CHAPTER 4

Context-Awareness for Wired Devices

This chapter focuses on best practices to enable and use Cisco Context-Aware Services for wired endpoint location tracking in Unified Access solutions. It is intended as a guide to producing functional designs that incorporate the Cisco Mobility Services Engine (MSE) and Cisco Wireless Control System (WCS) in order to help fulfill common business needs in enterprise environments.

Note that while this chapter is comprehensive, it is first and foremost intended as a design considerations document and is not intended to be a configuration guide to Cisco Catalyst switches, the Mobility Services Engine, or the Wireless Control System. For comprehensive configuration and general deployment guidance, the reader should refer to the various in-depth configuration guides referenced throughout this chapter.

Introduction

What Are Context-Aware Services?

Context-Aware Services provide value-added functionality to business applications by capturing, integrating, and consolidating intelligence about users and their endpoint devices from various points in the network. Together with Unified Access wired and wireless networks, Cisco Context-Aware Services help enhance application functionality by making this information readily available via an established application programming interface.

Cisco Context-Aware software is a mobility service with these characteristics:

- Acts across multiple edge technologies, such as 802.11 wireless and 802.3 wired networks.
- Provides a value-add function across multiple network elements.
- Provides an interface to a value-add function for external applications and servers using a well-defined API.
- Adds intelligence to the network and enhances usability.
- Provides visibility into the network that applications and servers would not otherwise easily obtain.
- Integrates with other network services to deliver higher-order functionality and value.
- Can be managed using other Cisco tools or the mobility service API.
- Can be deployed across multiple engines to scale the function it provides.

Location Services associated with the Cisco Context-Aware Mobility Solution provide a single unified view of contextual information through the Mobility Services Engine API. Cisco Context-Aware Location Services enable both queries for contextual information as well as server registration for asynchronous events occurring among both wired and wireless access controllers.

In some cases, it may be necessary to track an asset with a high degree of accuracy throughout an enterprise, such as when a valuable missing asset must be located. On the other hand, some applications using Context-Aware Services may only require general indication of whether an asset is in or out of a permissible zone (such as the confines of a shipping and receiving dock, for example).

Context-Aware Services can also provide location and other contextual information for wired devices attached to Cisco Catalyst LAN switches, such as the 3560, 3750, 4500, and 4900 series. Catalyst switches can provide civic location details for wired devices to the Cisco Mobility Services Engine based on pre-configured information specified for each switch port. This information can then be presented to users in a tabular format combined with other contextual information, such as user name, device serial number, and emergency location identifier numbers (ELINs).

Note A civic location specifies the civic address and postal information for a physical location using fields such as the number, street or road name, community, and county assigned to residential, commercial, institutional, and industrial buildings (e.g., 31 Main Street, Alpharetta, Georgia 30004). An emergency location identifier number (ELIN) is a number that can be used by the local public safety answering point (PSAP) to look up the geographic location of the caller in a master database known as the automatic location information (ALI) database. The ELIN also allows the PSAP to contact the emergency caller directly in the event the phone call is disconnected.

Additional information regarding civic address location is available from the IETF in the following RFCs:

- DHCP Option for Civic Addresses Configuration Information: http://www.rfc-editor.org/rfc/rfc4776.txt
- Revised Civic Location Format for Presence Information Data Format Location Object: http://www.rfc-editor.org/rfc/rfc5139.txt

For a detailed explanation of Context-Aware Services, see: http://www.cisco.com/en/US/netsol/ns788/index.html

Why Should I Use Context-Aware Services?

Cisco Context-Aware Services provide the capability to determine physical location in the network, as well as additional contextual information such as the MAC address, IP address, serial number, or 802.1x username associated with the tracked entity. For example, at the time this document was published, tracked entities include wired switches, wired or wireless endpoints such as phones, physical computers, virtual machines (VM), Digital Media Players (DMP), and IP video cameras. This is by no means intended to be an exhaustive list, but rather is intended to illustrate the flexibility of Cisco Context-Aware Services in tracking a variety of different types of wired physical and virtual devices that are capable of connecting to Ethernet networks.

In order to provide delivery of a best-in-class network experience to end users, a Unified Access network solution should be aware of the dynamics surrounding users and endpoint devices, such as their current location. A wide variety of devices can appear on the network, both wired (switches, routers, IP phones, PCs, access points, controllers, video digital media players, and so on) and wireless (mobile devices, wireless tags, rogues, and so on). Often, locating missing assets in modern-day enterprises becomes a time consuming, error-prone, and costly manual process. In these cases, it has all too often become

simply more cost effective to replace the asset rather than expend employee productivity attempting to locate it. However, shrinkage costs and productivity losses will mount over time and can take their toll on the balance sheet. The inability to quickly and efficiently locate missing assets and ensure their availability when and where they are needed in the business cycle can severely throttle the productivity of even the best-managed organizations.

Cisco Context-Aware Services have been a part of Cisco wireless networks for some time and with recent enhancements regarding wired location capabilities, they are poised to play an increasingly important role in Unified Access networks. For example, future Context-Aware Services will enable location to be considered as a security policy attribute. Thus, a phone in the lobby of an office building can be assigned different policies than a phone in a conference room or in an employee's office. To date, authorization policies have been statically administered for endpoint devices based on characteristics such the endpoint's MAC address or the user's personal access credentials. The location of the endpoint has traditionally not been considered in the determination of which policies might be most appropriate depending on location. Context-Aware Services enable security polices to evolve such that policy assignment will be based not only on the user's credentials and MAC address, but will include location and other contextual information. This culminates in the ability to offer security policies offering greater differentiation that what has been traditionally possible.

While integration of Context-Aware Services with security authorization and authentication network services has the potential to revolutionize how we grant access to network users, the context-aware story does not stop there. The information about wired and wireless endpoints contained within the context-aware databases can be used to bring contextual awareness to a host of other applications, including Unified Communications and Unified Presence. Thus, integration between the pillars of the Cisco Borderless Networks Architecture will make use of shared contextual information to change how voice and instant messaging users, for example, might detect user presence on the network. Instead of simply knowing if a user is present and available, the user's presence status could changed based on contextual information such as their current location, whether they are stationary or in motion, or other characteristics.

In some cases, endpoints themselves may already be aware of their location. Context-Aware Services must be capable of making use of such endpoint intelligence. Examples of endpoints that can dynamically determine their own location would include those equipped with onboard Global Positioning System (GPS) capabilities. In other cases, endpoints derive their location from the network (either wirelessly or through a wired medium). In these cases, Context-Aware Services can play a key role in learning or supplying these endpoints with their current location via well known protocols such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), and Link Layer Discovery Protocol for Media Endpoint Devices (LLDP-MED). And in other cases, endpoints may already be statically configured with their location information beforehand or receive such static location configuration from application servers. Dynamic location update provides better support of mobile endpoints, whether they are wired or wireless.

For example, if a wired phone is provisioned for Seat 1A/02, Room 100, Floor 1, Sorenson Building, and if the person whose phone this is moves his cubicle to Seat 3F/16, Room 306, Floor 3, Paine Building, the location information associated with this phone would need to be manually updated if there is no means by which it can be updated dynamically. However, if the endpoint is capable of "learning" its location from its point of attachment in the enterprise network, this location can be used by the endpoint as well as other network applications with which the endpoint interacts. The contextual information associated with the device can be recorded by a location server and be made available via an API to administrative and other applications wishing to track the whereabouts of the device, even if the device itself is not able to interact directly at the current time.

