCP server  Windows Phone Device  USB to Micro-USB cable | Windows Server 2012 R2  Windows 10 Technical Preview for phones  Windows HLK Controller & Windows HLK studio  TShell & virth ethernet |
| Camera tuning | Light booth Qualcomm recommended, Light meter, ND filter, Macbeth Color Checker,18% gray chart, Diffuser filter , Q14 gray step card, ISO12233 chart, Laser range finder, Oscilloscope | Qualcomm chromatix 6 software, Qualcomm chromatix light software, Qualcomm Qtes image test software , Imatest image test software |
| Touch Panel | DTR3-2210-T-SIG Robot \* 1  WP QRD Device with test image \* 2  USB to Micro-USB cable. \* 2  USB Serial Converter \*1  Copper bar (Size: 5, 6, 7, 8, 9MM)  Small sprinkler \*1  Oscilloscope \*1  Ruler \*1 | DaWin\_Pro(Eng\_Ver1.4.1).exe  QC Robot test scripts:  MotProgram(CAL.PGM, JIT.PGM, LIN.PGM, MOI.PGM…) & PointFile(\*\*.PNT)  QC CTP Robot test template:  Test-Report-CTP-Template-20131009.xlsm  WP8\_AutoTestTools\_Release\_AnyCPU.xap  Isetool.exe\_x64 |

**Exercise 1: Windows HLK hands-on**

Section 1: Install the Windows HLK Controller (Done in advance)

Please refer **to Appendix A**

**Download URL:** [**http://pan.baidu.com/s/1kUgebaB**](http://pan.baidu.com/s/1kUgebaB)

Local path [\\172.24.220.210\ak-hlk-mtbf\HLK\_10586.0.151029-1700.zip](\\\\172.24.220.210\\ak-hlk-mtbf\\HLK_10586.0.151029-1700.zip)

Section 2: Install the Windows HLK client on Windows Phone

In this section, you will learn how to install the Windows HLK client on a Windows Phone device. The Windows HLK client is responsible for communication between the test system and the Windows HLK Controller, and for executing tests.

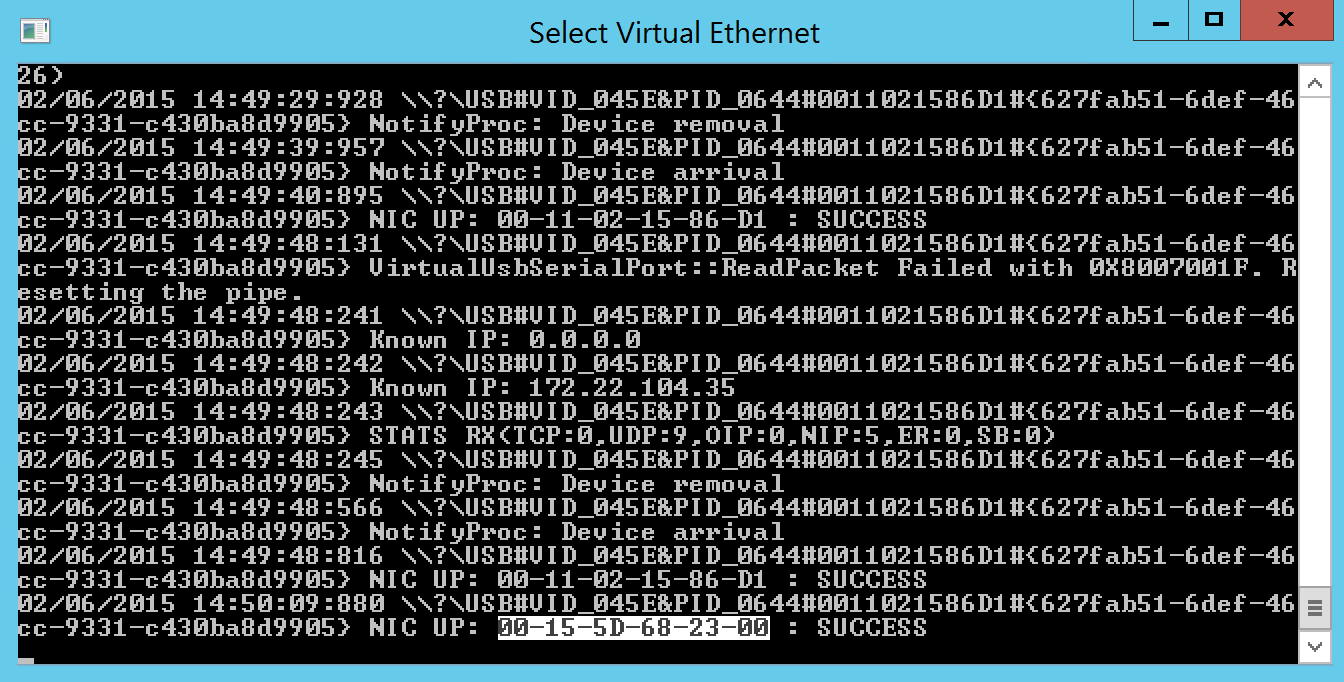
Step 1: Establish network connectivity to the phone

