

An abstract graphic consisting of several thick, curved lines that originate from a central point on the left and radiate outwards towards the right. The lines are in various colors: light blue, yellow, orange, and red. Some lines are straight, while others are curved or spiral. The overall effect is dynamic and modern.

**Raising The Call: Integrating Aruba Wireless  
LANs With Microsoft Lync's Emergency  
Call Architecture**

## Table of Contents

<b>Introduction</b>	<b>3</b>
<b>Microsoft Lync E9-1-1 architecture</b>	<b>3</b>
<b>The Aruba solution for Lync emergency calls</b>	<b>4</b>
<b>Administration considerations with the Aruba-Lync solution</b>	<b>5</b>
<b>Access points, radios and BSSIDs</b>	<b>5</b>
<b>Other architectures for E9-1-1 and the WLAN</b>	<b>6</b>
<b>E9-1-1 and accuracy requirements</b>	<b>6</b>
<b>Conclusion</b>	<b>7</b>
<b>Appendix A – A sample PowerShell script</b>	<b>7</b>
<b>About Aruba Networks, Inc.</b>	<b>13</b>

## Introduction

As traditional TDM PBXs have progressively been replaced with IP-PBXs and unified communications systems, vendors have been called upon to integrate these systems with the North American E9-1-1 emergency call system. Among other characteristics, E9-1-1 requires automatic location identification of the caller who dialed 911.

With a traditional PBX, the physical location of each TDM desk phone was associated with its PBX port at the time of installation. Assuming that association was unchanged, the mapping of a phone number to a location was static and easily documented. IP PBXs use IP desk phones that can be connected to any Ethernet port. To locate the position of the caller, E9-1-1 systems automatically associate the phone's IP switch port, discovered using the LLDP-MED protocol, with a known physical location documented when the switch and cabling was installed. Microsoft Lync uses the IP switch location information for wired client devices.

Voice-over-Wi-Fi-enabled PCs and handsets are not amenable to this model of location identification. Since they are mobile and not tied to any specific port or switch, wireless IP phones cannot be pinpointed by switch port. While it is true that most Wi-Fi access points are associated with a switch port, the IP subnets on which they reside can range over wide geographic distances. Moreover, the LLDP-MED discovery method is not available to wireless LAN clients. The upshot is that identifying the location of a 911 caller over a wireless LAN requires a new approach.

Microsoft has an elegant solution for locating Lync 2010 clients on wired LANs. The Location Information Server (LIS) maintains a central database that maps IP subnets, LAN switch ports and wireless LAN access point (AP) BSSIDs to street addresses in the format used by the E9-1-1 network. Wired devices are well-served by this architecture but maintaining database entries for APs can be time-consuming and error-prone if a manual system is used. This is because the placement and configuration of enterprise access points often changes with the adds, moves and changes made to offices. There is consequently the need for an automated system to keep the LIS database synchronized with changes in the wireless LAN topology.

Aruba has developed a solution that extracts from its AirWave wireless LAN management suite all access point instances and location information, and then uses a script to write this data into the LIS. By setting this script to run at regular intervals, one can ensure that the LIS is brought up to date with changes in the wireless LAN, and thereby maintain the integrity of the E9-1-1 related location information. Script integration uses existing interfaces with the LIS, and requires no new features in either Microsoft Lync or the AirWave tool.

## Microsoft Lync E9-1-1 architecture

The LIS is a central database for location information within the Lync architecture. The database maintains a street address for each LAN IP subnet, port and access point used in the network. During an emergency call, a look-up to the database extracts the street address, which is then used by an external E9-1-1 routing service to deliver the call to the pertinent public service answering point (PSAP) with sufficient address information to dispatch emergency services to the correct location.

The LIS is populated using Microsoft PowerShell commands, which are defined at <http://technet.microsoft.com/en-us/library/gg413069.aspx>. Wired switches are covered using IP subnets, chassis ID or ports. Wireless access points are identified by BSSID.

While Microsoft allows individual configuration or bulk configuration using a CSV file, the challenge with a wireless LAN is locating the data with which to construct such a command or configuration file. While tables mapping BSSIDs to AP locations may exist in a number of subsystems, assembling the data into files with which the LIS can be populated can be cumbersome.