While the information that resides within the context-aware databases of the Mobility Services Engine is, in and of itself, highly valuable, the true power of Context-Aware Services blossoms when it is put to work in a true cross-pillar integration scenario and combined with the power of Cisco Unified Access solutions and Cisco Network Services. This results in enhanced capabilities that have the potential to shatter the boundaries of what we have come to accept as traditional enterprise computing networks.

Use Cases For Context-Aware Services and Wired Endpoints

Context-Aware Services can help businesses answer important questions concerning enterprise assets as well as the users of those assets. In this way, Context-Aware Services can help contribute towards improving an organization's efficiency, increasing productivity, and helping to lower costs.

There are several important use cases where Context-Aware Services can yield real-world business benefits in Unified Access networks today, including:

- Asset Recovery—Context-Aware Services provide a central portal to the location of wired endpoint devices on the network. In the wired environment, context-aware location is capable of tracking not only the MAC address, assigned IP address, and 802.1x user name of attached wired devices, but the serial number or Unique Device Identifier (UDI)[1] information of endpoints compatible with CDP or LLDP-MED. This information can be used individually or in combination to determine not only the switch port to which the wired endpoint is attached, but also the civic location or ELIN information that is associated with this switch port. Having all of this information at the administrator's disposal can significantly accelerate recovery of missing assets. For example:
  - Recover misplaced assets that have been temporarily relocated within the enterprise legitimately, but never returned to their original location after such temporary use was completed.
  - Recover stolen assets or assets that have been relocated without proper authorization (perhaps between departments or divisions). This might include:
    - Assets purchased by an enterprise and "borrowed" by another business unit. While these assets are likely still on company property and in legitimate use, the original business unit in all likelihood wishes to have the asset returned and not have to incur the cost of asset replacement.
    - Assets that have been the target of theft or otherwise converted to unauthorized use outside the scope of the enterprise business mission. A good example might be a laptop that was removed by a student resident employed part-time by a university. The student removes the laptop from a laboratory and converts it to personal use in his or her dormitory, attaching it to the high-speed university wired network. Unknown to the student, the entire university campus (including all dormitories) makes use of Context-Aware Services as part of its deployment of a Cisco Validated Design (CVD), Unified Access Network Design. Thus, whenever the student uses the laptop on the network, network administrators and university loss prevention staff have the ability to identify the laptop as well note its location using the Cisco WCS and the Mobility Services Engine. Additionally, the Mobility Services Engine via its API allows for third-party applications to access historical asset location and identify a pattern of movement. This makes it possible to establish a record of illegal asset movement and determine the time and date assets were last seen on the network and where they were located at that time. Using the API, third-party applications can use the current

1. The Unique Device Identifier (UDI) is the Cisco Systems product identification standard for hardware products. More information about UDI can be found at: http://www.cisco.com/en/US/products/products_identification_standard.html.

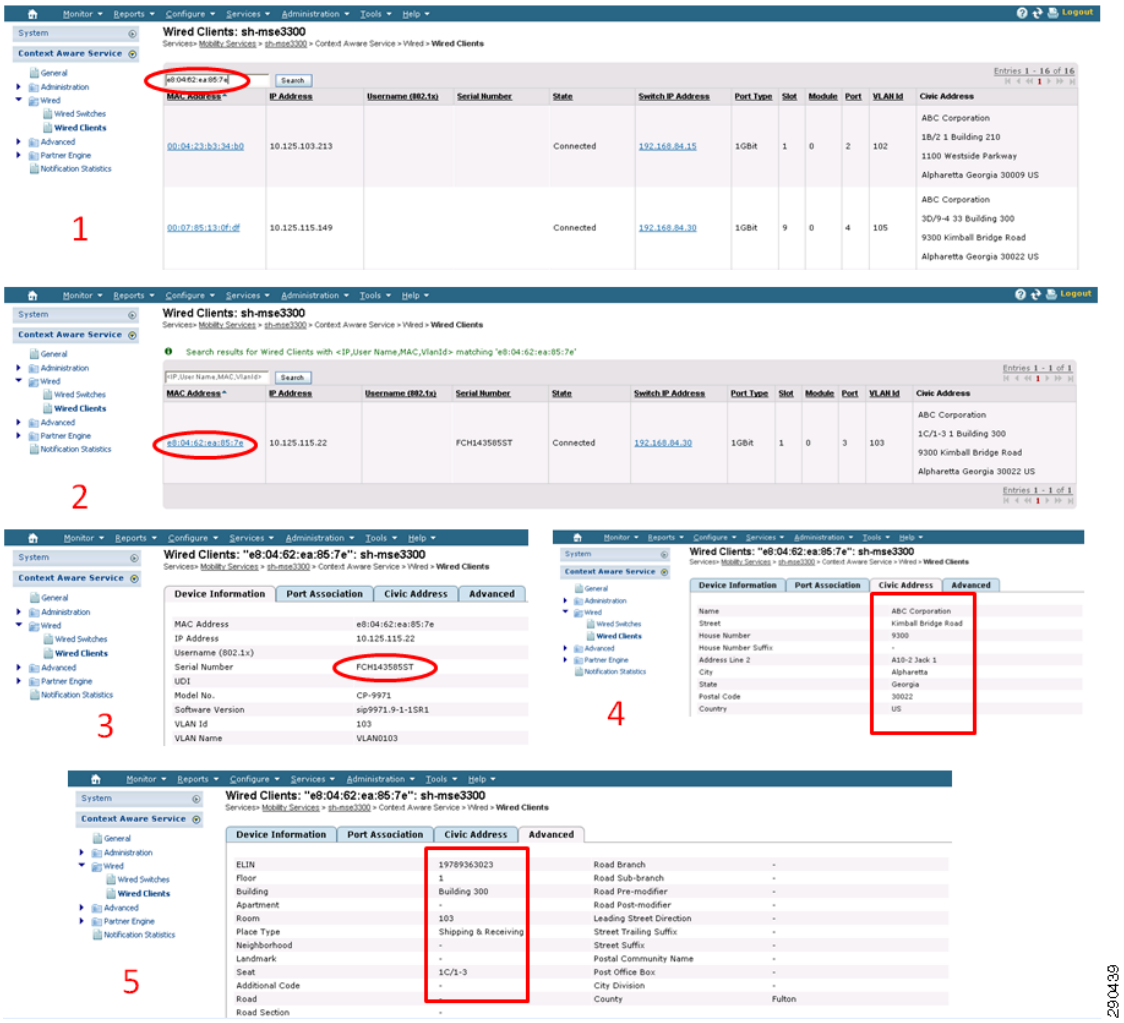
and historical location databases to issue alerts and notifications and apprise loss prevention and other authorities as to the appearance of a “wanted” asset on the ports of any monitored Catalyst switch.

Figure 4-1 is a visual depiction of how Context-Aware Services for wired endpoints could be used to support asset recovery efforts within an enterprise. This example assumes we are attempting to locate a missing Cisco 9971 IP phone with the MAC address E8:04:62:EA:75:7E. This device was originally installed on port 1/0/4 in a Catalyst switch with the IP address 192.168.84.11. However, it is no longer there. If this 9971 phone is present on the network, we would like to know its physical location so that it can be retrieved and returned to its original location and owner. Using Context-Aware Services, identifying and locating this wired IP phone is a straightforward procedure. First, we access the list of all wired clients of which the Mobility Services Engine is currently aware. We then search on the MAC address of the IP phone. We can see in image #2 that the device is currently connected on switch port 1/0/3 of the switch with IP address 192.168.84.30. Clicking on the device MAC address in image #2, we are presented with an array of information about the device and its current location. We can verify the serial number of the phone in image #3, allowing us to increase our confidence that this is indeed our missing asset and reassure ourselves that we are not the victim of MAC address spoofing. Image #4 and #5 in Figure 4-1 provide us with the information we seek, namely, that port 1/0/3 on this switch terminates at cubicle/seat identifier 1C/1-3 in room 103 on the first floor of building 300 of ABC Corporation. This building appears to be used by ABC Corporation for shipping and receiving. In case we are unfamiliar with where building 300 is, we can see that it is located at 9300 Kimball Bridge Road, in the City of Alpharetta, Fulton County, Georgia, postal zip code 30022.

