

# Windows Mobile 6 Wireless Tests ("Before" Example)

---

## Introduction

The following content extract has been reproduced from the Windows Mobile 6 Developer Documentation compiled help file (.CHM) originally released to customers in November, 2006.

## Table of Contents

Introduction .....	1
Table of Contents .....	1
Table of Contents -- Windows Mobile 6 RTM Content.....	1
Wireless (802.11) Tests .....	1
<i>One-Card WLAN Card Miniport Driver Test .....</i>	<i>2</i>
Prerequisites for the One-Card WLAN Card Miniport Driver Test.....	2
Setting Up and Configuring the One-Card WLAN Card Miniport Driver Test.....	3
Test Cases for the One-Card WLAN Card Miniport Driver Test .....	5
Command Line Parameters for the One-Card WLAN Card Miniport Driver Test .....	8
<i>Two-Card WLAN Card Miniport Driver Test .....</i>	<i>8</i>
Prerequisites for the Two-Card WLAN Card Miniport Driver Test .....	9
Setting Up and Configuring the Two-Card WLAN Card Miniport Driver Test.....	10
Connecting the Access Points and Devices.....	11
Test Cases for the Two-Card WLAN Card Miniport Driver Test .....	12
Command Line Parameters for the Two-Card WLAN Card Miniport Driver Test .....	14

## Table of Contents As Delivered--Windows Mobile 6 RTM Content

The following screen-shot shows the relevant portion of the Windows Mobile 6 Table of Contents as it appears in the compiled help file, originally released to customers in November 2006.

- [-] Wireless (802.11) Tests
  - [-] One-Card WLAN Card Miniport Driver Test
    - [+] Prerequisites for the One-Card WLAN Card Miniport Driver Test
    - [+] Setting Up and Configuring the One-Card WLAN Card Miniport Driver Test
    - [+] Test Cases for the One-Card WLAN Card Miniport Driver Test
    - [+] Command Line Parameters for the One-Card WLAN Card Miniport Driver Test
  - [-] Two-Card WLAN Card Miniport Driver Test
    - [+] Prerequisites for the Two-Card WLAN Card Miniport Driver Test
    - [+] Setting Up and Configuring the Two-Card WLAN Card Miniport Driver Test
    - [+] Test Cases for the Two-Card WLAN Card Miniport Driver Test

## Wireless (802.11) Tests

11/15/2006

The tests in this section assess the functionality of the miniport driver for wireless LAN cards in various configurations.

## IN THIS SECTION

### [One-Card WLAN Card Miniport Driver Test](#)

Assesses the functionality of a miniport driver for a single wireless LAN card, and verifies that the driver supports Network Driver Interface Specification functionality

### [One-Card WLAN Card Miniport Driver Test](#)

#### Two-Card WLAN Card Miniport Driver Test

Assesses the functionality of a wireless miniport driver on a target device with respect to interaction with another wireless card

## SEE ALSO

### Other Resources

CETK Tests

## One-Card WLAN Card Miniport Driver Test

The One-Card WLAN Card Miniport Driver Test assesses the functionality of a miniport driver for a single wireless LAN card. It also verifies that the driver supports Network Driver Interface Specification (NDIS) functionality. The operation of this test is very similar to the *One-Card Network Card Miniport Driver Test*, which tests functionality common to all network cards. This test has additional wireless-specific test setup requirements.

## IN THIS SECTION

### [Prerequisites for the One-Card WLAN Card Miniport Driver Test](#)

Specifies the hardware and software requirements for this test

### [One-Card WLAN Card Miniport Driver Test](#)

### [Setting Up and Configuring the One-Card WLAN Card Miniport Driver Test](#)

Describes the setup and configuration of this test

### [One-Card WLAN Card Miniport Driver Test](#)

### [Test Cases for the One-Card WLAN Card Miniport Driver Test](#)

Describes the test cases for this test

### [One-Card WLAN Card Miniport Driver Test](#)

### [Command Line Parameters for the One-Card WLAN Card Miniport Driver Test](#)

Describes the command line parameters for this test

## SEE ALSO

### Other Resources

## Prerequisites for the One-Card WLAN Card Miniport Driver Test

The following table shows the hardware requirements for the One-Card WLAN Card Miniport Driver Test.

