

Configuration Guide

How to Configure a BYOD Environment with the Unified AP in Standalone Mode



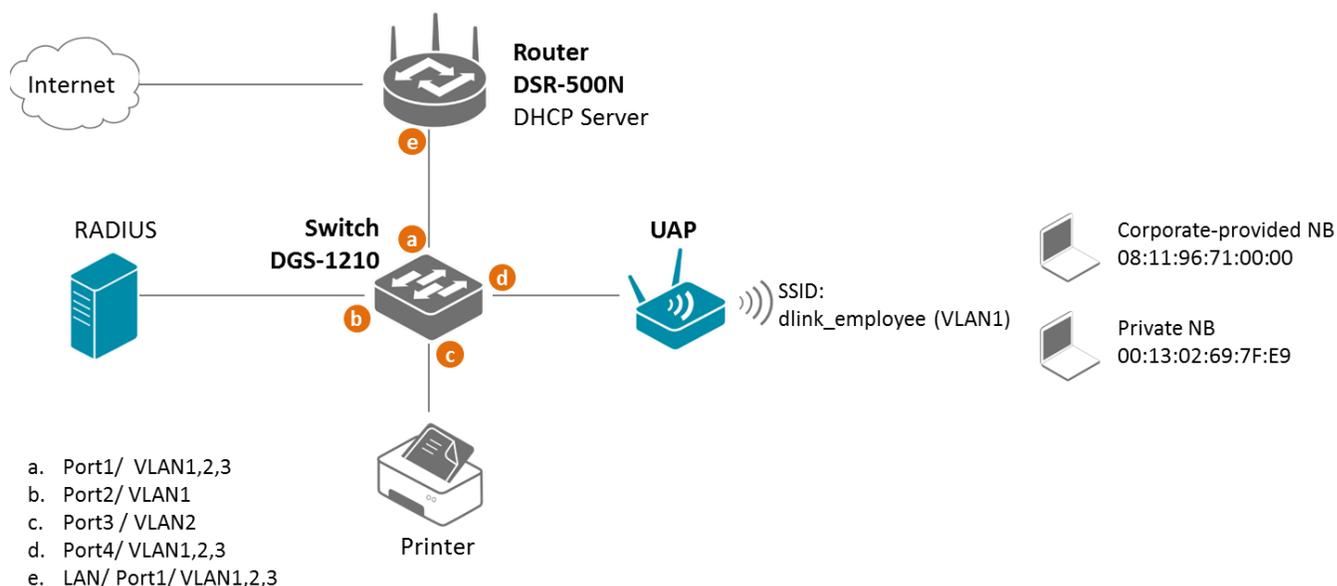
Overview

This guide describes how to configure and implement BYOD environment with the D-Link Unified Access Point for user and device authentication.

Situation Note

The trend of Bring Your Own Device (BYOD) in working place is a new challenge on network security and management. Many corporations that allow employees to use their own device at work expecting have better performance and productivity; however, on the downside, corporations also concern the network security and information leakage by using private device. How to distinguish corporate-provided device and private device (BYOD device), and give different authorities is the major task for IT teams.

The scenario in this guide shows you how to implement a BYOD environment with Unified Access Point in standalone mode and external RADIUS (FreeRADIUS) server. Use username, password, and device MAC info to assign particular VLAN. All connection from the SSID required performing authentication before granted authority.



The Unified AP is in the standalone mode. The security protocol on SSID dlink_employee is WPA2 Enterprise. The authentication database is external RADIUS server. In the RADIUS database, one user account includes username, password, and device MAC address which is the corporate-provided. The authorized network is assigned based on authentication information:

- If authentication info matches username, password, and device MAC address of the user account, the user is authorized in VLAN2 network.
- If authentication info matches username and password, but it doesn't match the device MAC address (for example, use the Private NB to log on), the user is authorized in VLAN3 network.
- If authentication info doesn't match either username or password, the user doesn't get any access.

NOTE: The screenshots in this guide are from the DWL-6600AP's firmware version 4.2.0.9_B001. If you are using an earlier version of the firmware, the screenshots may not be identical to what you see on your browser.

Configuration Steps (FreeRADIUS)

1. Basic Requirement

In order to setup the RADIUS server, the following is the minimum requirement.

- A standard x86/x86-64 PC
- Installed Fedora Linux distribution (Fedora 18+ is preferred)
- 10GB HDD storage at least
- 1GB ram at least
- Internet connection

2. Recommend Software Package list

All configuration steps are verification base on software version below:

Software Type	Software Name	Version
Operation System	Fedora	3.9.5-301.fc19.x86_64
FreeRadius	freeradius	2.2.0-6.fc19.x86_64
FreeRadius	freeradius-utils	2.2.0-6.fc19.x86_64
FreeRadius	freeradius-postgresql	2.2.0-6.fc19.x86_64
Postgresql	postgresql-server	9.2.6-1.fc19.x86_64
Postgresql	postgresql-libs	9.2.6-1.fc19.x86_64

3. Configure IP address on Fedora via GUI.

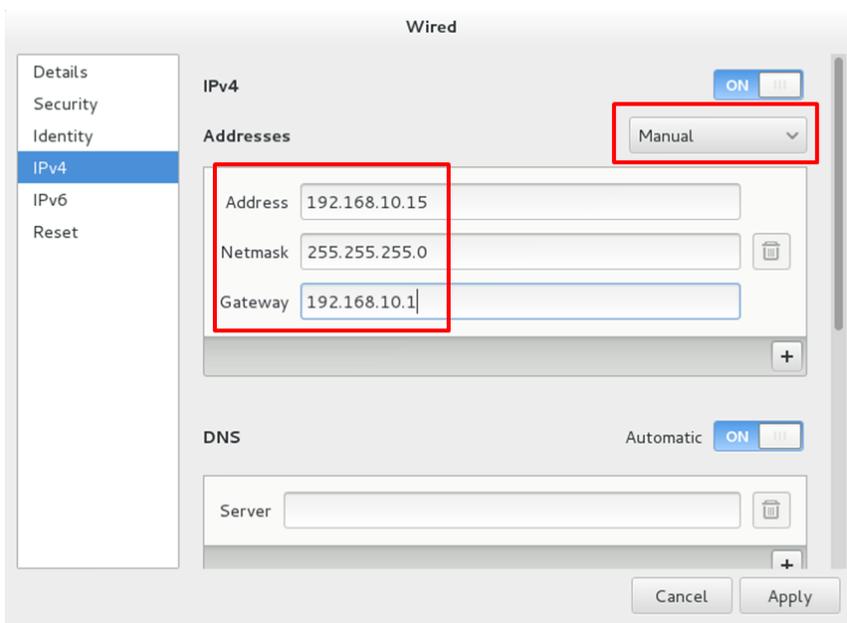
3-1. Log in as root in GUI.

3-2. Select Network Settings.



3-3. Click the gear. Manually set the IP address, Netmask and Gateway. In this case, set the FreeRADIUS IP address as 192.168.10.15. The Netmask is 255.255.255.0. The Gateway IP address is 192.168.10.1.

Note: Make sure the RADIUS server connect to internet before process following procedures.



4. Manual-Installation Procedure

Install FreeRADIUS steps-by-steps through the following description.