1. Launch the **Virtual Ethernet** shortcut from the desktop.

Virtual Ethernet download URL: <http://pan.baidu.com/s/1jGjM2fO>

Local path [\\172.24.220.210\share\jrd-tool\microsoft-win8.1\14219\OEM\BlueGDR1\_9651.14219\_OEM\BlueGDR1\_9651.14219\_OEM\IHV\_Tools](file:///\\172.24.220.210\share\jrd-tool\microsoft-win8.1\14219\OEM\BlueGDR1_9651.14219_OEM\BlueGDR1_9651.14219_OEM\IHV_Tools)

1. Connect the phone to the PC by using the USB cable. Look for an indication that the phone is connected. There should be a **MAC address** in the output. See the image below for example output:



Step 2: Connect to the phone via TShell

1. Launch the **TShell** shortcut as an **Administrator** from the **desktop**.
2. Connect to the phone by entering the following command at the **TShell** command prompt:

PS> ***Open-Device*** *<mac address>*

Note:

You can get the < *mac address> from Step 1. In this sample, it’s “00-15-5D-68-23-00”.*

1. Verify your connection to the phone by entering the **ipconfig** command at the **TShell** command prompt. Make sure that the connected phone has a valid IP address.

Example:

|  |
| --- |
| **​**PS C:\Users\Administrator\Desktop> ***cmdd ipconfig /all*** |

Step 3: Install the Windows HLK Client

*Note: The following steps will work only for devices that are in the same workgroup as the controller. Generally, the default workgroup for phones is WORKGROUP; therefore, your controller must be joined to WORKGROUP instead of being domain-joined.*

1. Find the name of the Windows HLK Controller by entering the **hostname** command at the TShell command prompt.
2. Create a connection to the Windows HLK install share on the controller by entering the following command:

execd net use \\< *ControllerName* >\hlkinstall <controller password> /u:<controller domain>\<controller user> /Persistent:yes

<ControllerName>: Computer name

<controller password>: Login user password

<controller domain>: Computer domain. Windows phone default domain is workgroup, so computer do not need to join the domain, it is Member of Workgroup.

<controller user>: Login username

1. Run the install script from the install share by entering the following command:

cmdd \\<*ControllerName*>\hlkinstall\coresystemclient\installwttclient.cmd

Note: You must use a different command-line to install on certain 3G phones that have limited space on the main OS partition.

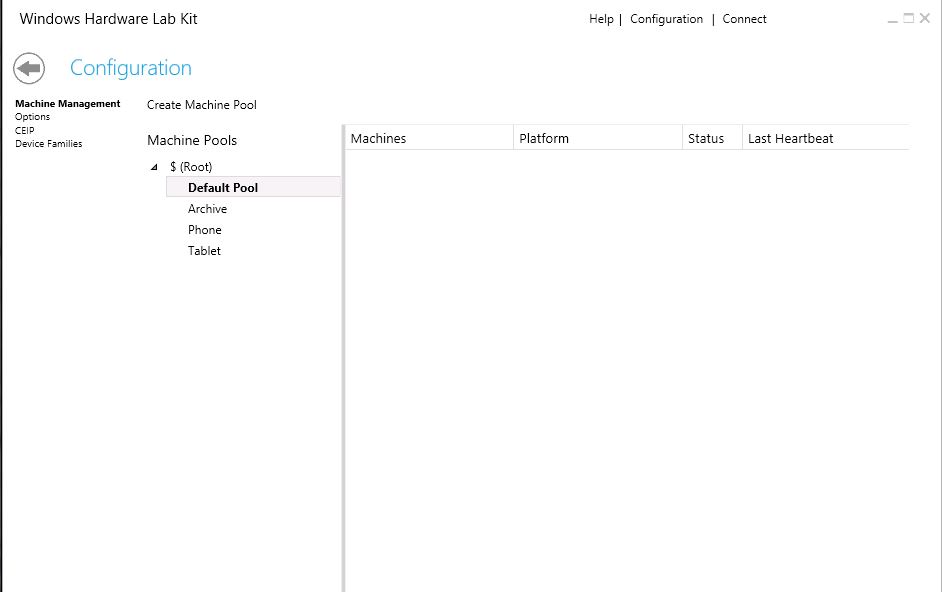
cmdd \\< *ControllerName* >\hlkinstall\coresystemclient\installwttclient.cmd JWDDrive:<Drive|D|E>