Requirement	Description
A Windows Embedded CE-based device with a wireless network card and another network card	Device containing the wireless (802.11) network card for which the driver is being tested. A second card is needed to establish a TCP/IP connection to the development workstation.
Access points	Used for the test; must be configured based on the network types

(802.11 a, b, g, a/b, or a/g) and on whether the card supports WPA and/or WPA2. Setup and configuration for these is found in [One-Card WLAN Card Miniport Driver Test Setting Up and Configuring the One-Card WLAN Card Miniport Driver Test](#)

This test is comprised of two binaries, Ndt.dll and Ndt\_1c\_wlan.dll. The Ndt.dll binary file is a protocol driver that binds to the test and support cards. The protocol driver communicates with the underlying miniport drivers through an NDIS wrapper and registers as a stream driver. The Ndt\_1c\_wlan.dll binary file controls the test itself. The following table shows the software requirements for the One-Card WLAN Card Miniport Driver Test.

Requirement	Description
Tux.exe	Test harness, required for test execution
Kato.dll	Logging engine, required for test execution
Ndt.dll	Protocol driver for the test
Ndt_1c_wlan.dll	Test library
DummyWzcsvc.cab	Must be installed on all retail devices to replace the existing wzcsvc with a dummy. This enables the test to take complete control of the wireless cards without interference from wzcsvc.

**Note:**

The miniport driver being tested should be recognized by NDIS. You should assign a name to the network card in order for the test to function correctly.

**SEE ALSO**

**Other Resources**

[One-Card WLAN Card Miniport Driver Test](#)

**Setting Up and Configuring the One-Card WLAN Card Miniport Driver Test**

The following table shows which access point types are required for each network type

Test card network type	NDTEST_WEP_AP1	NDTEST_WEP_AP2	NDTEST_WEP_AP3	NDTEST_WPA_AP1	NDTEST_WPA2_AP1
802.11a	802.11a	802.11a	-	802.11a	802.11a
802.11b	802.11b	802.11b	-	802.11b	802.11b
802.11g	802.11b	802.11g	-	802.11g	802.11g
802.11a/b	802.11b	802.11a	-	802.11a	802.11a
802.11a/g	802.11b	802.11a	802.11g	802.11g	802.11g

The access points for the One-Card WLAN Card Miniport Driver Test are selected based on the driver needs. Wired equivalent privacy (WEP)-only access points must be selected as a minimum; cards that support WPA or WPA2 must select additional access points.

**CONFIGURING THE ACCESS POINTS**

Configure each access point selected from the table above according to the following information.

To configure NDTEST\_WEP\_AP1

Set the SSID to **NDTEST\_WEP\_AP1**.

Set the Wired equivalent privacy (WEP) status to **enabled**.

Set the WEP key at index 1 to **0123456789**.

Set the WEP key at index 2 to **9876543210**.

Set the WEP key at index 3 to **1020304050**.

Set the WEP key at index 4 to **5040302010**.

Set the active WEP index to **1**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **enabled**. This is the only access point which will have DHCP enabled.

To configure NDTEST\_WEP\_AP2

Set the SSID to **NDTEST\_WEP\_AP2**.

Set the Wired equivalent privacy (WEP) status to **enabled**.

Set the WEP key at index 1 to **0123456789**.

Set the active WEP index to **1**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **disabled**.

To configure NDTEST\_WEP\_AP3

Set the SSID to **NDTEST\_WEP\_AP3**.

Set the Wired equivalent privacy (WEP) status to **enabled**.

Set the WEP key at index 1 to **0123456789**.

Set the active WEP index to **1**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **disabled**.

To configure NDTEST\_WPA\_AP1

Set the SSID to **NDTEST\_WPA\_AP1**.

Set the WPA-PSK status to **enabled**.

Set the Pre Shared key to **0123456789**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **disabled**.

To configure NDTEST\_WPA2\_AP1

Set the SSID to **NDTEST\_WPA2\_AP1**.

Set the WPA2-PSK status to **enabled**.

Set the Pre Shared key to **0123456789**.