4-1. Open a terminal console and switch to root account

Use the `su` command and enter root's password to get the root privilege as the following steps are all needed root privilege.

```
[scottie@localhost ~]$ su
Password:
[root@localhost scottie]# _
```

4-2. Install the required package (the table listed in above)

Use the following command to install freeradius, postgresql, and the libraries. In default, the installation path for FreeRADIUS is `/etc/raddb`.

```
-----
yum install postgresql-server postgresql-libs freeradius freeradius-postgresql freeradius-utils
-----
```

```
[root@localhost scottie]# yum install postgresql-server postgresql-libs freeradius freeradius-postgresql freeradius-utils
Loaded plugins: langpacks, presto, refresh-packagekit
fedora/19/i386/metalink | 8.0 kB 00:00:00
updates/19/i386/metalink | 5.5 kB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package freeradius.i686 0:2.2.0-5.fc18 will be installed
--> Package freeradius-postgresql.i686 0:2.2.0-5.fc18 will be installed
--> Package freeradius-utils.i686 0:2.2.0-5.fc18 will be installed
--> Package postgresql-libs.i686 0:9.2.4-1.fc18 will be installed
--> Package postgresql-server.i686 0:9.2.4-1.fc18 will be installed
--> Processing Dependency: postgresql(x86_32) = 9.2.4-1.fc18 for package: postgresql-server-9.2.4-1.fc18.i686
--> Running transaction check
--> Package postgresql.i686 0:9.2.4-1.fc18 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
freeradius i686 2.2.0-5.fc18 updates 1.4 M
freeradius-postgresql i686 2.2.0-5.fc18 updates 79 k
freeradius-utils i686 2.2.0-5.fc18 updates 148 k
postgresql-libs i686 9.2.4-1.fc18 updates 226 k
postgresql-server i686 9.2.4-1.fc18 updates 3.6 M
Installing for dependencies:
postgresql i686 9.2.4-1.fc18 updates 3.2 M
=====
Transaction Summary
-----
Install 5 Packages (+1 Dependent package)
Total download size: 8.7 M
Installed size: 39 M
Is this ok [y/N]: _
```

4-3. Configure FreeRADIUS. All configuration files for FreeRADIUS will be stored under `/etc/raddb`.

Add the management VLAN in the FreeRADIUS. Edit `/etc/raddb/client.conf`. Add shared secret for each client or each subnet. And save.

The fill in information is as below:

- `short_name` : the name of this entry
- `secret` : the secret for to this entry
- `ipaddr` and `netmask` : the ip address for this entry, you can specify an address or a subnet

The red items are the options you can edit

```
-----
client short_name{
    secret = shared_secret
    ipaddr = 192.168.0.0
    netmask = 24
}
```

```
}
-----
```

In this case, add VLAN1 IP subnet. For example, add a new entry named **test1**, secret is **testing123**, and the subnet is **192.168.10.0/24**

```
-----
client test1{
    secret=testing123
    ipaddr=192.168.10.0
    netmask=24
}
-----
```

```
root@localhost:~
#####
#
# Per-socket client lists. The configuration entries are exactly
# the same as above, but they are nested inside of a section.
#
# You can have as many per-socket client lists as you have "listen"
# sections, or you can re-use a list among multiple "listen" sections.
#
# Un-comment this section, and edit a "listen" section to add:
# "clients = per_socket_clients". That IP address/port combination
# will then accept ONLY the clients listed in this section.
#
#
#clients per_socket_clients {
#   client 192.168.3.4 {
#       secret = testing123
#   }
#}
client test1{
    secret=testing123
    ipaddr=192.168.10.0
    netmask=24
}
-- INSERT --
```

4-4. Setup SQL server is as source database. Uncomment sql.conf in `/etc/raddb/radiusd.conf`.

Remove “#” in the beginning of “`$INCLUDE sql.conf`” to enable SQL as the data source of FreeRADIUS. And save.

```
#
# $INCLUDE sql.conf
#
```

4-5. Setup database type, host name and server username/ password.

Edit below info under `/etc/raddb/sql.conf`. And save.

4-5-1. Set "database" = "postgresql"

4-5-2. Set "server" = the database server ip. Leave it as "localhost" if you don't have separate database.

4-5-3. Change "password" as desired. Suggest keep it as "radpass"

```
#
database = "postgresql"

#
# Which FreeRADIUS driver to use.
#
driver = "rlm_sql_${database}"

# Connection info:
server = "localhost"
#port = 3306
login = "radius"
password = "radpass"
```

4-6. Edit log in format.

Edit below info under `/etc/raddb/sql/postgresql/dialup.conf`. And save.

4-6-1. Remove "#" in the beginning of "sql_user_name = "%{Stripped-User-Name}:-%{User-Name}:-none}" "

4-6-2. Add "#" in the beginning of "sql_user_name = "%{User-Name}" "

```
sql_user_name = "%{Stripped-User-Name}:-%{User-Name}:-none}"
# sql_user_name = "%{User-Name}"
```

4-7. Enable Authorize and Accounting function on the SQL.

Edit below info under `/etc/raddb/sites-enabled/default`. And save.

4-7-1. Remove "#" in the beginning of "sql" in the sections of "authorize", "accounting"

```
accounting {
# sql
#
```

```
authorize {
# sql
#
```

4-7-2. Please insert text below to the `/etc/raddb/sites-enabled/default` after line 511 and save change.

```

-----
if ( "%{request:Calling-Station-Id}" != "" && "%{request:Calling-Station-Id}" == "%{sql: SELECT
callingstationid FROM radmacvlan WHERE username='%{User-Name}' and
callingstationid=upper('%{request:Calling-Station-Id}')}" ) {
    update reply {
        Tunnel-Private-Group-ID := "%{sql: SELECT tunnelprivategroupid FROM
radmacvlan WHERE username='%{User-Name}' and callingstationid=upper('%{request:Calling-
Station-Id}')}"
        Tunnel-Type := "%{sql: select value from radgroupreply right outer join
radusergroup on radgroupreply.groupname=radusergroup.groupname where
radusergroup.username='%{User-Name}' and radgroupreply.attribute='Tunnel-Type' }"
        Tunnel-Medium-Type := "%{sql: select value from radgroupreply right outer join
radusergroup on radgroupreply.groupname=radusergroup.groupname where
radusergroup.username='%{User-Name}' and radgroupreply.attribute='Tunnel-Medium-
Type' }"
    }
}
else {
    update reply {
        Tunnel-Private-Group-Id := "%{sql: select value from radgroupreply right outer join
radusergroup on radgroupreply.groupname=radusergroup.groupname where
radusergroup.username='%{User-Name}' and radgroupreply.attribute='Tunnel-Private-Group-
Id' }"
        Tunnel-Type := "%{sql: select value from radgroupreply right outer join
radusergroup on radgroupreply.groupname=radusergroup.groupname where
radusergroup.username='%{User-Name}' and radgroupreply.attribute='Tunnel-Type' }"
        Tunnel-Medium-Type := "%{sql: select value from radgroupreply right outer join
radusergroup on radgroupreply.groupname=radusergroup.groupname where
radusergroup.username='%{User-Name}' and radgroupreply.attribute='Tunnel-Medium-
Type' }"
    }
}
}
-----

```

3-8. Edit `/etc/raddb/sites-enabled/inner-tunnel`

Remove “#” in the beginning of “sql” in the sections of “authorize”

```
# See "Authorization Queries" in sql.conf
sql
```

5. Setup PostgreSQL server

5-1. Start Postgresql service

Execute the following commands to init and start postgresql. And save.

```
-----
service postgresql initdb
service postgresql enable
service postgresql start
-----
```

```
root@localhost scottie]# service postgresql initdb
hint: the preferred way to do this is now "postgresql-setup initdb"
initializing database ... OK
root@localhost scottie]# service postgresql enable
redirecting to /bin/systemctl enable postgresql.service
ln -s '/usr/lib/systemd/system/postgresql.service' '/etc/systemd/system/multi-user.target.wants/postgresql.service'
root@localhost scottie]# service postgresql start
redirecting to /bin/systemctl start postgresql.service
root@localhost scottie]# _
```

5-2. Create a database user for FreeRADIUS.