1. Enter the following command:

|  |
| --- |
| **​** cmdd net use \* /d /y |

Section 3: Create a machine pool

In this exercise, you will place the test phone into a logical group called a machine pool. Launch Windows HLK Studio by searching for it in the start menu. Click on **Configuration** in the top right corner:



Separate your test device from other mobile systems by using separate machine pools:

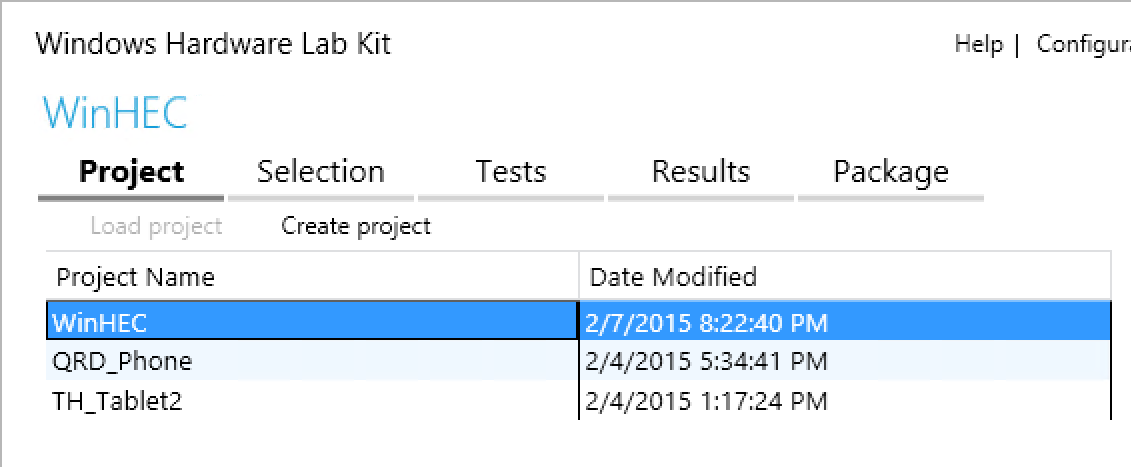
1. Select the **$(Root)** node.
2. Click the **Create Machine Pool** button to create a new machine pool, and name this new pool ‘**Phone**’.
3. Drag the test phone into this new ‘Phone’ pool.
4. Select the ‘Phone’ pool and verify that the phone that is running Windows 10 is present.
5. Right-click the device, and change the state from **Not Ready** to **Ready**.

Section 4: Create a new project

In this exercise, you will learn how to create a new project and select tests.

1. On the **Project** tab, click **Create project**.
2. Rename the project to “WinHEC”.
3. Press **Enter** to create the project.

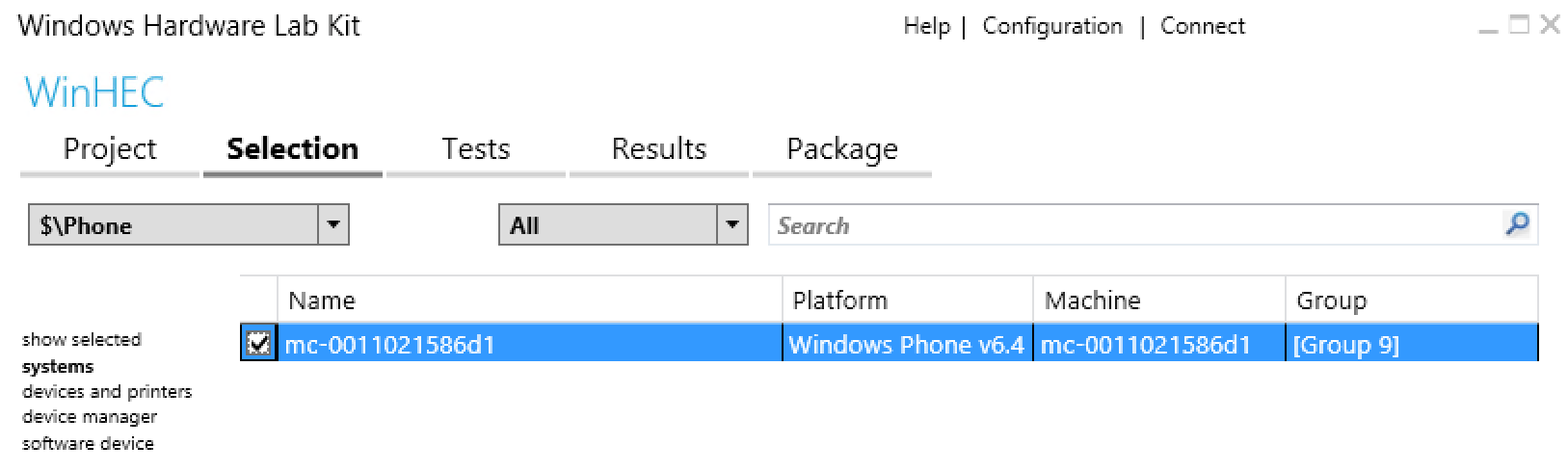
When the project name appears on the page, the project is created.