In summary, with the information provided by Context-Aware Services in Figure 4-1, we can:

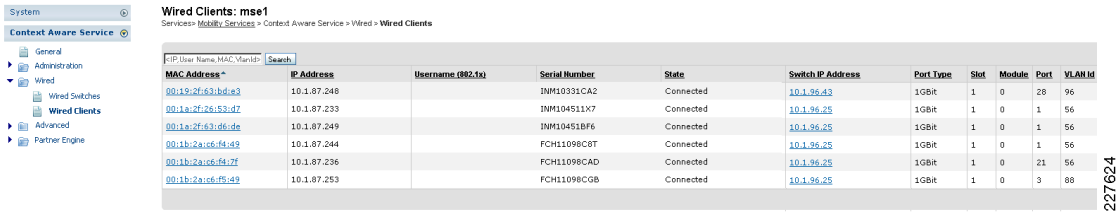
1. Search for the missing asset by MAC address.
2. Determine if the asset is currently connected to the network and, if so, identify its connected switch and switch port.
3. Optionally verify this is indeed the asset we seek by comparing its reporting serial number against any serial number for the asset that we may have in our records from its time of purchase or original deployment.
4. Determine the physical location where the switch port in #2 above is terminated. If the information in the switch is provisioned accurately and correctly, the location of the device can be determined right down to the floor, room and cubicle, office, or seat location. This is often sufficient to expedite efficient retrieval of the asset.

Figure 4-1 Example of Wired Asset Identification and Location



- Asset Inventory—Context-Aware Services make use of CDP and LLDP-MED protocols to acquire serial number information from compatible wired endpoint devices. Any available wired endpoint serial number information acquired is displayed alongside location information for all wired endpoints that the system is currently tracking. Consequently, this capability can be used to conduct impromptu inventories of wired devices across the enterprise by examining wired device listings by serial number and comparing this information to actual deployment records. This can help enterprise IT administrators and others responsible for asset deployment to better determine whether assets have been moved between offices, buildings, or campuses improperly or without authorization. Figure 4-2 illustrates an example of how this capability could be used to provide a quick inventory of all Cisco IP phones that are attached to one or more Catalyst switches participating in Context-Aware Services. Clicking on the MAC address hyperlink for any of these phones would provide the administrator not only with the civic location of the phone, but also the assigned ELIN information that has been defined in the switch.

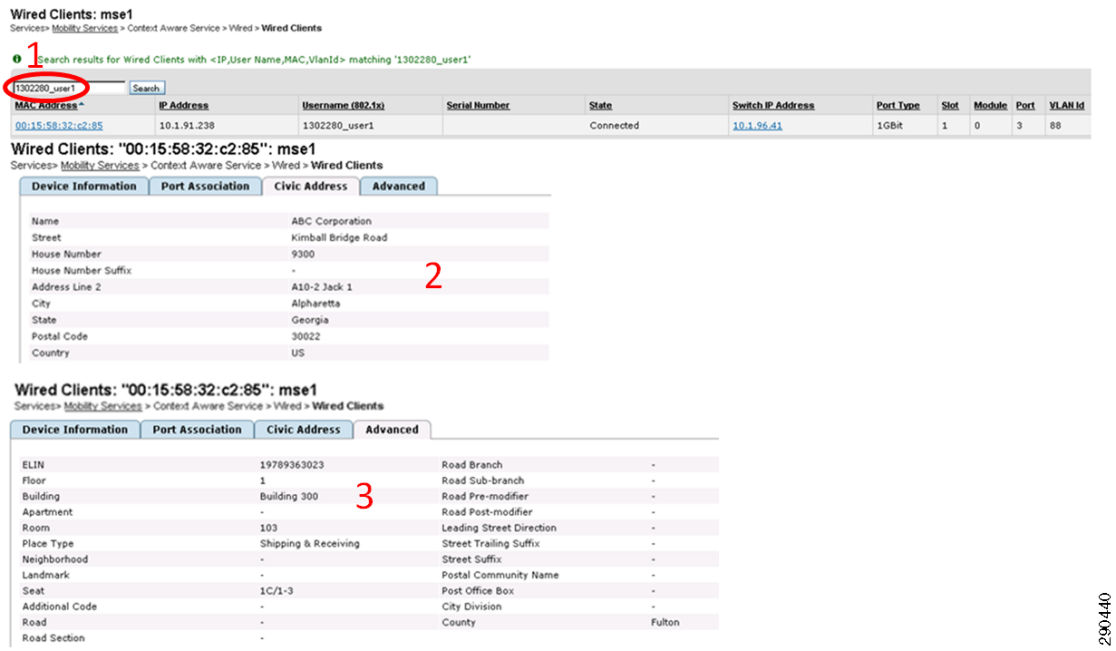
Figure 4-2 Example of Quick Inventory of IP Phones



- Location by User Name—In addition to locating the endpoint device that a user may be utilizing to access the network by its MAC or IP address, Context-Aware Services for wired endpoints also provide us with the ability to locate a wired user by the username entered during 802.1x authentication. This information is not only shown during a general listing of wired clients, but it is also a search parameter. While there are other sources that can locate the switch and switch port assigned to an 802.1x user in the Unified Access network, Context-Aware Services provide the administrator with the ability to quickly and efficiently determine not only what switch and port the user has been assigned, but the location in which the switch port terminates. Thus, the administrator is provided with a "one-stop shopping" source that can translate between a person's 802.1x username and their location without any intermediate translation.

  Figure 4-3 illustrates the process of searching on 802.1x username and how easily the search results are translated into an actual location. In image #1 we search for the username "1302280_user1" across all of the wired clients of which the Context-Aware Services for this Unified Access network is currently aware. User 1302280_user1 is located in image #2 and we can see the civic address information as well as any information pertaining specifically to how the Ethernet jack may be labeled. Image #3 allows us to retrieve additional information about where this user is located, including the building number, the ELIN, and the room and seat number assigned.

Figure 4-3 Using for a Wired Client by 802.1x Username



- Location of Host Servers:

  - Location of Physical Servers—In addition to wired end user devices such as laptops and desk top computers, Context-Aware Services for wired endpoints can also be used to efficiently track the location of physical servers and even the virtual machines that operate on these physical servers. Locating a specific physical server among a variety of often identical servers in massive and often monolithic computing environments can be a daunting task. Using Context-Aware Services and at least one known characteristic of the server (MAC address, IP address, 802.1x username, or VLAN ID), the location of the server can be quickly and efficiently retrieved using WCS and the MSE. The MSE can provide a very precise location for the server, right down to a civic address consisting of the building, floor, room, and even rack and occupied slot number where the server is housed. This is a tremendous benefit in large computing environments, where hours of employee productivity could be wasted attempting to locate server hardware whose placement records have not been kept up to date. A good example of how this can be applied is in ensuring that decommissioned physical hardware is accurately located, cleansed of all sensitive enterprise information, and transferred to its appropriate final destination. Context-Aware Services can help ensure that decommissioned hardware is not somehow returned to service without proper authorization. Such “cradle-to-grave” tracking in the enterprise is especially useful in environments where physical servers may be leased and must be decommissioned and returned to leasing companies at the conclusion of their lease and not misappropriated for other uses within the enterprise. This is an all too common occurrence that can result in unnecessary leasing penalties and added expense.
  - Location of Virtual Machines—Locating a mislabeled physical server in an environment when there might be hundreds of other identical servers may seem like the quintessential “needle in the haystack” experience. Imagine then the challenge of searching for a specific virtual machine within this huge inventory of physical hardware. This might be the case if a virtual machine is relocated to a new physical host, with the same virtual IP address and MAC address that it has used elsewhere, but with no record of its movement. Context-Aware Services can prove useful to the harried administrator faced with resolving this situation quickly and efficiently. For example, a misplaced virtual machine operating on a VMware ESX physical host can be located using WCS and the MSE by searching on characteristics uniquely associated with the virtual machine, such as its virtual IP address or MAC address. If the virtual machine is moved from one physical host to another (which implies movement from one Catalyst switch port to another), Context-Aware Services will track its movement and its new location as if it were just another wired device. The virtual machine will appear in its new location using the civic location information associated with its current physical host. Because of its ability to search across all of the Catalyst switches attached to an entire server farm, Context-Aware Services can be used to locate a virtual machine, right down to a civic address consisting of the building, floor, room, and even rack and slot number of where the current parent physical server to the virtual machine is housed. No matter which connected physical host the virtual machine has been relocated to, Context-Aware Services for wired devices can contribute significantly to saving employee time and helping increase productivity.

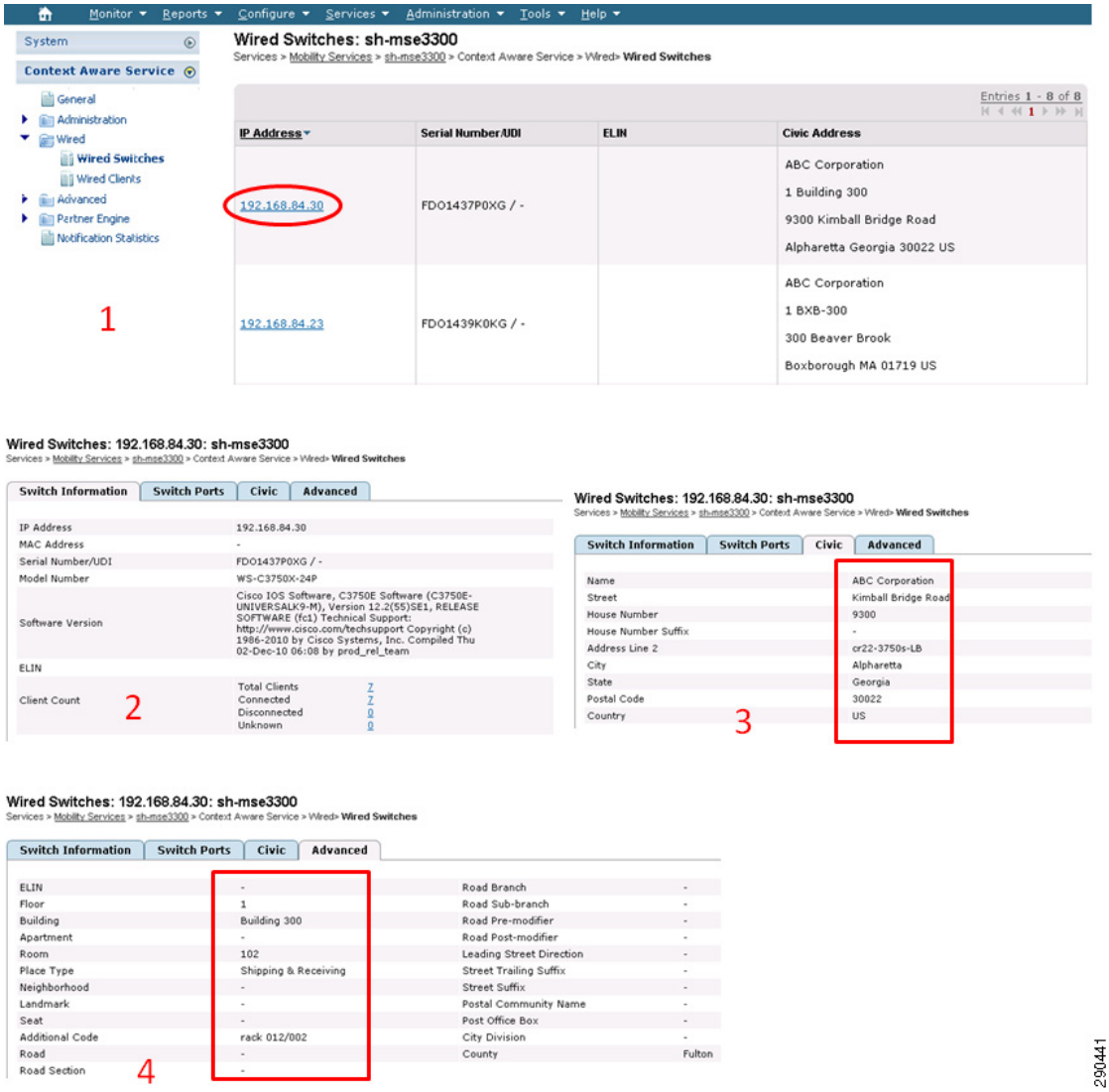
- Catalyst Switch Chassis Location—In addition to quickly locating end user devices, physical servers, and even virtual machines, Context-Aware Services can also be used to quickly query the location of a deployed Catalyst switch chassis itself. By assigning civic location characteristics regarding its deployed location to the out-of-band management port of a Catalyst switch, a network administrator can use WCS and the MSE to query the location of the switch itself. If defined accurately and maintained judiciously, this capability may reduce the need to maintain separate lists of deployed switch locations in many cases.

  Figure 4-4 provides a visual example of how context-aware wired endpoint location can be used to locate a Catalyst switch chassis in the Unified Access network. Imagine that a rookie network technician or administrator, unfamiliar with a large enterprise network, needs to quickly determine the location of a context-aware Catalyst 3750X switch with IP address 192.168.84.30. They can use the wired endpoint location capabilities of Cisco Context-Aware Services by simply finding this

switch in the list of wired switches in WCS and clicking on the civic location tab associated with it. As long as the civic location information assigned to the management port interface is kept accurate and up to date, all of the information to identify the rack slot, rack number, room, floor, and building associated with the 3750X switch can be obtained from the MSE via WCS. There is no need for the administrator or technician to query another database for the deployed location of the 3750X switch. Using the sequence of events shown in Figure 4-4, we see that it is simply a matter of a few mouse clicks in WCS to determine that the switch in question is located in rack 12, slot 2, in room 102 on the first floor of building 300. This building is operated by ABC Corporation for the purposes of Shipping and Receiving. It is located at 9300 Kimball Bridge Road, in the city of Alpharetta, Fulton County, Georgia, postal zip code 30022.

Context-Aware Services help to integrate under one roof not only the location information for deployed wired endpoints, but for the switch chassis itself, thereby placing all of it at the fingertips of the WCS user in an efficient manner. This reduces response times, increases productivity, and can help reduce the overall costs associated with maintaining the enterprise network.