Populating the LIS at initial installation is just the start of the administrative work: administrative procedures are also required to track every relevant change in the wireless LAN and BSSIDs so they are reflected in an updated LIS.

If the LIS falls out of synchronizatin with the wireless LAN, emergency calls made from wireless devices may not be associated with the correct location, or the location may not be in the LIS database at all. If the BSSID is not in the LIS database, the LIS will attempt other lookups, for instance using the IP subnet, but this usually covers a wider area than the BSSID and so will be less accurate. The consequence could be delayed E9-1-1 response.

In the absence of a protocol that instantly identifies changes to BSSID-location tables, the next best solution is to automatically update the LIS on a regular basis so that it maintains a current view of the network. It is this solution that is presented in Figure 1.

## The Aruba solution for Lync emergency calls

Mobility is a primary attribute of a Wi-Fi network, and wireless LAN vendors have developed systems that can locate all Wi-Fi clients which are associated. Aruba's multi-vendor location engine, AirWave VisualRF™, supports wireless LAN solutions from Aruba and many other vendors. Providing an accurate view of the entire network, VisualRF automatically generates maps of the RF environment and the underlying wired topology, showing what the network looks like in real time. VisualRF builds this map using RF measurements gathered from active, authorized wireless access points and controllers, without the need for separate location appliances. AirWave VisualRF displays exactly who is on the network, where they are, and how the network is performing.

The basis for all location in AirWave is a diagram of the floor plan. Each area to be covered by Wi-Fi is defined as a "site", and a floor plan diagram can be uploaded in any popular format including .jpg, .gif and .dwg. The floor plan provides a visual reference for network administrators, since many AirWave operations use coordinates on the floor plan as parameters. AirWave administrators set up the system by dragging and dropping access point icons to the correct location on the floor plan as they are provisioned. Typically each floor of a building has a separate floor plan. Every access point it mapped to its associated floor plan using x- and y-coordinates measured from a datum on the floor plan diagram.

AirWave maintains a hierarchy of "campus", "building", "floor/site" for every network, and each floorplan inherits these attributes. This data is maintained in AirWave's database, and from this data it is possible to identify every access point by its building (and hence street address), floor, and its position on the floor plan

AirWave provides an XML-over-https API through which external systems and applications can look-up location information. Various query formats are supported, and the one used for Lync integration discovers all access points in use together with their associated building and floor. Each AirWave building entry has an "address" field, making it possible to associate each BSSID with a street address and floor. Making this information available to LIS over the API enables Lync to locate wireless LAN E9-1-1 calls.

The mechanism used to transfer this data via the API is a Microsoft PowerShell script. The script can be run on the LIS server itself or elsewhere in the network, and when activated it first logs into the AirWave 'Visual RF' module. Given a campus ID, the script extracts the list of access points, including their associate street addresses and floors, and BSSIDs, and then writes each into the LIS using documented commands.