5-2-1. Create a database user for FreeRADIUS. Please note that the username and password must be matched with username/password which set in `/etc/raddb/sql.conf`. In the settings of previous steps, the username/ password are radius/ radpass.

```
-----
sudo -u postgres createuser radius --no-superuser --no-createdb --no-createrole -P
-----
```

```
root@localhost /]# sudo -u postgres createuser radius --no-superuser --no-createdb --no-createrole -P
enter password for new role:
enter it again:
root@localhost /]# _
```

5-2-2. Create a database for FreeRadius

Create a database for FreeRADIUS. The owner of this database should be the one we defined in `/etc/raddb/sql.conf`.

```
-----
sudo -u postgres createdb radius --owner=radius
-----
```

5-2-2-1. Modify PostgreSQL listen address

Set IP address that PostgreSQL are listened on. Edit `/var/lib/pgsql/data/postgresql.conf`. Remove “#” in the beginning `listen_addresses`. And save.

```
#listen_addresses = 'localhost'      # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)
```

5-2-2-2. Edit `/var/lib/pgsql/data/pg_hba.conf`.

Remove “#” in the beginning of “local all all peer”.

Add two pieces info in the next line.

```
-----
local  all  all  md5
host   all  all  0.0.0.0/0  md5
-----
```

```
root@localhost:~
# use "pg_ctl reload" to do that.

# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records.  In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# TYPE  DATABASE  USER  ADDRESS  METHOD

# "local" is for Unix domain socket connections only
#local  all      all    peer
local  all      all    md5
host   all      all    0.0.0.0/0  md5
# IPv4 local connections:
host   all      all    127.0.0.1/32  ident
# IPv6 local connections:
host   all      all    ::1/128  ident
# Allow replication connections from localhost, by a user with the
-- INSERT --
```

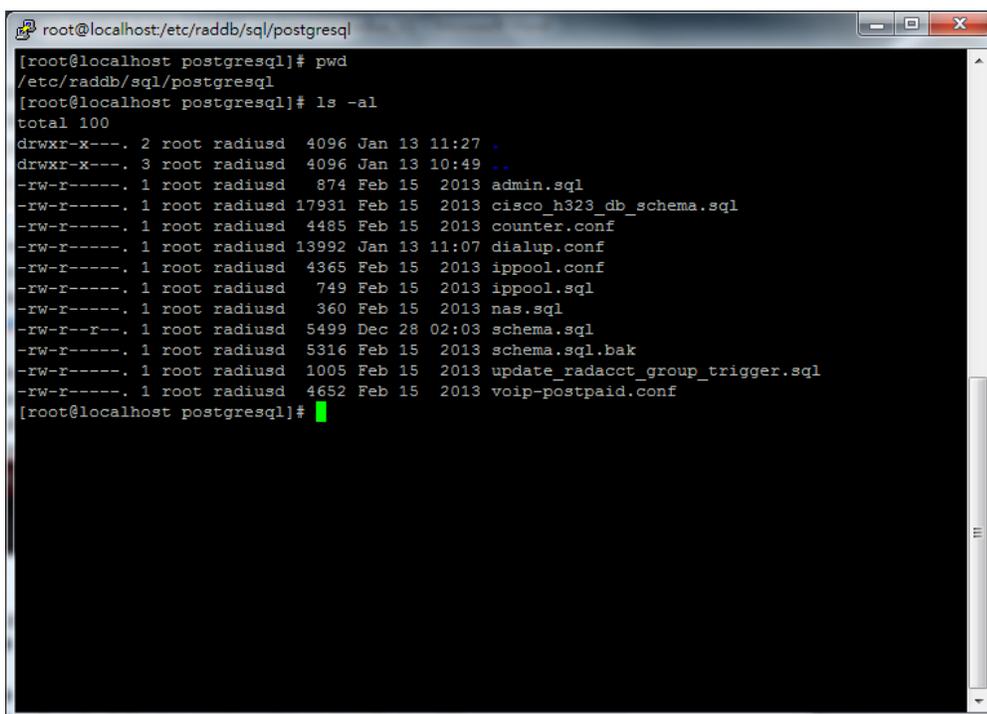
5-2-2-3. Run the following command to re-start PostgreSQL

```
-----  
service postgresql restart  
-----
```

5-2-3. Import FreeRADIUS schemas.

Create a default group and insert a test user into the database. Please copy the schema.sql file which provide by D-Link to replace the existence one under `/etc/raddb/sql/postgresql/`.

```
-----  
cd /etc/raddb/sql/postgresql/  
chown root:radius schema.sql  
-----
```



```
root@localhost:/etc/raddb/sql/postgresql  
[root@localhost postgresql]# pwd  
/etc/raddb/sql/postgresql  
[root@localhost postgresql]# ls -al  
total 100  
drwxr-x---. 2 root radiusd 4096 Jan 13 11:27 .  
drwxr-x---. 3 root radiusd 4096 Jan 13 10:49 ..  
-rw-r-----. 1 root radiusd 874 Feb 15 2013 admin.sql  
-rw-r-----. 1 root radiusd 17931 Feb 15 2013 cisco_h323_db_schema.sql  
-rw-r-----. 1 root radiusd 4485 Feb 15 2013 counter.conf  
-rw-r-----. 1 root radiusd 13992 Jan 13 11:07 dialup.conf  
-rw-r-----. 1 root radiusd 4365 Feb 15 2013 ippool.conf  
-rw-r-----. 1 root radiusd 749 Feb 15 2013 ippool.sql  
-rw-r-----. 1 root radiusd 360 Feb 15 2013 nas.sql  
-rw-r--r--. 1 root radiusd 5499 Dec 28 02:03 schema.sql  
-rw-r-----. 1 root radiusd 5316 Feb 15 2013 schema.sql.bak  
-rw-r-----. 1 root radiusd 1005 Feb 15 2013 update_radacct_group_trigger.sql  
-rw-r-----. 1 root radiusd 4652 Feb 15 2013 voip-postpaid.conf  
[root@localhost postgresql]#
```

Use the command below to create the table schema for database.

```
-----  
sudo cat /etc/raddb/sql/postgresql/schema.sql | psql -U radius radius  
-----
```

5-2-4. Set the default attribute to the default group.

Please use commands below to add the 3 default attributes to default group.