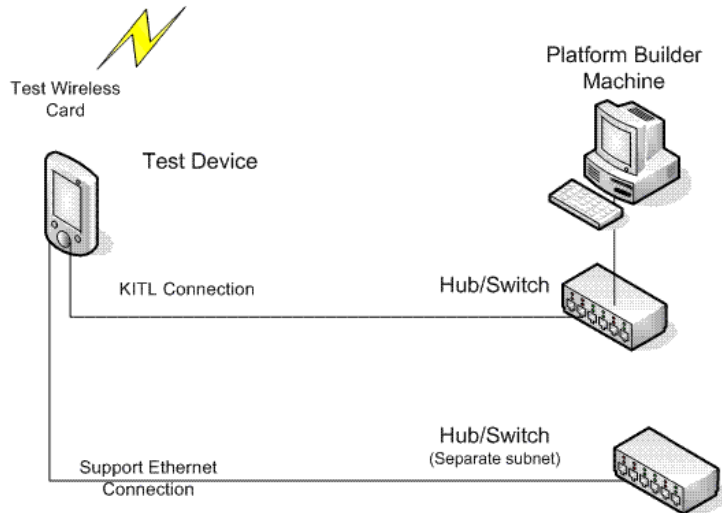
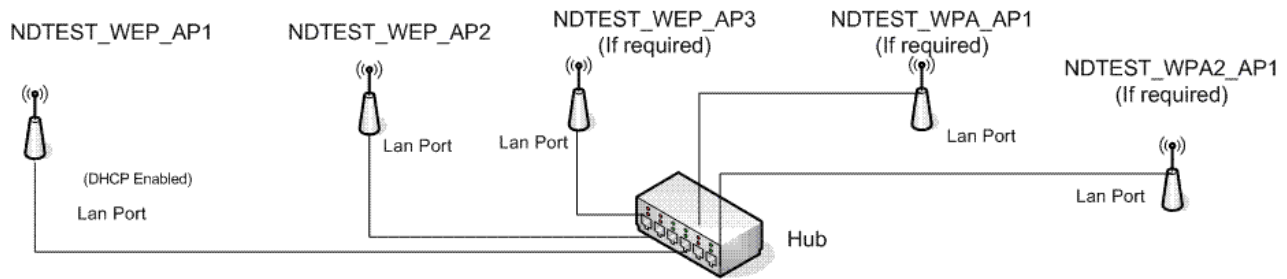
Set authentication mode to **Open only**.

Set WPA Encryption to **TKIP+AES**.

Set the DHCP on the LAN port to **disabled**.

## **CONNECTING THE ACCESS POINTS AND DEVICES**

The access points and test devices should be connected as shown in the following diagram.



Only the LAN ports of the access points are used. The WAN port of the access points is left unconnected. Only the NDTEST\_WEP\_AP1 access point has DHCP enabled.

**SEE ALSO**

**Other Resources**

[One-Card WLAN Card Miniport Driver Test](#)

**Test Cases for the One-Card WLAN Card Miniport Driver Test**

The following table shows the test cases for the One-Card WLAN Card Miniport Driver Test. Each test case has several variations as subtests; if any variation fails, the entire test case is classified as a failure.

Test case	Description
1	WEP: Associatetime Tests the ability to associate with a WEP access point within 2 seconds. The test case expects a Media connect event within that time period.
2	WEP: Bssid The first variation will associate the card and verify the MAC address of the access point is returned. The second variation will verify NDIS_STATUS_ADAPTER_NOT_READY is returned on querying OID_802_11_BSSID while the device is not associated.

3	<p>WEP: Bssidlist</p> <p>The first variation will query OID_802_11_BSSID_LIST with a buffer that is too short and verify one of the correct status codes is returned. The second variation will perform a list scan and verify all the required access points are visible in the list. The third variation will verify that no list items are returned when the radio is off. The fourth variation is skipped currently. The fifth variation will verify that the currently associated SSID is returned in the list. The sixth variation will associate the card, perform numerous list scans and verify that the card does not disconnect while performing the scan. The seventh variation will start a list scan, issue an NdisReset and verify the scan completes successfully. The eighth variation will associate while a scan is in progress and verify the request succeeds. The ninth variation will verify the data fields for each known access point in the environment are set correctly.</p>
4	<p>WEP: Configuration</p> <p>The first variation will check if querying OID_802_11_CONFIGURATION succeeds. The second variation will check if the oid can be successfully set. The third variation will verify that fields returned while querying the oid.</p>
5	<p>WEP: Disassociate</p> <p>The first variation will set OID_802_11_DISASSOCIATE and verify that the device disassociates. The second variation will set the oid when the device is already disassociated and verify that NDIS_STATUS_SUCCESS is returned. The third variation will set the oid while the device is scanning and will verify that the device disconnects.</p>
6	<p>WEP: Mediaevents</p> <p>The first variation will verify a connect event is indicated after association. The second variation will associate with an access point and then associate with the same access point and verify a connect event is indicated. The third variation will associate with an access point and then associate with a different access point and verify that disconnect and connect event are indicated. The fourth variation will verify a disconnect event is indicated after disassociation.</p>
7	<p>WEP: Ndisoids</p> <p>The first variation will query the driver for the supported oid list. The second variation will verify all mandatory and recommended oids are returned in the supported list. For any missing mandatory oid, the test is flagged as an error. For any missing recommended oids, the log will indicate an error message, but the test is classified as passed.</p>
8	<p>WEP: Networktypeinuse</p> <p>The first variation loads and unloads the driver (currently not implemented). The second variation will set OID_802_11_NETWORK_TYPE_IN_USE and verify it succeeds. The third variation will set the oid with an invalid value and verify the request fails. The fourth variation will set the oid with each of the values returned in the supported list. The next variation will associate with each of the access points and verify the appropriate network type is returned. If the access points with appropriate network type is not selected according the table described in this section, then this test will fail.</p>
9	<p>WEP: Powermode</p> <p>The first variation will just associate with the access point. The second variation will verify that querying OID_802_11_POWER_MODE succeeds. The third variation will verify that setting the oid for all valid power modes succeeds. The fourth variation will set power mode to Ndis802_11PowerModeCAM.</p>