Section 5: Select the phone to test

With the ‘WinHEC’ Project selected, click the **Selection** tab. In this lab, we will be running tests against the entire phone system.

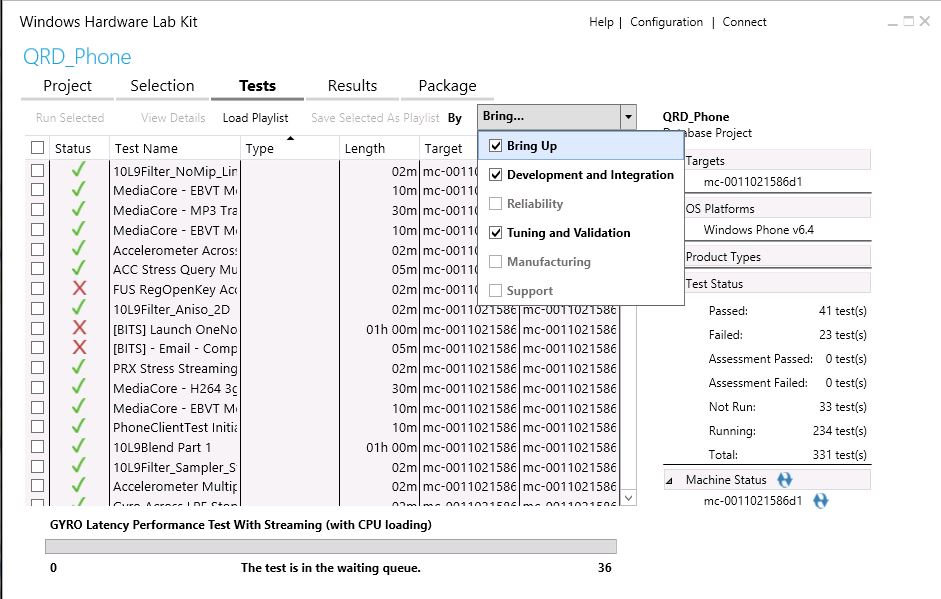
1. Click **Selection** to go to the Selection tab.
2. From the machine pool list (upper left drop-down list), select the Phone pool that you created.
3. From the left pane, select the **systems** view, and check the box next to the device.



Section 6: Schedule jobs

In this exercise, you will learn to schedule and run test jobs.

1. Click on the **Tests** tab, and notice that tests have been automatically added to the project.
2. Check the boxes next to the tests that you want to run in the window (as above Section 6).
3. Click **Run Selected** to schedule and run these test jobs.



A progress bar appears. A slight delay occurs when you run a test. Before the first test is executed, the client machine is provisioned for testing. This is a one-time event that will take 3-5 minutes.

1. Wait for all tests to complete.

Section 7: View results

In this exercise, you will learn how to view test results and diagnostic logs as you debug a test failure.

To view test results, click on the **Results** tab. The **Results** tab displays detailed information about each test. As each test completes, the status column updates with the result – pass or fail. A green checkmark means that it passed; a red X means that it failed. The right pane displays project summary information, including the target(s) selected, operating systems being tested, product types you qualify for, and the status of all tests.

Section 8: Investigate failures

Let’s learn more about why a test failed.

1. From the list, select a failed test (let’s call it test C). The red X indicates that the test has failed.
2. Expand the **Test Name** node, expand the **Logs** node, and then double-click the log file.
3. View the test commands.