The values need to change:

- `groupname`: Define by user. We can only define one default vlan in the demo scenario.
- `default_vlan_id`: Define by user. We can only define one default vlan in the demo scenario.

```
-----  
echo "insert into radgroupreply (groupname,attribute,op,value) values('groupname','Tunnel-  
Private-Group-Id',':','=','default_vlan_id');" | psql -U radius radius  
-----
```

```
-----  
echo "insert into radgroupreply (groupname,attribute,op,value) values('groupname',' Tunnel-  
Type',':','=','13');" | psql -U radius radius  
-----
```

```
-----  
echo "insert into radgroupreply (groupname,attribute,op,value) values('groupname',' Tunnel-  
Medium-Type',':','=','6');" | psql -U radius radius  
-----
```

In this case, set the default VLAN as VLAN₃. While the authentication information matches username/ password but doesn't match MAC address, the RADIUS accepts the authentication but assign attribute default VLAN, VLAN₃, to this client. The setting information is as below.

```

root@localhost/etc/raddd/sql/postgresql
[root@localhost postgresql]# echo "insert into radgroupreply (groupname,attribute,op,value)
values('testgroup','Tunnel-Private-Group-Id',':','=','3');" | psql -U radius radius
Password for user radius:
INSERT 0 1
[root@localhost postgresql]# echo "insert into radgroupreply (groupname,attribute,op,value)
values('testgroup','Tunnel-Type',':','=','13');" | psql -U radius radius
Password for user radius:
INSERT 0 1
[root@localhost postgresql]# echo "insert into radgroupreply (groupname,attribute,op,value)
values('testgroup','Tunnel-Medium-Type',':','=','6');" | psql -U radius radius
Password for user radius:
INSERT 0 1
[root@localhost postgresql]# echo "select * from radgroupreply;" | psql -U radius radius
Password for user radius:
 id | groupname | attribute | op | value
-----+-----+-----+---+-----
  7 | testgroup | Tunnel-Private-Group-Id | := | 3
  8 | testgroup | Tunnel-Type | := | 13
  9 | testgroup | Tunnel-Medium-Type | := | 6
(3 rows)

[root@localhost postgresql]#

```

5-2-5. Create accounts in the database.

Please use command below to create accounts (username/ password/ MAC address) in database for testing users.

The values need to change:

- Username: Define by user.
- Groupname: Define by user. We can only define one default vlan in the demo scenario.

```

-----
echo "insert into radusergroup (username,groupname,priority)
values('username','groupname','1');" | psql -U radius radius
-----

```

The values need to change:

- Username: Define by user.
- Value: Password for user

```

-----
echo "insert into radcheck (username,attribute,op,value) values ('test','Cleartext-
Password',':','=','test');" | psql -U radius radius
-----

```

The values need to change:

- Username: Define by user.
- Macaddr: MAC address of device
- Vlanid: Define by user

```
-----
echo "insert into radmacvlan (username,callingstationid,tunnelprivategroupid)
values('username','macaddr','vlanid');" | psql -U radius radius
-----
```

In this case, set the username/ password are as test/ test. The MAC address is the one of the corporate-provided NB (08:11:96:71:00:00). While three factors are matched, the RADIUS assign attribute VLAN2 to this client. The setting information is as below.

```

root@localhost/etc/radbb/sql/postgresql
[root@localhost postgresql]# echo "insert into radusergroup(username,groupname,priority) values('test','testgroup','1');" | psql -U radius radius
Password for user radius:
INSERT 0 1
[root@localhost postgresql]# echo "insert into radcheck(username,attribute,op,value) values('test','Cleartext-Password',':','=','test');" | psql -U radius radius
Password for user radius:
INSERT 0 1
[root@localhost postgresql]# echo "insert into radmacvlan(username,callingstationid,tunnelprivategroupid) values('test','08-11-96-71-00-00','2');" | psql -U radius radius
Password for user radius:
INSERT 0 1
[root@localhost postgresql]# echo "select * from radusergroup;" | psql -U radius radius
Password for user radius:
username | groupname | priority
-----+-----+-----
test     | testgroup | 1
(1 row)

[root@localhost postgresql]# echo "select * from radcheck;" | psql -U radius radius
Password for user radius:
id | username | attribute | op | value
---+-----+-----+---+-----
4  | test    | Cleartext-Password | := | test
(1 row)

[root@localhost postgresql]# echo "select * from radmacvlan;" | psql -U radius radius
Password for user radius:
username | callingstationid | tunnelprivategroupid
-----+-----+-----
test     | 08-11-96-71-00-00 | 2
(1 row)

[root@localhost postgresql]#

```

6. Stop the firewall process on FreeRadius server

```
-----
service firewalld disable
```

```
service firewalld stop
```

7. Start FreeRADIUS service

7-1. Enable and start FreeRADIUS service

Use the following commands to enable and start FreeRADIUS service

```
service radiusd enable
service radiusd start
```

7-2. Test FreeRADIUS

Use the tool **radtest** of FreeRADIUS to check if FreeRADIUS run well. The example command is as below.

```
radtest username password radius_ip o shared_secret
```

If the test is passed, it will show Access-Accept as below:

```

root@localhost:~#
ERROR: syntax error at or near "="
LINE 1: update radcheck attribute='Cleartext-Password' where id='2';
                                     ^
radius=> update radcheck set attribute='Cleartext-Password' where id='2';
UPDATE 1
radius=> select * from radcheck;
  id | username | attribute | op | value
-----+-----+-----+---+-----
   2 | test    | Cleartext-Password | := | test
(1 row)

radius=> \q
-bash-4.2$ exit
logout
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# radtest test test localhost 0 testing123
Sending Access-Request of id 172 to 127.0.0.1 port 1812
  User-Name = "test"
  User-Password = "test"
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 0
  Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=172, length=35
  Tunnel-Private-Group-Id:0 = "3"
  Tunnel-Type:0 = VLAN
  Tunnel-Medium-Type:0 = IEEE-802
[root@localhost ~]#
    
```

8. Post check after installation with RADIUS client

8-1. Download the FreeRadius client

There are many FreeRadius clients can be used for testing. The example in below is using **NTRadPing** which is downloaded from internet.

7-2. Install the RADIUS client in your laptop which running with Win7. After installed, you can configure RADIUS client through GUI.

Set few parameters when before start testing.

RADIUS Server/port: 192.168.10.15

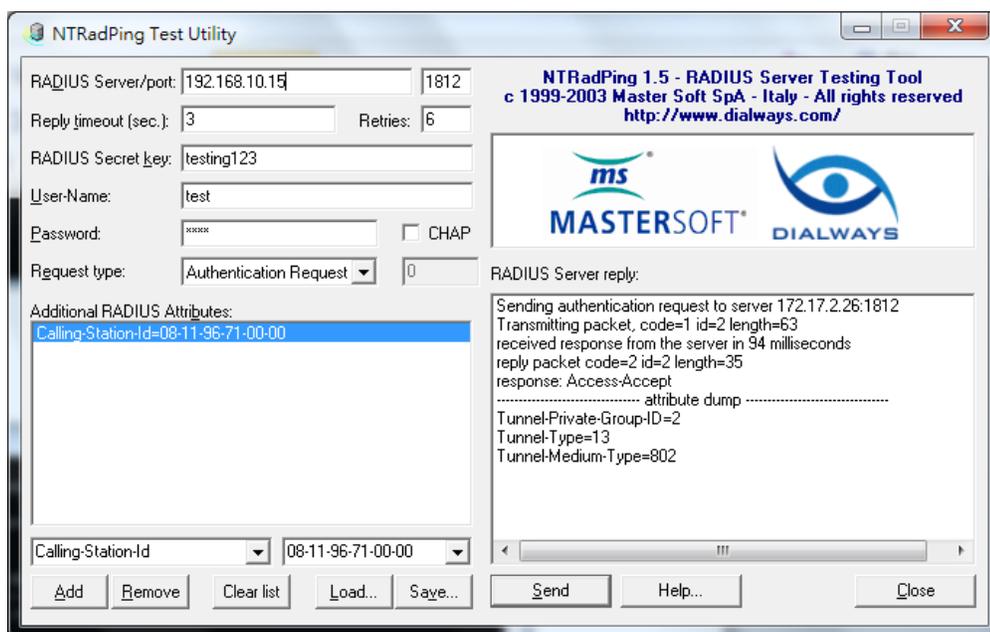
Port: No need to change, default is 1812.