10	<p>WEP: Ssid</p> <p>The first variation will query OID_802_11_SSID with an invalid buffer length. The second variation will associate with each known access point in the environment. The third variation will verify that querying the oid returns the correct SSID after association. The fourth variation will verify OID_802_11_SSID query returns 0 for SSIDLength when device is not associated. The fifth variation will verify that the device can associate with any available SSID by specifying a zero length string for the SSID. The sixth variation will set the oid with the SsidLength field set to a value larger than 32 and verify the driver fails the set request.</p>
11	<p>WPA: WPAAssociationinfo</p> <p>The first variation will verify association with a WPA access point. The next variations will query OID_802_11_ASSOCIATION_INFORMATION and verify the data fields.</p>
12	<p>WPA: WPAEncryption</p> <p>The first variation will associate with a WPA access point and verify querying OID_802_11_ENCRYPTION returns TKIP encryption. The second variation will verify setting oid with Ndis802_11Encryption2Enabled to enable TKIP encryption and verify request succeeds.</p>
13	<p>WPA: WPANetworktypeinuse</p> <p>The first variation will verify at least one type is returned on querying OID_802_11_NETWORK_TYPES_SUPPORTED. The next variation will verify the appropriate network type/types is returned depending on the network type of the test wireless card.</p>
21	<p>WPA2: WPA2Authentication</p> <p>The first variation will set OID_802_11_AUTHENTICATION with an invalid value and verify the request fails. The second variation will set the oid with Ndis802_11AuthModeWPA2 with no key set and verify the request succeeds. The third variation will set the oid with Ndis802_11AuthModeWPA2PSK with no key set and verify the request succeeds. The fourth variation will associate the card with a WEP only AP using WPA2-PSK and verify the association fails.</p>
22	<p>WPA2: WPA2Bssidlist</p> <p>The first variation will query OID_802_11_BSSID_LIST and verify RSN IE is returned.</p>
23	<p>WPA2: WPA2Capability</p> <p>The first variation will query OID_802_11_CAPABILITY to verify support. The second variation will verify device supports Open authentication with no encryption. The third variation will verify device supports Open authentication with WEP encryption. The fourth variation will verify the driver supports WPA2 authentication with AES encryption. The fifth variation will verify the capability fields returned on querying the oid. The sixth variation will verify each of the authentication encryption values returned are valid enumerations.</p>
24	<p>WPA2: WPA2Encryption</p> <p>The first variation will verify Ndis802_11Encryption3Enabled is return when associated with a WPA2-PSK AP. The second variation will verify Ndis802_11Encryption3KeyAbsent is returned when all keys are removed.</p>

25	<p>WPA2: WPA2Pmkid</p> <p>The first variation will set OID_802_11_PMKID with an invalid buffer length and verify NDIS_STATUS_INVALID_LENGTH is returned. The second variation is skipped currently. The third variation will set the oid with valid PMKID data, then query it and verify the data returned matches the data previously set. The fourth variation will verify PMKID list is cleared when BSSIDInfoCount is set to zero. The fifth variation will set the oid with maximum number of supported PMKIDs and verify the request succeeds. The sixth variation will set the oid with more PMKIDs than what the driver supports and verify the driver fails the request. The seventh variation will set the oid with more than 16 PMKIDs and verify the driver fails the request.</p>
----	--

## SEE ALSO

### Other Resources

[One-Card WLAN Card Miniport Driver Test](#)

### Command Line Parameters for the One-Card WLAN Card Miniport Driver Test

The One-Card Wireless Network Card Miniport Driver Test executes the tux `-o -d ndt_1c_wlan -c"-t DEVICE_NAME -nounbind"` command line, where **DEVICE\_NAME** is the name of the wireless network card being tested; for example, CISCO1. You can modify the test by further editing the command line. For information about how to edit the command line for a test, see *Editing the Command Line for a Test*. The following table shows the command line parameters for the *One-Card Wireless Network Card Miniport Driver Test*.