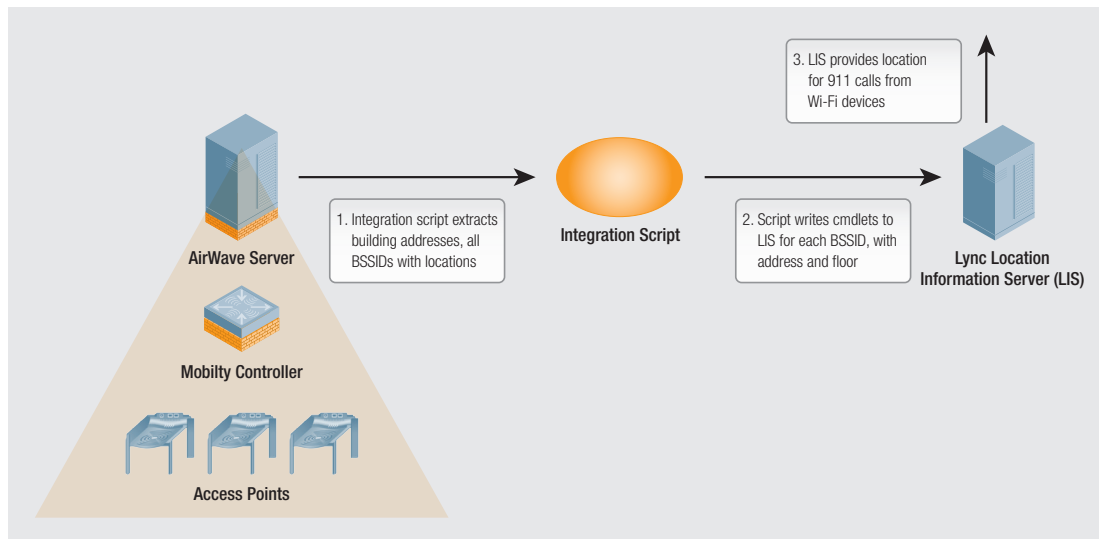


Figure 1: Automatically updating the LIS access point database from AirWave

## Administration considerations with the Aruba-Lync solution

By running the script at regular intervals, say weekly, the LIS will remain in synch with the wireless LAN with minimal administrative overhead, or operator-induced errors. Since the building address will not change and is entered in only one place, the AirWave management station, it can be validated for compliance with the Master Street Address Guide (MSAG) format and left alone thereafter. Only the BSSIDs and their locations in the building will change.

The initial set-up requires that the address for each building be entered into the AirWave database in a format consistent with the MSAG and the PowerShell script. The script can take any form so long as it can extract the address into Microsoft's schema. An example script using Java Script Object Notation (JSON) follows below:

```
"HouseNumber": "1344", "StreetName": "Crossman Avenue", "City": "Sunnyvale", "State": "California": "PostalCode": "94089", "Country": "USA"
```

The AirWave address field is currently an unstructured string, however it could in the future be structured to enforce the Microsoft address format or MSAG-compliant addresses. Microsoft's LIS address format can be found at <http://technet.microsoft.com/en-us/library/gg413069.aspx> and the command to validate addresses in the LIS against the MSAG can be found at <http://technet.microsoft.com/en-us/library/gg412808.aspx>. This validation should be run when a new address (new building) is added in AirWave, but should not be necessary to run it thereafter.

The script automatically populates the "location" parameter of the address with the floor number, using AirWave's internal convention for using the floor number of the building. The LIS includes a "description" field that can be populated based on the x, y coordinates of the access point and these mappings can be easily changed by modifying the script.

## Access points, radios and BSSIDs

This section reviews the relationship between access points, radios, and BSSIDs. It is important to understand the use and meaning of these terms in order to make sense of the location script.

An "access point" is a physical device that is the part of the network infrastructure responsible for sending and receiving Wi-Fi frames. An access point contains one or more radios, where a radio is a transmitter/receiver serving a single band/channel. Most enterprise access points today comprise two radios, one for the 2.4GHz band and the other for 5GHz, and they can serve clients in both bands simultaneously. In AirWave, an icon for an access point is dragged and dropped to the appropriate point on the floor plan to indicate its location.

A service set identifier (SSID) is a name that identifies a particular wireless LAN. Each access point radio supports a number of virtual APs, usually synonymous with SSIDs. Thus a single radio could transmit SSIDs for “employee,” “contractor,” and “guest services simultaneously on the same channel. Each virtual AP has a basic service set identifier (BSSID) that uniquely identifies the MAC address of the access point. Each physical access point can support many BSSIDs, which is commonly the case in enterprise wireless LANs. For example, assigning three SSIDs over two bands will result in a total of six SSIDs per access point.

Each access point is factory-configured for a given base BSSID per radio, using the same AB:CD:EF:01:23:45 hex format used for MAC addresses. While by convention a newly-configured virtual AP will take the next available BSSID, counting up from the base, this is not guaranteed. However, all BSSIDs will be in the range of 0 to 15 or 0 to 7 above the base BSSID, and the base BSSID for a given radio will never change.

The PowerShell script makes a list of all possible BSSIDs for each AP on a campus and then writes them into the LIS. This means that re-configuration of an existing AP will not affect the LIS: every BSSID that a client might report will already be in the LIS database. Unless access points are moved, or new ones added to the network, the script will not need to be run again to keep the LIS synchronized with the wireless LAN.