RADIUS Secret Key: Define by user.

User Name/Password: Define by user.

Additional RADIUS Attributes: Please select Calling-Station-Id in the left and input the MAC Address of your device in the right.

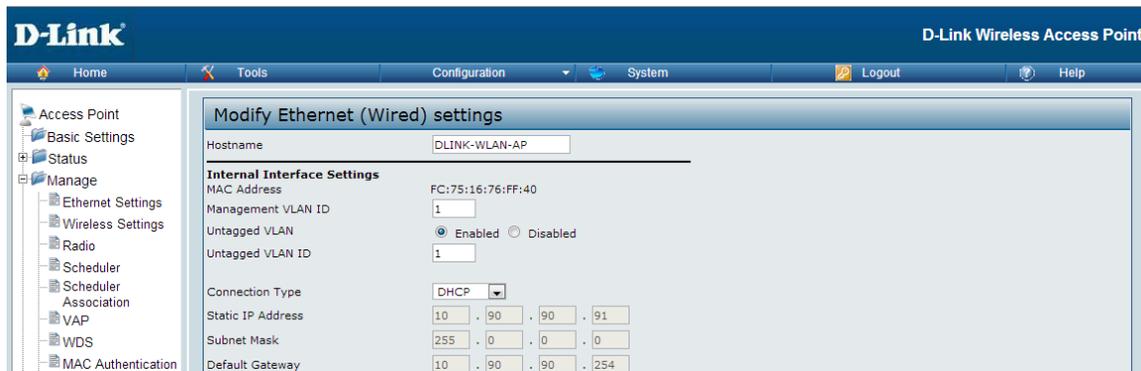
Click Send to send the Authentication Request to the RADIUS server, you can find the reply from RADIUS server in RADIUS Server reply window.



Configuration Steps (Unified Access Point, DWL-6600AP)

9. Set up UAP device IP address. Either enables Connection Type as DHCP getting IP address of VLAN1 subnet (192.168.10.0/24) from DSR-500N, or configure with a static IP address in the VLAN1 subnet. If set static IP address, make sure the IP does not duplicate with DSR-500N interface IPs and RADIUS server IP.

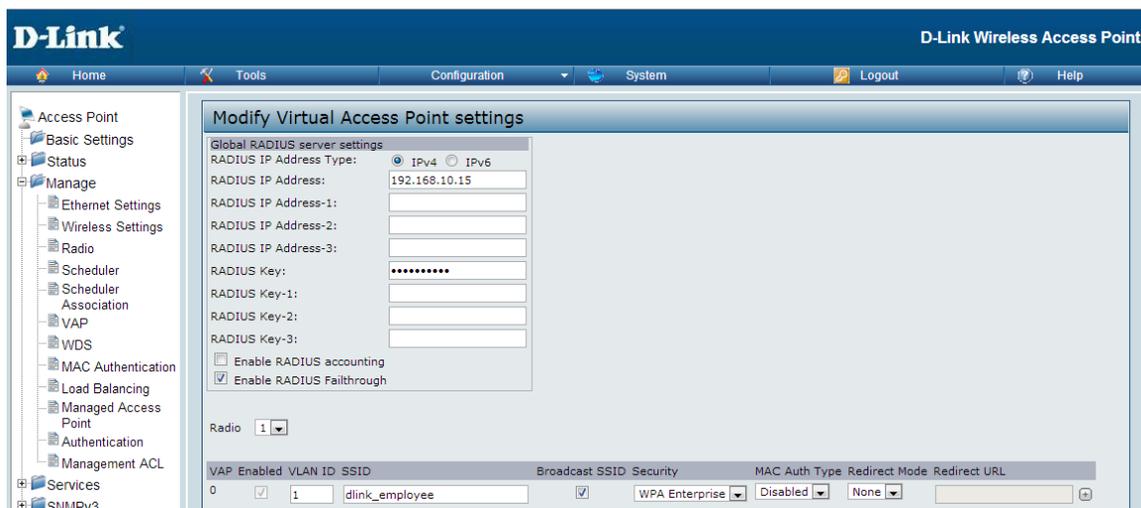
Navigate to Access Point> Manage> Ethernet Setting. Select DHCP of Connection Type.



10. Set up a VAP and RADIUS server. Navigate to Access Point> Manage> VAP.

2-1. Set RADIUS server. Fill in RADIUS IP Address and RADIUS key.

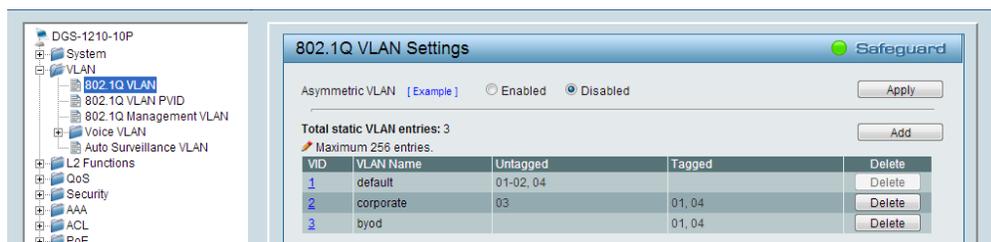
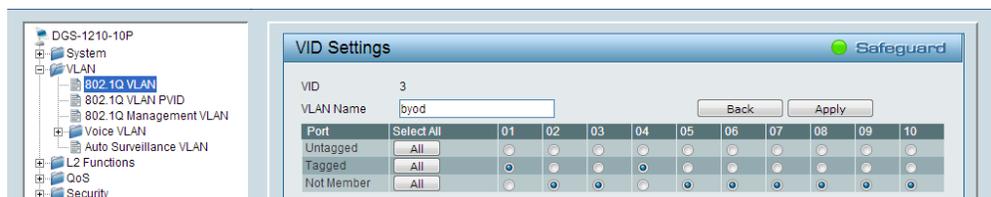
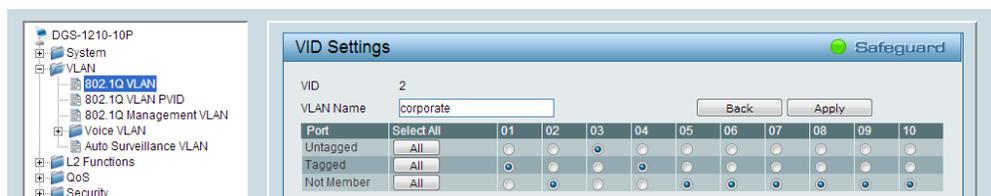
2-2. Create or modify the VAP dlink_employee. Assign VLAN1 on this SSID. Enable Security WAP Enterprise.