Section 9: Package results

In this exercise, you will learn how to package test logs for debugging scenarios.

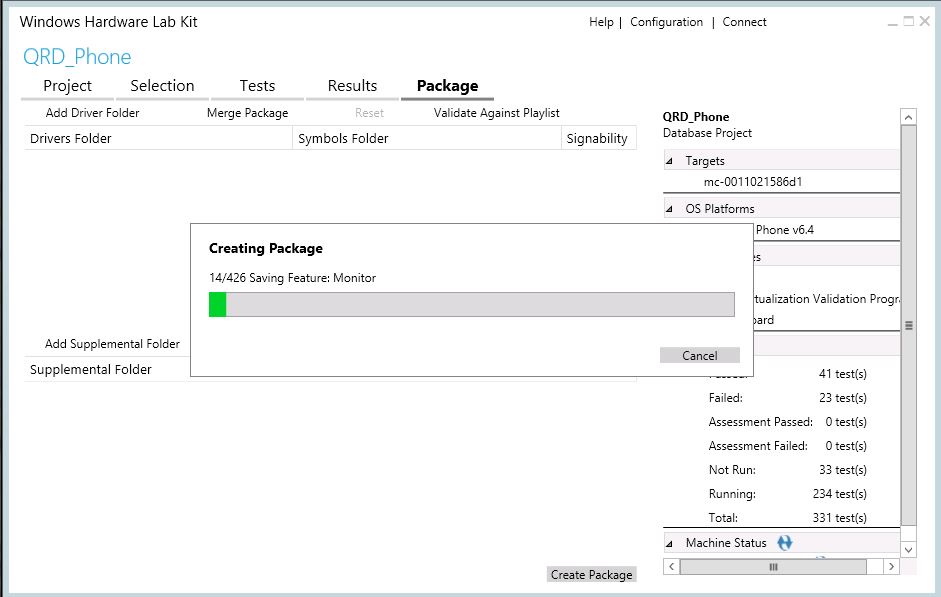
In a debugging scenario where a driver issue is discovered, test logs might be useful for the driver developer. The best way to gather the test logs and information about the test environment is to use a partial package. To create a partial package, follow these steps:

1. On the **Results** tab, locate the failed test and expand the test node.
2. Right-click the result node and select **Create Partial Package**.
3. Choose the Desktop as the package output location.

You have now created a partial package. The partial package contains the failed test result and all associated logs. Additionally, the partial package saves the details of the associated project.

You can also create a full Windows HLK test package:

1. Click the **Package** tab.
2. Click **Create Package**.



Section 10: Clean up

After testing, you will often need to clean up the test results and temporary data. Follow these steps to reset the phone and delete the config files from Windows HLK Studio.

1. Reset the test phone.

Tap **All settings** -> **About** -> **Reset Phone** in the phone, and then wait for 5-10 minutes for the phone to reset.

1. Delete the phone from the machine pool.
2. Delete the project from Windows HLK Studio.

**Appendix A. Installing Windows HLK**

Windows HLK can be installed on Windows Server 2008 R2, Windows Server 2012, or Windows Server 2012 R2. Windows HLK is comprised of two components, the Windows HLK Controller and Windows HLK Studio. Windows HLK Controller is the software engine that manages tests that are run on the test machines. Windows HLK Studio is the management UI tool that lets you select and run tests against the test machines that are connected to the server.

To set up the test server, follow these steps.

1. Download Windows HLK from the Windows Developer Center.
2. From the download location, click **Download > Now**.
3. When prompted, click **Run.**
4. When the **Specify Location** screen appears, select appropriate option:
   1. Install option – Select **Install the Windows HLK to this computer**, and then click **Install**.
   2. Download option – Select **Download Windows HLK for installation on a separate computer,** and then click **Next**.
5. Select the **Controller + Studio** option. If you are installing Windows HLK directly, you must open a port on your server. Select **Yes** to open the port.
6. When the **Join the Customer Experience Improvement Program (CEIP)** screen appears, select **Yes** or **No,** and then click **Next.**
7. Review the License Agreement, and then click **Accept** to proceed.
8. If you selected the install option, the installation takes about 45 minutes. If the Microsoft .NET Framework 4.5 isn't already installed on the computer, follow the prompts to install it. After the computer restarts, you must repeat the installation instructions from **Step 1** to install Windows HLK on this computer.

If you selected the download option, copy your download to your test server. Run HLKSetup.exe and repeat the installation instructions from **Step 3** to install Windows HLK on this computer.