Also note that the script does not delete unused BSSIDs. If existing BSSIDs were in the LIS before and their details are changed, the records will be over-written, but if access points were removed from the network, their entries will not be deleted by the script.

## Other architectures for E9-1-1 and the WLAN

While the current Lync E9-1-1 architecture is centered on using the LIS for location, Microsoft may introduce additional options in the future. Aruba's wireless LAN supports other means for ensuring an accurate address is delivered to the E9-1-1 system when Wi-Fi clients make emergency calls.

The fall-back method for location in nearly all IP-PBX/UC systems is to parse the IP subnet and match it to an address. This is an acceptable method for most wired networks, but with wireless LANs, especially when voice or video service is offered, mobility zones can extend over wide areas to preclude the disruption to voice calls caused by repeatedly assigning new IP addresses. While the IP address may not be a very accurate guide to the caller's location, the method can be used for an Aruba wireless LAN running Lync and using the LIS and/or third-party 911 servers.

Aruba has other features for emergency call location, primarily for use with other IP-PBXs and 911 manager servers. In this architecture, voice devices are tracked as they traverse the wireless LAN, and whenever such a client moves to a new access point, the 911 manager is updated with its new access point. With a pre-configured 911 server that maps access points to addresses and zones (ELIN-based system), the 911 manager now has a view of the client's location, that is as accurate as that for wired devices. Aruba has demonstrated integration with the RedSky 911 manager using this API.

## E9-1-1 and accuracy requirements

E9-1-1 requirements are legislated on a state-by-state basis in the U.S. RedSky has compiled an excellent set of legislative guides and posted them at [http://www.redskytech.com/e911\\_information\\_center/e911\\_legislation/](http://www.redskytech.com/e911_information_center/e911_legislation/). Most E9-1-1 laws require location to within a sizeable ~35,000 square feet. Enterprises implementing E9-1-1 focus on providing the correct street address for the building entrance, and the floor from which the call is originated. Some add supplementary information such as “north-west corner” to localize further.

There is a limit on the amount of information that can be transmitted to the current 911 public safety dispatch center, and the above is usually deemed an appropriate level of location accuracy. However, large enterprises and universities with their own 911 dispatch centers, often desire more detail, sometimes down to the office level.

Aruba can meet these requirements within the limitation that wireless users are mobile and therefore locations are best estimates with some level of uncertainty. An Aruba wireless LAN offers three levels of location accuracy. Least accurate is the IP Subnet mapping as described above. A voice subnet might cover several floors, or even buildings. For greater accuracy, the access point to which the client is associated can be located. Wi-Fi clients generally choose the strongest signal, and hence the closest access point, but sometimes they may find a better match with a more distantly located access point. The 'associated access point' method typically offers location to within 30 meters, but there will be statistical variations in accuracy accounting for clients that associated with access points on other floors. The advantage of defining location by access point is that access points are stationary, making them suitable for the LIS database.

The third location method leverages the AirWave suite, which continuously locates all clients in the wireless LAN using multiple techniques including measuring the signal strength of client transmissions at different access points. This location method has a lower chance of choosing the wrong floor of the building and yields greater accuracy. AirWave views of client location can be superimposed on building floor plans, making caller location an easier task for first responders and dispatch centers. An AirWave based solution is outside the scope of public E9-1-1 architecture, as it involves giving responders access to an AirWave client screen, but it offers better accuracy and more fine grained address and location information.

## Conclusion

The process of populating the LIS database with location data is the choice of the Lync customer, and there are several ways in which this can be accomplished in connection with a wireless LAN. Lync offers cmdlets to enter BSSIDs in the LIS location database, however if a site has thousands of Wi-Fi access points, the data entry process can be time consuming to implement.

The AirWave suite maintains a database containing the building, floor and coordinates of every BSSID under management. Using an existing XML-over-https API, the Lync LIS database can be populated from the AirWave database. Doing so allows LIS to leverage the client device's associated BSSID to return the correct street address and floor for the caller.