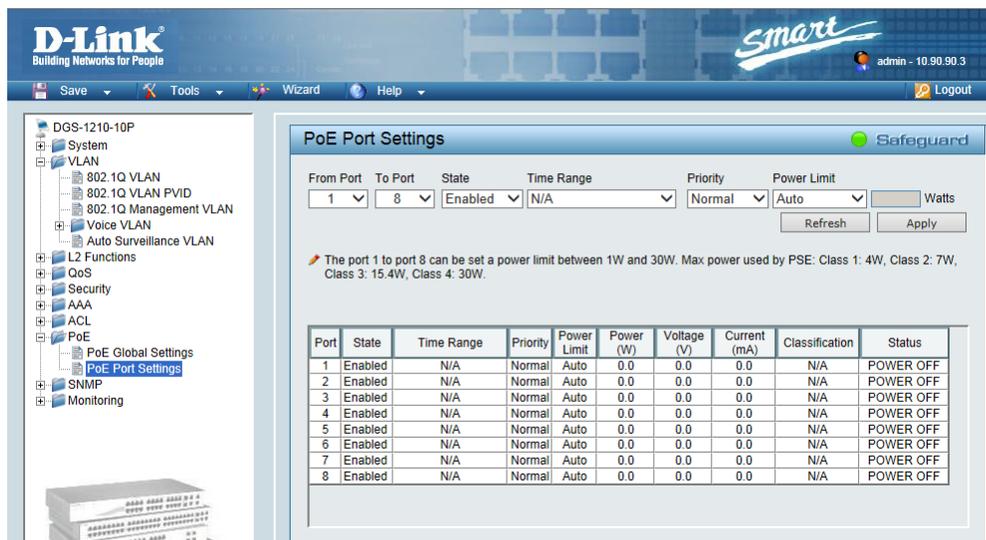
Configuration Steps (DGS-1210)

1. Set up VLANs based on the network architecture. Create three VLANs. VLAN1 is the default VLAN for AP management and external RADIUS server, VLAN2 is for the user using corporate-provided NB with full access on internal resources (for example, internet and printer), and VLAN3 is for the user using private NB with limited access (for example, internet). As DWC-1000 VLAN1 is un-tag VLAN, set VLAN1 as un-tag VLAN on switch. The VLAN table is as below.

	Port1	Port2	Port3	Port4
VLAN1	Un-tag	Un-tag	-	Un-tag
VLAN2	Tag	-	Un-tag	Tag
VLAN3	Tag	-	-	Tag



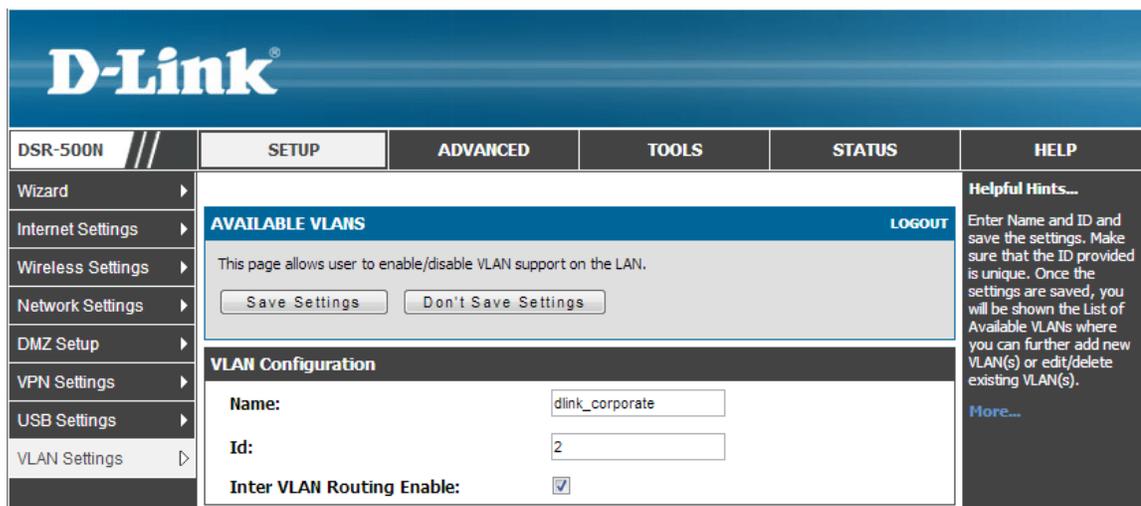
- (Option) Enable PoE on the ports which connect with APs if needed. In default, all ports are enabled auto PoE detection.



Configuration Steps (DSR-500N)

- Set up VLANs based on the network architecture. Create three VLANs. VLAN1 is the default VLAN for AP management and external RADIUS server, VLAN2 is for the user using corporate-provided NB with full access on internal resources (for example, internet and printer), and VLAN3 is for the user using private NB with limited access (for example, internet).

1-1. Set up VLAN2 and VLAN3. Navigate to SETUP> VLAN Settings> Available VLANs.



How to Configure a BYOD Environment with the DWC-1000

DSR-500N	SETUP	ADVANCED	TOOLS	STATUS	HELP
Wizard					Helpful Hints... Enter Name and ID and save the settings. Make sure that the ID provided is unique. Once the settings are saved, you will be shown the List of Available VLANs where you can further add new VLAN(s) or edit/delete existing VLAN(s). More...
Internet Settings	AVAILABLE VLANS LOGOUT				
Wireless Settings	This page allows user to enable/disable VLAN support on the LAN. <input type="button" value="Save Settings"/> <input type="button" value="Don't Save Settings"/>				
Network Settings	VLAN Configuration				
DMZ Setup	Name: <input type="text" value="dlink_byod"/>				
VPN Settings	Id: <input type="text" value="3"/>				
USB Settings	Inter VLAN Routing Enable: <input checked="" type="checkbox"/>				
VLAN Settings					