Command line parameter	Description
-packets	(Optional) Logs information when a test confirms that a packet has been sent or received.
-nounbind	(Required for WPA and WPA2 testing) Disables unbinding of other protocol drivers from the test adapter before the test is run.
-displayssidlist	(Optional) Displays the list of access points as seen by the test network card before the start of the test. This is useful to confirm whether the Access points required by the test are visible.
-strictness level	(Optional) Controls the strictness of error reporting of the test by the level number, an integer between 1 and 5. Default is 5 (highest strictness), which means all behavior that deviates from the ndis miniport guide is flagged. By passing value 1 (lowest strictness), the test will allow some failures which do not prevent the card from working with the existing operating system. Other values are currently unused.

## SEE ALSO

### Other Resources

[One-Card WLAN Card Miniport Driver Test](#)

### Two-Card WLAN Card Miniport Driver Test

The Two-Card Wireless Network Card Miniport Driver Test assesses the functionality of a Wireless miniport driver on a target device with respect to interaction with another wireless card. These

tests require either a second wireless card on the device or another device with a second wireless card called the support wireless card. The operation of this test is very similar to the *Two-Card Network Card Miniport Driver Test*, which tests functionality common to all network cards. This test has additional wireless-specific test setup requirements.

## IN THIS SECTION

### [Prerequisites for the Two-Card WLAN Card Miniport Driver Test](#)

Specifies the hardware and software requirements for this test

### [One-Card WLAN Card Miniport Driver Test](#)

### [Two-Card WLAN Card Miniport Driver Test](#)

### Setting Up and Configuring the Two-Card WLAN Card Miniport Driver Test

Describes the setup and configuration of this test

### [One-Card WLAN Card Miniport Driver Test](#)

### [Two-Card WLAN Card Miniport Driver Test](#)

### Test Cases for the Two-Card WLAN Card Miniport Driver Test

Describes the test cases for this test

### [Command Line Parameters for the One-Card WLAN Card Miniport Driver Test](#)

Describes the command line parameters for this test

## SEE ALSO

### Other Resources

### [Prerequisites for the Two-Card WLAN Card Miniport Driver Test](#)

The following table shows the hardware requirements for the Two-Card WLAN Card Miniport Driver Test.

Requirement	Description
Device with two wireless network cards and a third network card -OR- Two devices, each with a wireless network card and a second network card	The target device must have a wireless (802.11) network card for which the miniport driver is to be tested, and another network card to connect to the development workstation.  The target device can also have a second wireless network card. Alternatively, a second device with a wireless network card and a second network card can be used.
Access points	Used for the test; must be configured based on the network types (802.11 a, b, g, a/b, or a/g) and on whether the card supports WPA and/or WPA2. Setup and configuration for these is found in <a href="#">One-Card WLAN Card Miniport Driver Test</a> <a href="#">Two-Card WLAN Card Miniport Driver Test</a> Setting Up and Configuring the Two-Card WLAN Card Miniport Driver Test.

This test is comprised of two binaries, Ndt.dll and Ndt\_2c\_wlan.dll. The Ndt.dll binary file is a protocol driver that binds to the test and support cards. The protocol driver communicates with the underlying miniport drivers through an NDIS wrapper and registers as a stream driver. The

Ndt\_2c\_wlan.dll binary file controls the test itself. The following table shows the software requirements for the Two-Card WLAN Card Miniport Driver Test.

Requirement	Description
Tux.exe	Test harness, required for test execution
Kato.dll	Logging engine, required for test execution
Ndt.dll	Protocol driver for the test
Ndt_2c_wlan.dll	Test library
DummyWzcsvc.cab	Must be installed on all retail devices to replace the existing wzcsvc with a dummy. This enables the test to take complete control of the wireless cards without interference from wzcsvc.

The network cards that you use must be recognized by the Network Driver Interface Specification (NDIS) architecture. You must assign a name to each network card prior to running the test. The run-time image that you download to the target device must use shared miniport drivers on control cards. You must bind the TCP/IP protocol to one of the miniport drivers.

## SEE ALSO

### Other Resources

[One-Card WLAN Card Miniport Driver Test](#)

Two-Card WLAN Card Miniport Driver Test

## Setting Up and Configuring the Two-Card WLAN Card Miniport Driver Test

The following table shows which access point types are required for each network type.

Test card network type	NDTEST_WEP_AP1	NDTEST_WEP_AP2	NDTEST_WEP_AP3	NDTEST_WPA_AP1	NDTEST_WPA2_AP1
802.11a	802.11a	802.11a	-	802.11a	802.11a
802.11b	802.11b	802.11b	-	802.11b	802.11b
802.11g	802.11b	802.11g	-	802.11g	802.11g
802.11a/b	802.11b	802.11a	-	802.11a	802.11a
802.11a/g	802.11b	802.11a	802.11g	802.11g	802.11g

The access points for the Two-Card WLAN Card Miniport Driver Test are selected based on the driver needs. Wired equivalent privacy (WEP)-only access points must be selected as a minimum; cards that support WPA or WPA2 must select additional access points.