The LIS must be periodically synchronized with the wireless LAN, to reflect network changes following initial configuration. For example, as new access points are added, or existing access points are moved within the wireless LAN, these changes must be reflected in the LIS. Aruba's PowerShell script automatically updates the LIS at regular intervals to ensure that the wireless LAN and the LIS are synchronized. The script can be easily customized for needs specific to a particular customer.

## Appendix A – A sample PowerShell script

The script below was developed for use with the AirWave suite. Since it only extracts data, it does not intrude on the operation of an AirWave installation.

The script first establishes an https session with the Visual RF service running on the AirWave server. The example below uses basic authentication. If used with AirWave 7.3 or higher the script will have to be modified to account for cookie-based authentication.

Following authentication, the script discovers all buildings listed in the AirWave database and extracts their addresses. Since the address field in AirWave is currently an unstructured string, we developed a protocol to uniquely identify attributes using the Microsoft address schema. For this script example, the address should be entered in this JSON format:

```
"HouseNumber": "1344", "StreetName": "Crossman Avenue", "City": "Sunnyvale", "State": "California": "PostalCode": "94089", "Country": "USA"
```

Alternate formats can be used so long as the script is set up to parse correctly and construct a cmdlet.

The script next extracts all access points and radios configured for each floor of each building on the campus. For each radio, it determines how many SSIDs are supported, and then matches all possible BSSIDs and to their associated building addresses and floors. The script then constructs a PowerShell cmdlet for writing the BSSID into the LIS database.

Note that the script below constructs the command to write to the LIS, and just writes the command to the screen.

The example below is set up as a function called "WriteAPLocations" which takes the following parameters:

- airwave host in the format 'test.airwave.com'
- username, e.g. 'demo'
- password, e.g. 'demo'
- Campus number. Since an AirWave installation can include multiple campuses, this defines which campus is to be extracted. The first campus will be index 0.

```
Function WriteAPLocations ($airwavehost, $username, $password, [int]$campusindex){
    # #Set up login and url, build web client
    $webClient = New-Object system.Net.Webclient
    $urlstring = "https://" + $airwavehost + "/visualrf/campus.xml?buildings=1&sites=1&aps=1"
    Write-host $urlstring
    $url = $urlstring
    $webClient.credentials = new-object System.Net.NetworkCredential($username, $password)
    $webClient.Headers.Add("user-agent", "PowerShell")
    # #Query campus information, get down to buildings
    $data = $webClient.OpenRead($url)
    $reader = new-object system.io.StreamReader $data
    [string] $s = $reader.ReadToEnd()
    $data.Close()
    $reader.Close()
    $xmls= [xml]($s)

    ##Now we have a big xml document with campuses, buildings, sites, aps & radios...
    $ctr-campus = 0
    foreach($campus in $xmls.campuses.campus){
        if ($ctr-campus -eq $campusindex){
            Write-host "`nCampusindex_ " $campusindex "Campus name_" $campus.name
            $ctr-building = 0
            foreach($building in $campus.building){
                write-host "`nBuilding index_ " $ctr-building "Building name_" $building.name
                #foreach($building in $xmls.campuses.campus[0].building){
                #write-host "Campus name_" $xmls.campuses.campus[0].name

                ##Parsing the building address string in JSON
                #$formatbldgaddress = $campus.name + $building.name
                $formatbldgaddress = " "
                $a = $building.address
                #Write-host "`nBuilding address string_ " $a
                $b = $a.Split(",")
                #Write-host $b
            }
        }
    }
}
```

```

foreach ($i in $b){
    $c = $i.split(":")
    $t0 = $c[0]
    #Sinda = $t0.IndexOf("''")
    #Sindb = $t0.IndexOf("''", $inda + 1)
    #$property = $t0.substring($inda + 1, $indb - 1)
    $t0 = $t0.trim()
    $property = $t0.substring(1, $t0.length - 2)
    $t1 = $c[1]
    #Sindc = $t1.IndexOf("`")
    #Sindd = $t1.IndexOf("`", $indc + 1)
    #$value = $t1.substring($indc+ 1, $indd - 1)
    $t1 = $t1.Trim()
    $value = $t1.substring(1, $t1.length - 2)
    Write-host $property ":" $value
    $formatbldgaddress = $formatbldgaddress + " -$property " + $value
}
Write-host "`nBuilding address_ " $formatbldgaddress

##Site level extraction
$ctrsite = 0
foreach($site in $building.site){
    #write-host "Building index_" $ctrbuilding " Site index_" $ctrsite
    #write-host "Building name_"$building.name " Building address_" $building.address
    #write-host "Floor_" $site.floor
    #write-host " "
    ## Add floor number to formatted address as location
    $formatbldgsiteaddress = $formatbldgaddress + " -Location Floor " + $site.floor
    Write-host "`nSite address_"$formatbldgsiteaddress
    $ctrtrap = 0
    foreach($ap in $site.ap){
        #check whether the AP supports 8 or 16 SSIDs per radio
        $ctrradio = 0
        $bssidsperradio = 16
        [string] $apmodel = $ap.model
        [string] $apmanufacturer = $ap.manufacturer
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("65")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("80")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("52")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("1200")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("DUO")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("85")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("102")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("1250")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("RAP-2")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("RAP 2")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("101")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("105")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("60P")) {$bssidsperradio = 8 }
        If( $apmanufacturer.contains("Aruba") -and $apmodel.contains("185")) {$bssidsperradio = 8 }
    }
}