Figure 4-4 Locating a Catalyst Switch Chassis



Cisco Context-Aware Components

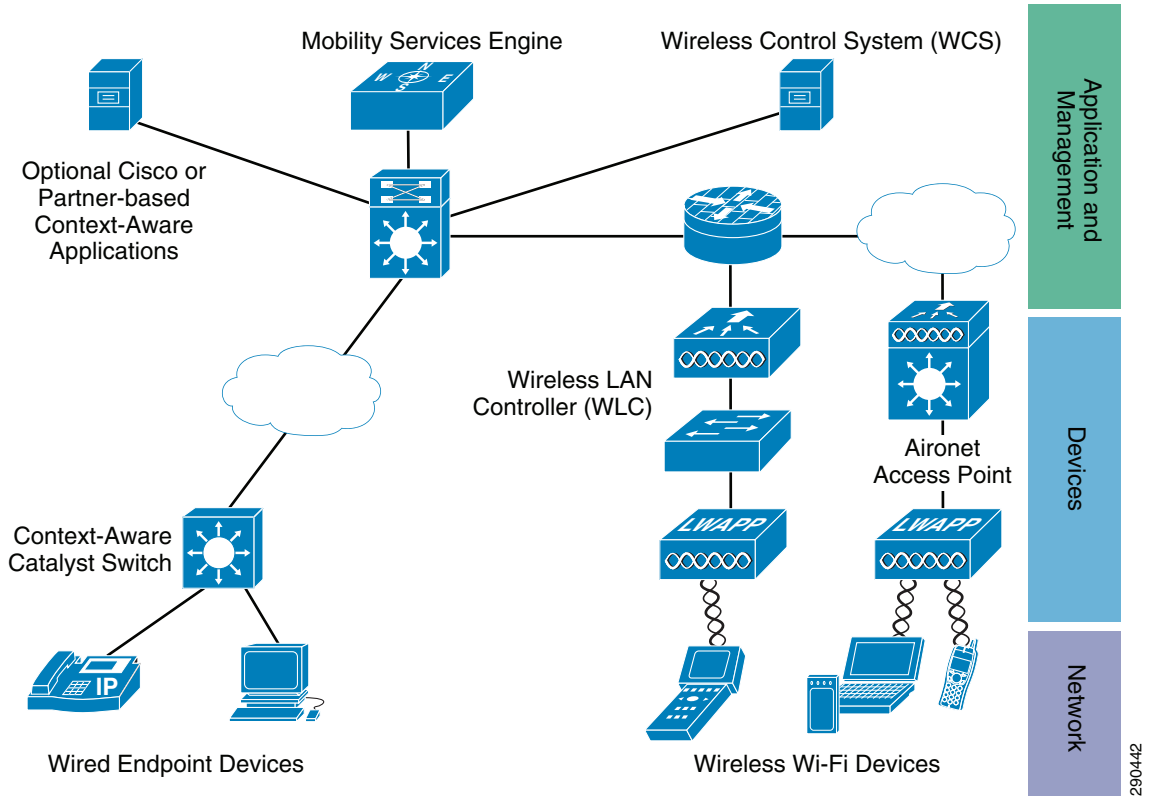
The components used to implement Cisco Context-Aware Services are shown in Figure 4-5. Context-Aware Services are designed to record contextual information (including location) as it pertains to wired and wireless devices in an enterprise and maintain a repository of those locations in a central server. The overall solution has four fundamental components:

- Cisco 3300 Series Mobility Services Engine (MSE)
- Cisco Wireless Control System (WCS)
- Cisco Catalyst context-aware Ethernet switches for wired client access
- Cisco Wireless LAN Controllers for wireless client access

Included in Figure 4-5 are optional context-aware applications. These might be value-added applications supplied by Cisco Technology Development Partners, such as RedSky and their E911 WiFi solution suite (http://www.redskye911.com/e911_products/e911_manager/wifi_e911/cisco/). Or these might be interfaces to other Cisco network services that represent other pillars of the Borderless Network Architecture, such as integrated security services or Unified Communications and Unified Presence applications.

It should be noted at this point that while Figure 4-5 includes wireless LAN components, the majority of this chapter is focused on the use of Context-Aware Services in the Unified Access design as it pertains to wired endpoints.

Figure 4-5 High Level View of Cisco Context-Aware Services



Wired Endpoint Devices

These are wired Ethernet devices that interact with the network and whose location and other contextual parameters can be monitored by Context-Aware Services. Wired devices are equipped with an Ethernet interface and are attached to a Cisco Catalyst Ethernet switch (such as the 3560, 3750, 4500, and 4900 series) that is participating in Context-Aware Services. Via the Mobility Services Engine's API, civic and ELIN location information received from context-aware Catalyst switches can be provided to WCS, other Cisco Borderless Network services components, or Cisco partner-developed context-aware applications.

Cisco Unified Access Network

This multipurpose network contains the infrastructure required to address converged data, voice, and video requirements, as well as providing the foundation for Context-Aware Services. While the network may be composed of various routers, switches, firewalls, and so on, the main wired infrastructure network component actively participating in the passage of information back to the MSE is the context-aware Catalyst Ethernet switch. These switches support the specification of civic address and emergency location identification number (ELIN) information for each switch port.

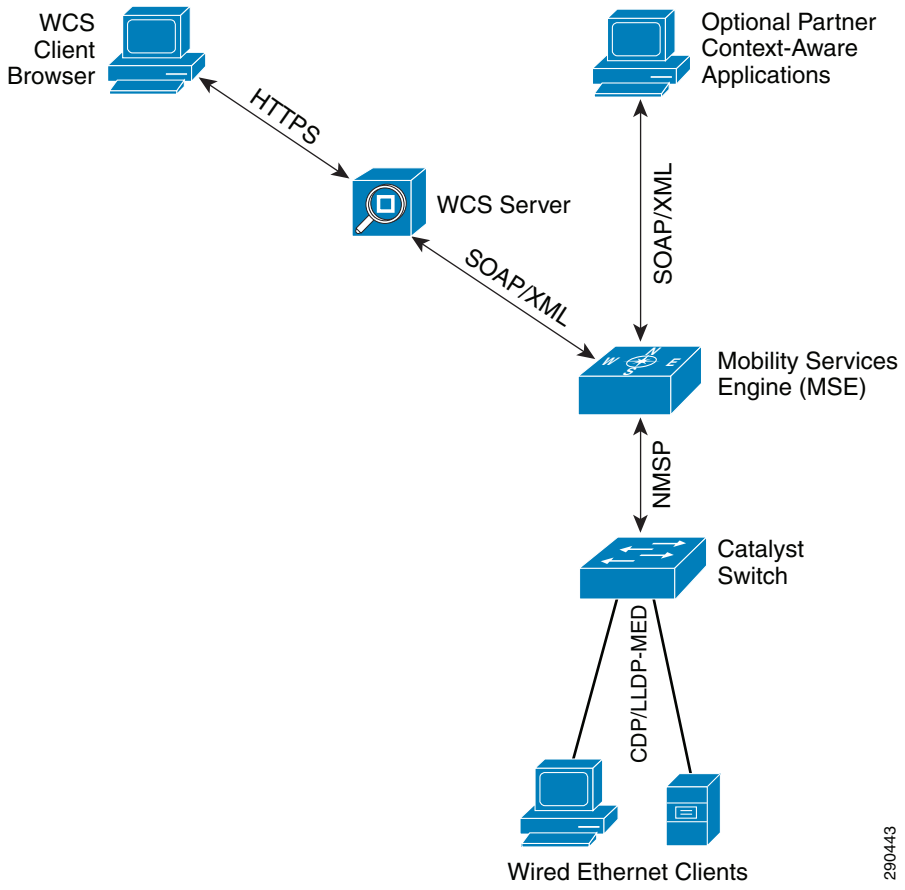
Context-aware switches pass information for all attached devices to the Mobility Services Engine via communications sessions established between the two partners. This information may include the physical mailing or street address location associated with the attached device (the civic address) as well as other information such as the IP address, MAC address, port, VLAN, and 802.1x user name. Typically, this information is obtained using switch features such as IEEE 802.1x, Dynamic Host Configuration Protocol (DHCP) snooping, Dynamic Address Resolution Protocol (ARP) Inspection (DAI), and IP Source Guard. Additionally, if the end device supports CDP or LLDP-MED protocols, additional information, such as the version and serial number, can also be sent to the MSE.