## CONFIGURING THE ACCESS POINTS

Configure each access point selected from the table above according to the following information.

To configure NDTEST\_WEP\_AP1

Set the SSID to **NDTEST\_WEP\_AP1**.

Set the Wired equivalent privacy (WEP) status to **enabled**.

Set the WEP key at index 1 to **0123456789**.

Set the WEP key at index 2 to **9876543210**.

Set the WEP key at index 3 to **1020304050**.

Set the WEP key at index 4 to **5040302010**.

Set the active WEP index to **1**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **enabled**. This is the only access point which will have DHCP enabled.

To configure NDTEST\_WEP\_AP2

Set the SSID to **NDTEST\_WEP\_AP2**.

Set the Wired equivalent privacy (WEP) status to **enabled**.

Set the WEP key at index 1 to **0123456789**.

Set the active WEP index to **1**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **disabled**.

To configure NDTEST\_WEP\_AP3

Set the SSID to **NDTEST\_WEP\_AP3**.

Set the Wired equivalent privacy (WEP) status to **enabled**.

Set the WEP key at index 1 to **0123456789**.

Set the active WEP index to **1**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **disabled**.

To configure NDTEST\_WPA\_AP1

Set the SSID to **NDTEST\_WPA\_AP1**.

Set the WPA-PSK status to **enabled**.

Set the Pre Shared key to **0123456789**.

Set authentication mode to **Open only**.

Set the DHCP on the LAN port to **disabled**.

To configure NDTEST\_WPA2\_AP1

Set the SSID to **NDTEST\_WPA2\_AP1**.

Set the WPA2-PSK status to **enabled**.

Set the Pre Shared key to **0123456789**.

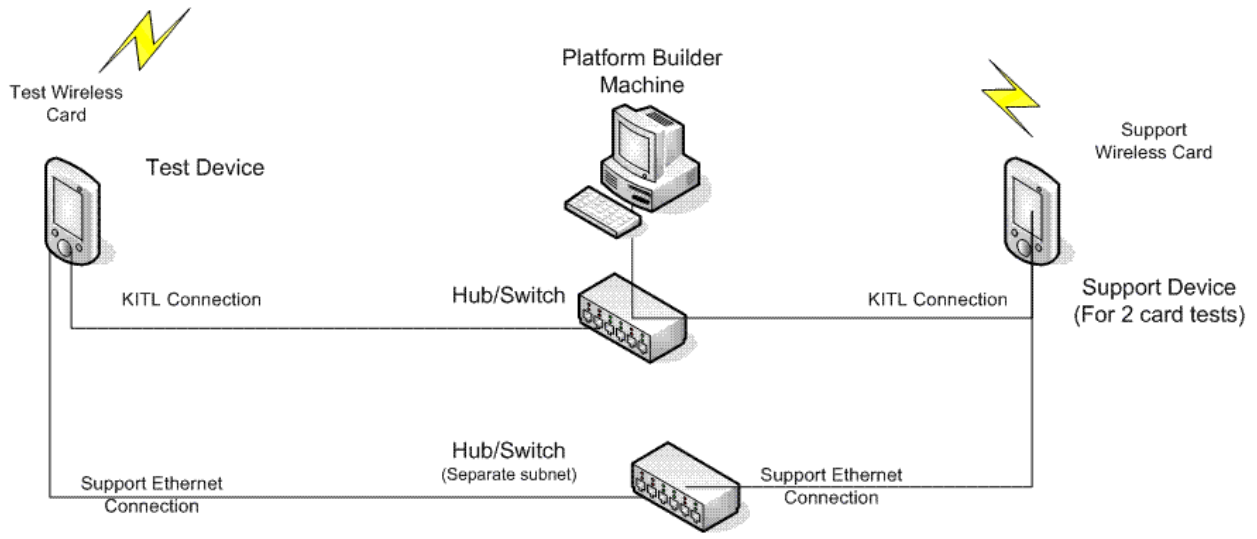
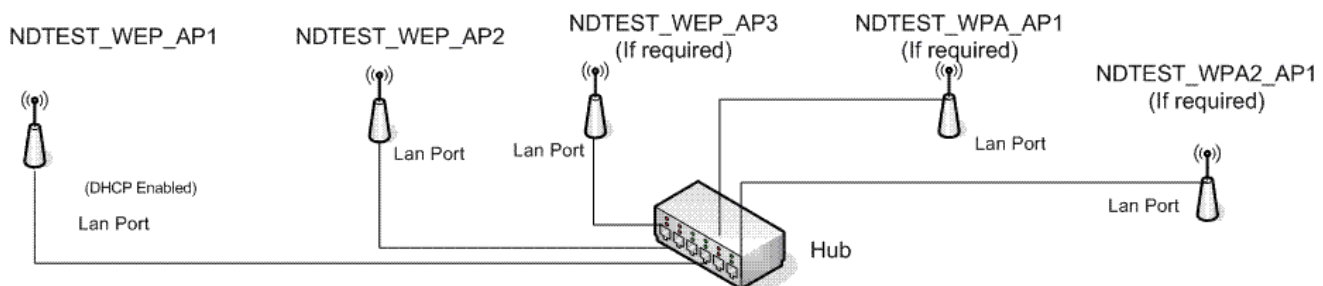
Set authentication mode to **Open only**.