```



**The output looks like this:**

Campusindex\_ 0 Campus name\_ Aruba Sunnyvale Campus

Building index\_ 0 Building name\_ 1344 Crossman

HouseNumber : 1344

StreetName : Crossman Avenue

City : Sunnyvale

State : California

PostalCode : 94089

Country : USA

Building address\_ -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California

-PostalCode 94089 -

Country USA

Site address\_ -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode

94089 -Coun

try USA -Location Floor 1.0

AP\_Name\_ 00:0b:86:64:f1:10 Radio\_ 0 Manufacturer\_ Aruba Model\_ AP 651 SSIDs\_Per\_Radio\_ 8 Base\_MAC\_ 00:0B:86:CF:11:

00

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:00 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:01 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:02 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:03 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:04 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:05 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:06 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

Set-CsLisWirelessAccessPoint -BSSID 00:0B:86:CF:11:07 -Description x:46.02,y:79.51 -HouseNumber 1344

-StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor

1.0

AP\_Name\_ AP125-fa:f0 Radio\_ 0 Manufacturer\_ Aruba Model\_ AP 125 SSIDs\_Per\_Radio\_ 16 Base\_MAC\_ 00:1A:1E:8F:AF:00

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:00 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:01 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:02 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:03 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:04 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:05 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:06 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:07 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:08 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:09 -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:0A -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:0B -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:0C -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:0D -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:0E -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

Set-CsLisWirelessAccessPoint -BSSID 00:1A:1E:8F:AF:0F -Description x:105.58,y:97.67 -HouseNumber 1344 -StreetName Crossman Avenue -City Sunnyvale -State California -PostalCode 94089 -Country USA -Location Floor 1.0

## About Aruba Networks, Inc.

Aruba Networks is a leading provider of next-generation network access solutions for the mobile enterprise. The company's Mobile Virtual Enterprise (MOVE) architecture unifies wired and wireless network infrastructures into one seamless access solution for corporate headquarters, mobile business professionals, remote workers and guests. This unified approach to access networks dramatically improves productivity and lowers capital and operational costs.

Listed on the NASDAQ and Russell 2000® Index, Aruba is based in Sunnyvale, California, and has operations throughout the Americas, Europe, Middle East, and Asia Pacific regions. To learn more, visit Aruba at [www.arubanetworks.com](http://www.arubanetworks.com). For real-time news updates follow Aruba on [Twitter](#) and [Facebook](#).



[www.arubanetworks.com](http://www.arubanetworks.com)

1344 Crossman Avenue. Sunnyvale, CA 94089

1-866-55-ARUBA | Tel. +1 408.227.4500 | Fax. +1 408.227.4550 | [info@arubanetworks.com](mailto:info@arubanetworks.com)