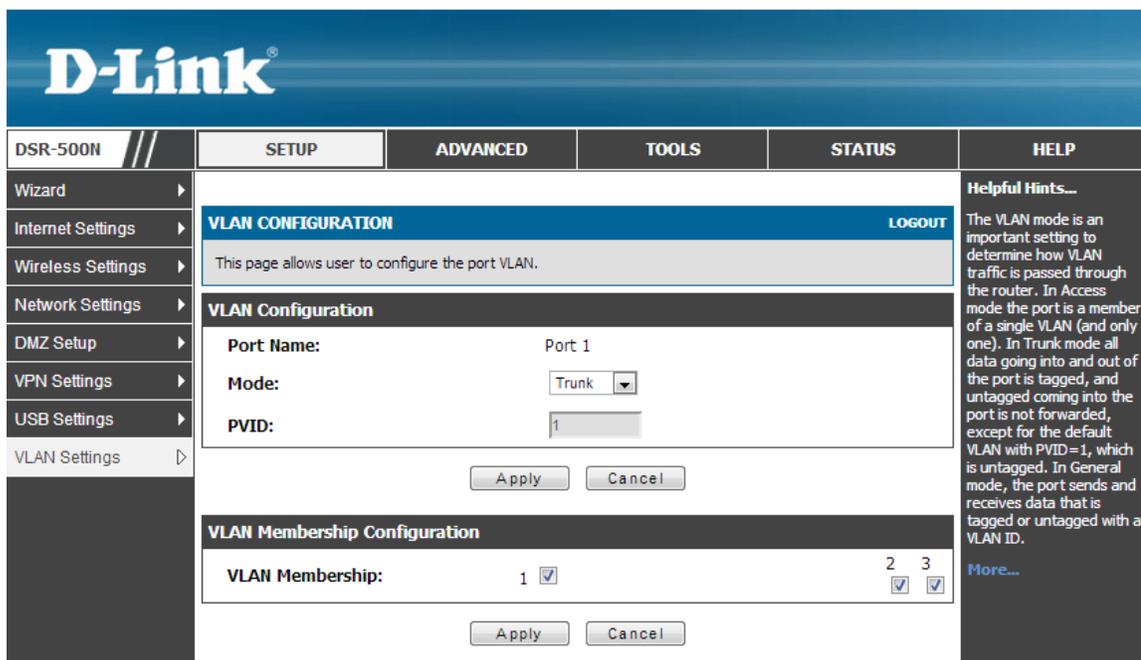
1-2. Enable DHCP server on default VLAN, VLAN2 and VLAN3. Navigate to SETUP> VLAN Settings> Multiple VLAN Subnets.

DSR-500N	SETUP	ADVANCED	TOOLS	STATUS	HELP
Wizard					Helpful Hints... By default, when you add a new VLAN, it is assigned an IP address of 192.168.2.1 with subnet-mask 255.255.255.0, the next added one is assigned 192.168.3.1 and so on. You can change the assigned IP address, subnet mask and many other options here. The only non-editable field in VLAN ID. More...
Internet Settings	MULTI VLAN SUBNET CONFIG LOGOUT				
Wireless Settings	This page shows the list of available multiple VLAN subnets. <input type="button" value="Save Settings"/> <input type="button" value="Don't Save Settings"/>				
Network Settings	MULTI VLAN SUBNET				
DMZ Setup	Vlan ID: <input type="text" value="1"/>				
VPN Settings	IP Address: <input type="text" value="192.168.10.1"/>				
USB Settings	Subnet Mask: <input type="text" value="255.255.255.0"/>				
VLAN Settings	DHCP				
	DHCP Mode: <input type="text" value="DHCP Server"/>				
	Domain Name: <input type="text" value="DLink"/>				
	Starting IP Address: <input type="text" value="192.168.10.100"/>				
	Ending IP Address: <input type="text" value="192.168.10.254"/>				

DSR-500N	SETUP	ADVANCED	TOOLS	STATUS	HELP
Wizard					Helpful Hints... By default, when you add a new VLAN, it is assigned an IP address of 192.168.2.1 with subnet-mask 255.255.255.0, the next added one is assigned 192.168.3.1 and so on. You can change the assigned IP address, subnet mask and many other options here. The only non-editable field in VLAN ID. More...
Internet Settings	MULTI VLAN SUBNET CONFIG LOGOUT				
Wireless Settings	This page shows the list of available multiple VLAN subnets.				
Network Settings	<input type="button" value="Save Settings"/> <input type="button" value="Don't Save Settings"/>				
DMZ Setup	MULTI VLAN SUBNET				
VPN Settings	Vlan ID: 2				
USB Settings	IP Address: <input type="text" value="192.168.0.1"/>				
VLAN Settings	Subnet Mask: <input type="text" value="255.255.255.0"/>				
	DHCP				
	DHCP Mode: <input type="text" value="DHCP Server"/>				
	Domain Name: <input type="text" value="DLink"/>				
	Starting IP Address: <input type="text" value="192.168.0.100"/>				
	Ending IP Address: <input type="text" value="192.168.0.254"/>				

DSR-500N	SETUP	ADVANCED	TOOLS	STATUS	HELP
Wizard					Helpful Hints... By default, when you add a new VLAN, it is assigned an IP address of 192.168.2.1 with subnet-mask 255.255.255.0, the next added one is assigned 192.168.3.1 and so on. You can change the assigned IP address, subnet mask and many other options here. The only non-editable field in VLAN ID. More...
Internet Settings	MULTI VLAN SUBNET CONFIG LOGOUT				
Wireless Settings	This page shows the list of available multiple VLAN subnets.				
Network Settings	<input type="button" value="Save Settings"/> <input type="button" value="Don't Save Settings"/>				
DMZ Setup	MULTI VLAN SUBNET				
VPN Settings	Vlan ID: 3				
USB Settings	IP Address: <input type="text" value="192.168.1.1"/>				
VLAN Settings	Subnet Mask: <input type="text" value="255.255.255.0"/>				
	DHCP				
	DHCP Mode: <input type="text" value="DHCP Server"/>				
	Domain Name: <input type="text" value="DLink"/>				
	Starting IP Address: <input type="text" value="192.168.1.100"/>				
	Ending IP Address: <input type="text" value="192.168.1.254"/>				

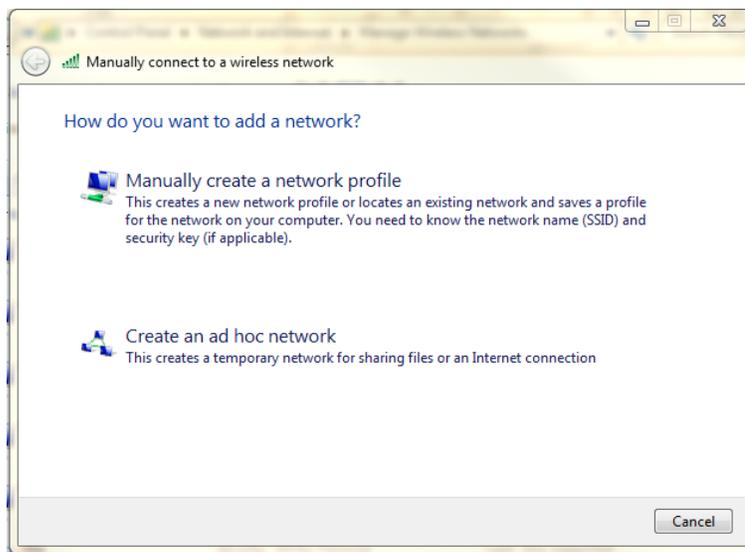
1-3. Associate VLAN1 to 3 in Trunk mode on Port1.