Refer to the following data sheets for more information about the Cisco Catalyst switches and their ability to participate in Cisco Context-Aware Service as part of the Cisco Borderless Experience: http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps6406/data_sheet_c78-584733_ps10744_Products_Data_Sheet.html

Management and Application Interaction

Figure 4-6 provides an illustration of the protocol interaction between the aforementioned Context-Aware Service components used with wired endpoints.

Figure 4-6 Context-Aware Component Interaction



- Cisco Mobility Services Engine (MSE)—The Cisco 3300 Mobility Services Engine can execute multiple independent services in support of Unified Access network infrastructures. Those services typically provide high-level capabilities, such as Cisco Context-Aware Services or the Wireless Intrusion Protection Service (wIPS). The Mobility Services Engine records contextual information for both wired and wireless assets on a continuous basis. It can do this by having the wired and wireless network infrastructure devices (WLAN controllers and context-aware Catalyst switches) send updated contextual information for attached assets to the MSE via NMSP as changes occur.

  The Unified Access network communicates with the MSE using the Cisco Network Management Services Protocol (NMSP), which is a Cisco-defined protocol used for secure communication between the MSE and other context-aware network infrastructure components. The switch sends location and attachment tracking information for its connected devices to the MSE. The switch notifies the MSE of device link up and link down events via NMSP location and attachment notifications.

  The MSE starts the NMSP connection to the switch, which opens a server port. When the MSE connects to the switch, there are a set of message exchanges to establish version compatibility and service exchange information followed by location information synchronization. After connection, the switch periodically sends location and attachment notifications to the MSE upon the expiration of an internal timer. Any link-up or link-down events detected during an interval are aggregated and sent at the end of the interval.

When the switch determines the presence or absence of a device on a link-up or link-down event, it obtains the client-specific information, such as the MAC address, IP address, and 802.1x username. If the client is LLDP-MED or CDP capable, the switch obtains the serial number or UDI via these protocols.

Depending on the device capabilities, the switch obtains the following client information at link up:

- Slot and port on which connection was detected
- Detected client MAC address
- Detected IP address
- 802.1X username (if applicable)
- Device category (wired station)
- Device state (add new)
- Serial number or UDI
- Model number
- Time in seconds since the switch first detected the association

Depending on the device capabilities, the switch obtains this client information at link down:

- Slot and port on which connection was disconnected
- Detected client MAC address
- Detected client IP address
- 802.1X username (if applicable)
- Device category (wired station)
- Device State (delete)
- Serial number or UDI
- Time in seconds since the switch first detected the disassociation

When the switch shuts down, it sends an attachment notification with the state delete and the IP address before closing the NMSP connection to the MSE. The MSE interprets this notification as disassociation for all the wired clients associated with the switch.

When location information is changed on the switch, the switch sends an NMSP location notification message that identifies the affected ports and the changed address information.

After the MSE records the current location of a wired endpoint device, it provides this information to WCS, external Cisco-partner supplied context-aware applications, or other Cisco network components. The two primary approaches used for information retrieval from the MSE via the API are:

- Queries—The external system typically sends a query asking for a device's location, including optional query criteria such as to return data for only a specific MAC address. The MSE would respond with the answer immediately upon receipt of the query.
- Subscribe and publish—The external system typically registers an event subscription for device location based on a set of criteria such as changes in location or changes in location beyond a prescribed area.

Context-Aware Services software is capable of servicing up to a maximum of 18,000 simultaneously tracked devices per MSE-3350 or MSE-3355 appliance and 2,000 simultaneously tracked devices per MSE-3310 appliance. In Release 7.0, all civic and ELIN location information is configured

locally at the switch, and these changes are propagated to the MSE via NMSP. NMSP is used between the switches and the MSE to maintain synchronization and alert the switch as to the connection or disconnection of devices.

Refer to the following data sheet for more information regarding the Cisco 3300 Series Mobility Services Engines:
http://www.cisco.com/en/US/prod/collateral/wireless/ps9733/ps9742/data_sheet_c78-475378.html

- Wireless Control System (WCS)—The Cisco Wireless Control System is a management platform that contains a context-aware client application interacting directly with the Mobility Services Engine. In this role the WCS provides access to the contextual information contained on the MSE using the MSE's context-aware application programming interface (API). WCS presents this information to the user in either a graphical or tabular format. The Cisco WCS also serves as a control client to be used to configure operational parameters on the MSE.

  For more detailed information on the Cisco Wireless Control System management server and its capabilities, including its ability to serve as a context-aware application client to manage the MSE, refer to the documentation at:
  http://www.cisco.com/en/US/prod/collateral/wireless/ps5755/ps6301/ps6305/product_data_sheet0900aecd802570d0.html.

- Other Context-Aware Applications—Although WCS provides an extremely powerful interface to the contextual information stored on the MSE, access to this information is not limited to only WCS. In fact, once contextual information has been captured, calculated, and stored in its resident databases by the Context-Aware software module on the MSE, it can be made available not only to WCS but to any business application that requiring access via the API, which is based on the Simple Object Access Protocol (SOAP) and XML protocol.

  Cisco-developed applications have access to this API of course, but so do context-aware applications developed by Cisco Technology Development partners. Access to this API is available to any Cisco technology partner and allows full integration into enterprise business processes. When planning for what can be accomplished in your organization with the information contained within the context-aware service on the MSE, it is important to remember that context-aware applications developed by Cisco Technology Development partners often target specific industry needs, delivering value, functionality, and enhanced capabilities that are often not available from any other source.

  For more information on the Cisco Context-Aware Services API, refer to:
  http://developer.cisco.com/web/contextaware/home.

Network Mobility Services Protocol (NMSP)

The Network Management Service Protocol (NMSP) was designed to define intercommunication between Mobility Service Engines and network access controllers over a switched or routed IP network. An access controller can provide network access for either wired or wireless endpoints. Within the scope of this Unified Access chapter, access controllers are represented by context-aware Cisco Catalyst Ethernet switches.

NMSP is a two-way protocol that can be run over a connection-oriented or a connectionless transport. Context-aware switches can use NMSP to communicate with one or more MSEs. NMSP is based upon a bidirectional system of requests and responses between the MSE and access controllers.

Note It is important to understand that the failure of an NMSP session has no direct impact on the ability of a Catalyst switch to pass normal client voice, data, and video session traffic to applications on the network. In other words, a failed NMSP session to a Catalyst switch may affect the ability of the MSE to provide updated contextual information for that switch and its resources, but it does not affect the ability of the attached devices to log on to applications residing on the network.

NMSP uses Transport Layer Security (TLS) and TCP port 16113 on the Catalyst switch. The MSE will initiate the connection to the Catalyst switch, after which messages may be transmitted in either direction. The TCP port (16113) that the Catalyst switch and MSE use for communication must be open on any firewall that exists between the switch and the MSE.

NMSP provides for a keep-alive protocol mechanism that allows either partner in a NMSP session to determine if the adjacent partner is still active and responsive. Should an MSE fail, the Catalyst switch will try to contact another MSE with which to communicate. If the Catalyst switch fails, all Context-Aware Services being provided to that Catalyst switch are disabled until the switch once again becomes active and re-establishes its NMSP session.