Set WPA Encryption to **TKIP+AES**.

Set the DHCP on the LAN port to **disabled**

### **Connecting the Access Points and Devices**

The access points and test devices should be connected as shown in the following diagram.



Only the LAN ports of the access points are used. The WAN port of the access points is left unconnected. Only the NDTEST\_WEP\_AP1 access point has DHCP enabled. Make sure that the IP address ranges for the access points and the KITL connection subnets do not overlap or the two-card test might fail to cleanup NDT from the support wireless card.

**SEE ALSO**

**Other Resources**

[One-Card WLAN Card Miniport Driver Test](#)

[Two-Card WLAN Card Miniport Driver Test](#)

**Test Cases for the Two-Card WLAN Card Miniport Driver Test**

The following table shows the test cases for the Two-Card WLAN Card Miniport Driver Test.

Test case	Description
1	<p>WEP: Adhoc</p> <p>The first variation will verify sending and receiving directed packets in IBSS mode. The second variation will verify sending and receiving broadcast packets in IBSS mode. The third variation will verify no media connect event is indicated when creating the first cell. The fourth variation will verify a media connect event is indicated by both IBSS nodes after joining each other. The fifth variation will verify the remaining cell indicates a disconnect event after all nodes leave. The sixth variation will verify association with IBSS in 2 seconds or less.</p>

2	<p>WEP: Reloaddefaults</p> <p>The first variation will verify setting <code>OID_802_11_RELOAD_DEFAULTS</code> does not cause disconnect. The second variation will verify setting the oid does not change the authentication mode. The third variation will verify setting <code>OID_802_11_RELOAD_DEFAULTS</code> changes encryption status. The fourth variation will verify setting the oid removes all default WEP keys.</p>
3	<p>WEP: Statistics</p> <p>The first variation will verify <code>OID_802_11_STATISTICS</code> is supported. The second variation will verify creation of adhoc cell on the second device and join the test device. The third variation will verify <code>TransmittedFragmentCount</code> and <code>ReceivedFragmentCount</code> counters are updated. The fourth variation will verify <code>MulticastTransmittedFrameCount</code> &amp; <code>MulticastReceivedFrameCount</code> counters are updated. The fifth variation will verify <code>FailedCount</code> is updated. The sixth variation will verify <code>RetryCount</code> and <code>MultipleRetryCount</code> are updated.</p>
4	<p>WEP: Wep</p> <p>The first variation will verify <code>OID_802_11_ENCRYPTION_STATUS</code> can be set with no key present. The second variation will verify directed packets can be sent and received using WEP. The third variation will verify broadcast packets can be sent and received using WEP. The fourth variation will verify broadcast and directed packets can be sent using each of the 4 default WEP keys. The fifth variation will verify packets are not received after WEP key is removed. The sixth variation will verify device does not disassociate after the WEP key is removed. The seventh variation will run stress while adding/removing keys. The eighth variation will verify <code>OID_802_11_ADD_WEP</code> fails when setting an invalid key index</p>
11	<p>WPA: WPAAddKey</p> <p>The first variation will associate the support wireless card with the WEP AP. The second variation will set <code>OID_802_11_ADD_KEY</code> with a pairwise key that has bits other than 31 and 30 set and verify the request fails. The third variation will set the oid with a pairwise key that does not have the transmit bit set and verify the request fails. The fourth variation will set <code>OID_802_11_ADD_KEY</code> with a pairwise key that has the BSSID set to <code>FF:FF:FF:FF:FF:FF</code> and verify the request fails. The fifth variation will set the oid with a TKIP pairwise key that is less than the required length and verify the request fails. The sixth variation will set the oid with a TKIP pairwise key that is larger than the required length and verify the request fails. The seventh variation will verify last pairwise key added overwrites previous pairwise key. The eighth variation will verify last group key added overwrites previous group key. The ninth variation will verify broadcast packets are not received with pairwise keys.</p>
12	<p>WPA: WPAAdhoc</p> <p>The first variation will set <code>OID_802_11_AUTHENTICATION_MODE</code> with invalid value and verify the request fails. The second variation will verify sending and receiving directed packets in IBSS mode. The third variation will verify sending and receiving broadcast packets in IBSS mode</p>
13	<p>WPA: WPAAuthentication</p> <p>The first variation will set <code>OID_802_11_AUTHENTICATION_MODE</code> with invalid value and verify the request fails. The second variation will verify all valid authentication modes are supported. The third variation will associate with an open WEP only AP using WPAPSK and verify the association fails</p>