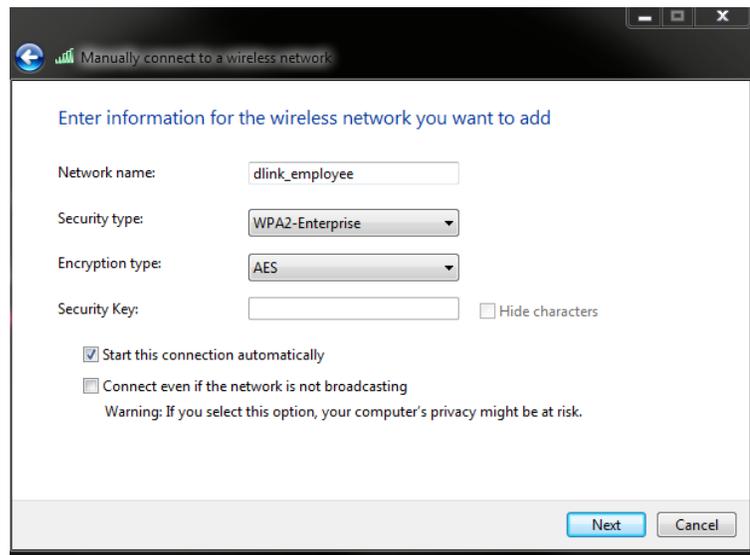
Configuration Steps (Notebook, Microsoft/ Win7)

1. Set up wireless security.

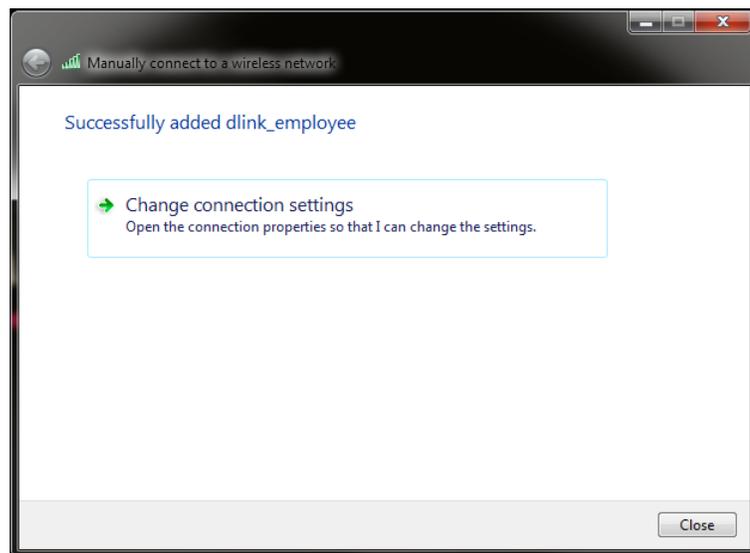
1-1. Navigate to START> Control Panel> Network and Sharing Center. Click "Manage wireless network". Click "Add" to add a new wireless network. Select "Manually create a network profile".



1-2. Fill in the network name. Select security type as WPA2-Enterprise. Select the Encryption is AES. Click "Next".



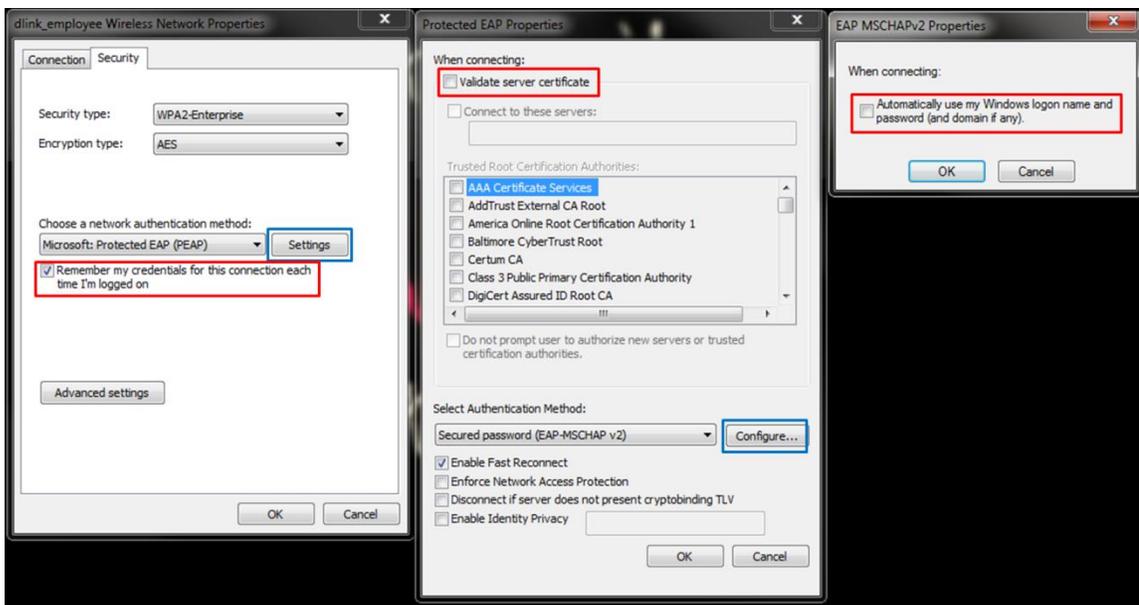
1-3. Click "Change connection settings".



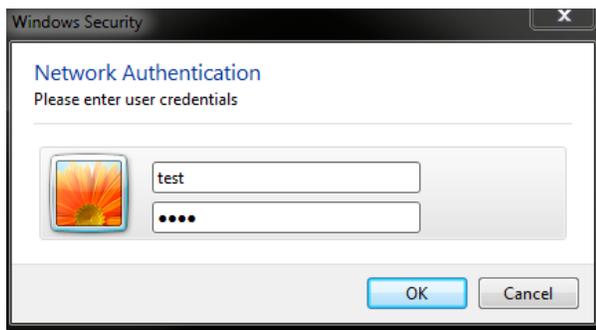
1-3-1. Click tab "Security". (Option) Tick "Remember my credentials for the connection each time I'm logged on" to keep the username/ password information in the computer.

1-3-2. Click "Settings" of "Choose a network authentication method". Un-check "Validate server certification".

1-3-3. Click "Configure.." of Select Authentication Method". (Option) Un-check "Automatically use my Windows logon name and password (and domain if any)" if the username/ password is not the same as Windows logon information.

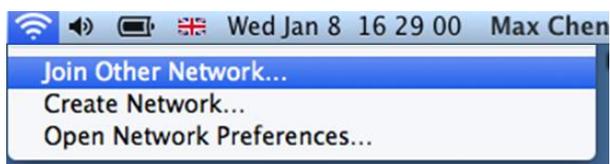


2. Connect the wireless. Insert the username and password.



Configuration Steps (Notebook, Apple/ iOS10)

1. Set up wireless security. Click WiFi and select "Join Other Network...".



- Fill in the network name. Select security type as WPA2-Enterprise. Click "Join".



- Click "Cancel" on Verify Certificate.



Proof of Concept

The NB with MAC 08:11:96:71, which is the corporate-provided device, is assigned VLAN2 after pass the authentication. The NB would get IP address of VLAN2 subnet (for example, 192.168.0.x). It can access resources on VLAN2, for example, printer and internet.

The NB with MAC 00:13:02:69:7F:E9, which is the private device, even use the same username/ password, as the MAC address doesn't match with the database, it is assigned VLAN3 after pass authentication and get IP address of VLAN3 subnet (for example, 192.168.1.x). It can access resources on VLAN3, for example, internet.

D-Link®

WWW.DLINK.COM

D-Link, D-Link logo, D-Link sub brand logos and D-Link product trademarks are trademarks or registered trademarks of D-Link Corporation and its subsidiaries. All other third party marks mentioned herein are trademarks of the respective owners.

Copyright © 2014 D-Link Corporation. All Rights Reserved.