The MSE and the Catalyst switch use Echo Request and Echo Response control messages to maintain an active channel of communication so that the data messages can be sent. The Echo Request message is a keep-alive mechanism that allows either NMSP session partner to determine if the other partner remains active and responsive. Echo Requests are sent periodically (upon expiration of a heartbeat timer) by the MSE or its session partner to determine the state of the NMSP session. When the Echo Request is sent, a NeighborDeadInterval timer is started. The NeighborDeadInterval timer specifies the minimum time a session partner must wait without having received Echo Responses to its Echo Requests before the other session partner can be considered non-responsive and the NMSP session is placed in an idle state.

Note Proper validation of certificates between context-aware service components requires the participants to possess sane clocks (clocks whose configured time does not differ from one another by large amounts). In order to facilitate this, it is highly recommended that the clocks in the MSE, WCS, and any Catalyst switches participating in Context-Aware Services be synchronized to a common time base using the Network Time Protocol (NTP). The lack of clock sanity among context-aware components in the network can cause NMSP sessions to fail if the configured date and time fall outside of the certificate's expiration.

Context-Aware Services in Unified Access

Figure 4-7 illustrates the test bed used to validate the use of Cisco Context-Aware Services with wired endpoints in the Unified Access design.

Figure 4-7 Test Bed Configuration

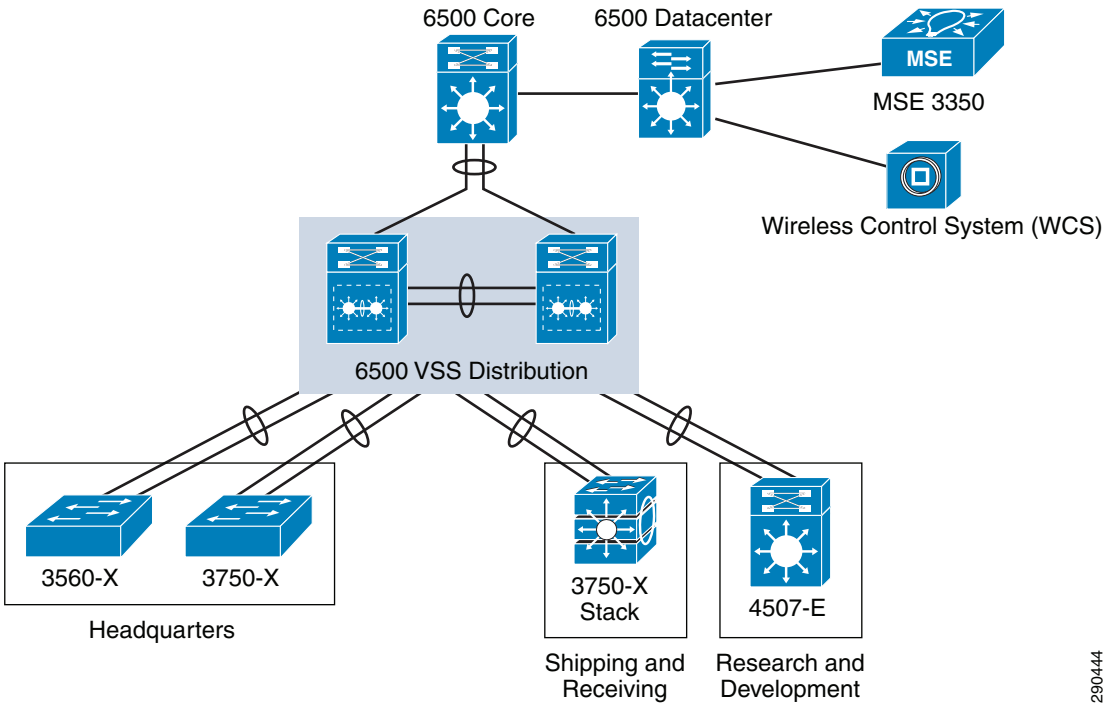


Table 4-1 Sample Civic Location Information for Headquarters 3750X

Identifier	bldg_210
County	Fulton
Street Group	Westside Parkway
Street number	1100
Name	ABC Corporation
Building	Building 210
Type of place	Headquarters
City	Alpharetta
State	Georgia
Postal code	30009
Country	US

Table 4-2 Sample Civic Location Information for Shipping and Receiving 3750X Stack

Identifier	bldg_300
County	Fulton
Street Group	Kimball Bridge Road
Street number	9300
Name	ABC Corporation

Table 4-2 Sample Civic Location Information for Shipping and Receiving 3750X Stack

Building	Building 300
Type of place	Shipping & Receiving
City	Alpharetta
State	Georgia
Postal code	30022
Country	US

Table 4-3 Sample Civic Location Information for Research & Development 4507R

Identifier	1/1
County	Fulton
Street Group	Webb Bridge Road
Street number	3680
Name	ABC Corporation
Building	Building 400
Floor	1
Room	101
Type of place	Research & Development
Seat	1A/1
City	Alpharetta
State	Georgia
Postal code	30023
Country	US

The access layer was configured so as to simulate a enterprise named “ABC Corporation” with local access switches spread among departments situated at various campus locations. Thus, we see in Figure 4-7 that ABC Corporation possesses shipping and receiving, research and development, and headquarters locations, all of which have deployed context-aware Catalyst switches. These switches establish NMSP sessions with the MSE located at the central data center. A WCS has also been deployed at the data center and access to the WCS (and hence to the user GUI for Context-Aware Services) is available to authorized users throughout the enterprise.

The test network is based on a three-tier routed access campus design, with Layer 2 and Layer 3 switches at the access layer and Layer 3 switches in the distribution and core layers. As seen in Figure 4-7, the Layer 2 access was provided in this design by a Catalyst 3750X switch. The Layer 3 switches used at the access layer were:

- Catalyst 3560X
- Catalyst 3750X, as a nine member switch stack
- Catalyst 4507
  - Chassis Type: WS-C4507R-E
  - Sup 6-E 10GE WS-X45-SUP6-E

  - 10/100/1000BaseT (RJ45)V WS-X4548-GB-RJ45V used for client connections
  - SFP, 10/100/1000BaseT (RJ45)V WS-X4506-GB-T used for uplink to distribution

All access switches were loaded with k9 IOS cryptographic images at a code train level supporting NMSP and civic address port location interface subcommands.

The distribution and core layers were composed of Catalyst WS-C6509-E switches with Sup 720 10GE. The distribution layer was configured to operate in VSS mode.

In the test bed, OSPF was used as the routing protocol. The campus core was configured to represent OSPF Area 0 and the access layer was configured to use stub areas.

In accordance with general industry network design best practices, out-of-band (OOB) network management was used where feasible. For example, the FastEthernet interfaces specifically designated for OOB network management on the Catalyst 3560 and 3750 switches were utilized for all NMSP and SNMP traffic to and from the switch. On the other hand, the management interface on the Catalyst 4507 was not used in this design. This interface is by default a VRF interface and does not currently support NMSP at this time. Therefore, all inbound and outbound NMSP to this switch was conducted in-band.