14	WPA: WPABssidlist The first variation will verify IBSS node is visible in the list. The second variation will verify BSSID information elements. The third variation will verify packets sent while a scan is in progress are received.
15	WPA: WPAInfrastructure The first variation will associate support wireless card with the WEP AP. The second variation will verify all keys are discarded when setting <code>OID_802_11_INFRASTRUCTURE_MODE</code> .
16	WPA: WPARemoveKey The first variation will associate support device with WEP AP. The second variation will remove a group key that has the transmit bit set and verify the request fails. The third variation will remove a pairwise key that has the transmit bit set and verify the request fails. The fourth variation will remove a key with an invalid key index and verify the request fails. The fifth variation will verify group keys can be removed. The sixth variation will verify pairwise keys can be removed.
17	WPA: WPASendCheck The first variation will associate support device with WEP AP. The second variation will verify directed packets can be sent and received using TKIP. The will verify broadcast packets can be sent using TKIP.
18	WPA: WPASendRecvAes The first variation will associate support device with WEP AP. The second variation will verify directed packets can be sent and received using AES. The will verify broadcast packets can be sent using AES.
21	WPA2: WPA2Misc The first variation will associate with a WEP AP using WPA2-PSK and verify the association fails. The second variation will associate with a WPA-PSK AP using WPA2-PSK and verify the association fails.
22	WPA2: WPA2SendCheck The first variation will associate support device with WEP AP. The second variation will verify directed packets can be sent and received using AES. The third variation will verify broadcast packets can be sent and received using AES.
23	WPA2: WPA2SendRecv The first variation will associate support device with WEP AP. If the card supports TKIP, the second variation will verify sending and receiving directed and broadcast packets can be sent and received using TKIP. If the card supports AES, the third variation will verify sending and receiving directed and broadcast packets can be sent and received using AES.

## SEE ALSO

### Other Resources

[One-Card WLAN Card Miniport Driver Test](#)

Two-Card WLAN Card Miniport Driver Test

### Command Line Parameters for the Two-Card WLAN Card Miniport Driver Test

The Two-Card Wireless Network Card Miniport Driver Test executes the `tux -o -d ndt_2c_wlan -c"-t DEVICE_NAME -s SUPPORT_DEVICE_NAME@IP_ADDRESS -nounbind"` command line, where **DEVICE\_NAME** is the name of the wireless network card being tested (for example, `-t CISCO1`) and

Writing Sample--Windows Mobile Wireless Tests—"Before" Example  
Katy Koenen

**SUPPORT\_DEVICE\_NAME@IP\_ADDRESS** is the device name of the supported wireless network card; **IP\_ADDRESS** is replaced by the IP address of the support Ethernet connection card on the support device (for example, `-s ISLP21@192.168.0.1`). You can modify the test by further editing the command line. For information about how to edit the command line for a test, see *Editing the Command Line for a Test*. The following table shows the command line parameters for the *One-Card Wireless Network Card Miniport Driver Test*.

Command line parameter	Description
<code>-packets</code>	(Optional) Logs information when a test confirms that a packet has been sent or received.
<code>-nounbind</code>	(Required for WPA and WPA2 testing) Disables unbinding of other protocol drivers from the test adapter before the test is run.
<code>-displayssidlist</code>	(Optional) Displays the list of access points as seen by the test network card before the start of the test. This is useful to confirm whether the Access points required by the test are visible.
<code>-strictness level</code>	(Optional) Controls the strictness of error reporting of the test by the level number, an integer between 1 and 5. Default is 5 (highest strictness), which means all behavior that deviates from the ndis miniport guide is flagged. By passing value 1 (lowest strictness), the test will allow some failures which do not prevent the card from working with the existing operating system. Other values are currently unused.
<code>-directedpassrate rate</code>	(Optional) Sets the expected percentage of packets to be received by the other adapter for the broadcast send tests to be declared as a pass. Default value is 85.
<code>-broadcastpassrate rate</code>	(Optional) Sets the expected percentage of packets to be received by the other adapter for the directed send tests to be declared as a pass. Default value is 60.

## SEE ALSO

### Other Resources

[One-Card WLAN Card Miniport Driver Test](#)

Two-Card WLAN Card Miniport Driver Test