The test bed configuration was based foundationally upon the concepts established in the *Borderless Campus Design Guide 1.0*, which can be found at: http://www.cisco.com/en/US/partner/docs/solutions/Enterprise/Campus/Borderless_Campus_Network_1.0/Borderless_Campus_1.0_Design_Guide.html

A variety of wired endpoints were included in our testing:

- Windows XP PCs
- Windows 7 PCs
- Red Hat Enterprise Linux AS release 4
- FreeBSD 6.1
- Cisco 9971 IP phones
- Cisco Digital Media Player DMP-4310
- CIVS-IPC-2500 Video Surveillance Camera
- CIVS-IPC-4500 Video Surveillance Camera
- Cisco AP-3502 Access Point (AP3G1-K9W8-M)
- Cisco AP-1142N Access Point (C1140-K9W8-M)

These endpoints were moved between switch ports repeatedly, both on a intra-switch chassis and inter-switch chassis basis. Acceptable behavior of endpoints moving between switch ports was based on the following:

- For access switch ports with spanning-tree-portfast configured, device attachment status and initial location information should be reflected on WCS within a time interval not exceeding the default nmsp-attachment-interval of 30 seconds plus a maximum 10 second processing delay window, for a total time equal to or less than 40 seconds. Optimal performance under these circumstances would assume a processing delay window of zero seconds, for a total time equal to or less than 30 seconds.
- For access switch ports without spanning-tree-portfast configured, device attachment and initial location information should be reflected on WCS within a time interval not exceeding the switch's maximum spanning tree port transition delay plus the default nmsp-attachment-interval of 30 seconds plus a 10 second processing delay window.
- When evaluating the ability of the system to reflect a disconnected client, the spanning tree transition time is not applicable. Therefore, client disconnect status should be reflected on WCS within 40 seconds or less.

The Cisco DMP-4310 was used to verify that our test bed configuration could pass civic location information contained in CDP and LLDP-MED TLVs (type, length, and value attributes) to a compatible Cisco medianet endpoint device. This is an area of great development and excitement and is leading the way to allow for location-assisted auto-configuration of medianet devices in the future. More information on the DMP-4310 and its ability to receive civic location information from Cisco Catalyst switches may be found in the *Medianet Reference Guide*: http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/Medianet_Ref_Gd/chap7.html#wp1216490.

Two VMWare ESXi 4.0 Servers were used to allow for testing of virtual machines being moved from one physical host to another. The virtual machines tested included:

- Windows XP
- Ubuntu 10.4 Server
- CentOS 5.4

Component Capacities

Mobility Services Engine

Each Cisco Mobility Services Engine has a maximum combined device tracking capacity, which is a "hard" limit that is dictated by the licensing purchased for the Context-Aware software as well as the presence of any other applications on the MSE. Once a Mobility Services Engine has reached its maximum tracking capacity, any new devices the MSE becomes aware of that exceed that limit are simply not tracked. It is important to note that while this section discusses the maximum device tracking limits for the MSE, licenses can be purchased supporting device limits significantly lower than these maximums. Refer to the MSE Licensing and Ordering Guide (http://www.cisco.com/en/US/prod/collateral/wireless/ps9733/ps9742/data_sheet_c07-473865.html) for more information regarding the various client tracking capacities available for the MSE.

For Release 7.0, the maximum device tracking capacities for the MSE when using only the Context-Aware Service are shown in Table 4-4.

Table 4-4 Maximum Device Tracking Capacities

Mobility Service Engine	Maximum Tracked Device Capacity
MSE-3350/3355	18,000
MSE-3310	2,000

If you intend to use the MSE to deliver other services in addition to Context-Aware, the maximum capacities shown above will be reduced. See the *MSE Licensing and Ordering Guide* (http://www.cisco.com/en/US/prod/collateral/wireless/ps9733/ps9742/data_sheet_c07-473865.html) for information on Context-Aware maximum tracked device capacities when used with other co-resident services.

When working within these maximum capacities, it is important to note that further category-specific limits can be instituted via the MSE configuration. This allows, for example, a maximum capacity of 2,000 tracked devices on a MSE-3310 to be further limited as 400 wired endpoints, 600 wireless endpoints, 500 RFID tags, 250 rogue access points and rogue clients, and 250 interferers. Partitioning the maximum tracking capacity of the context-aware software in this manner prevents any single device category from consuming more than its allotted share of the maximum tracking capacity. This is a very

important consideration when designing context-aware solutions that incorporate awareness of rogue access points, rogue devices, or interferers, since the anticipated size of each of these device categories is not always predictable. Failure by the context-aware solution designer to institute reasonable limits on the total number of rogues and interferers tracked could result in premature exhaustion of MSE capacity, thereby preventing the tracking of other categories of devices (such as wired or wireless client endpoints).

The Context-Aware System Performance chapter of the *Mobility Services Engine Context Aware Deployment Guide* (http://www.cisco.com/en/US/products/ps9742/products_tech_note09186a00809d1529.shtml#casysperf) also points out that a single MSE can support up to 500 total NMSP connections. This has been validated to 100 total NMSP connections and includes not only NMSP sessions to Catalyst switches, but also to any WLAN controllers participating in Context-Aware Services with this MSE.

Wireless Control System (WCS)

As mentioned earlier, WCS interacts with the MSE as both a Context-Aware Services application client as well as a control client. When used to locate endpoints in the network via the MSE, WCS does not execute any location algorithms itself. All location calculations are handled by the MSE. There are no NMSP sessions that are established between WCS and Catalyst switches or WLAN controllers. Because of this client-server relationship between WCS and the MSE, there are no direct constraints on the maximum number of tracked devices that are imposed by WCS.

There are, however, a few indirect constraints that currently exist of which the network designer should be aware:

- A single WCS can manage and interact with multiple MSEs. Cisco officially supports a single WCS managing and interacting with up to five (5) MSEs. While defining more than five MSEs to a single WCS is possible, Cisco has not validated this configuration.
- A single MSE should be managed by only one WCS. In other words, there is a 1:1 mapping that should be maintained between an MSE and the number of WCS systems attempting to manage and interact with that MSE. Care should be taken not to confuse this consideration with the bullet above.
- It is important to note that in release 7.0, non-root WCS virtual domain users cannot access WCS functions listed under Services > Mobility Services. This includes wired switch and wired endpoint device location. Therefore, since wired devices attached to context-aware Ethernet switches are displayed using Services > Mobility Services > Context Aware Service > Wired > Wired Clients, only users that are assigned to the WCS root virtual domain are able to display context-aware information for these devices.

Refer to Understanding Virtual Domains as a User, *WCS Configuration Guide 7.0* (http://www.cisco.com/en/US/docs/wireless/wcs/7.0/configuration/guide/7_0virtual.html) for a complete list of network resources that are not available in non-root virtual domains.

Catalyst LAN Switch

Each Catalyst LAN switches participating in Context-Aware Services within the Unified Access network design will participate in an NMSP session with the Mobility Services Engine. An important consideration to keep in mind is that while a single MSE can support concurrent NMSP sessions to many Catalyst LAN switches in the network, there is a 1:1 relationship between each participating Catalyst switch and the number of MSEs to which it communicates via NMSP.

Integration with the Unified Access Network Architecture

Clock Synchronization

All components participating in wired endpoint location using Context-Aware Services should have their internal clocks synchronized to a common time source. This includes the Mobility Services Engine, WCS, WLAN controllers, and any Catalyst switches. It is recommended that Coordinated Universal Time (UTC[2]) be utilized for this purpose. Because reliable certificate authentication relies on time-based consistency between participating components, it is important to ensure that context-aware components are time-synchronized throughout the network. In addition, having components synchronized to a common time source can help facilitate troubleshooting, especially when having to review events occurring in the logs of different network components. Any such log output, when coupled with accurate and consistent time stamps, will appear much more logical and will facilitate problem resolution.

The recommended approach to maintain time synchronization across all components is through the use of the Network Time Protocol (NTP).

NTP Configuration of the Mobility Services Engine

Configuration of the NTP server addresses used by the MSE is handled during installation and the execution of the MSE automatic configuration script. An excerpt of that script is shown below. Detailed information regarding the automatic configuration script can be found in the Automatic Installation Script section of the *Mobility Services Engine Getting Started Guide* (http://www.cisco.com/en/US/docs/wireless/mse/3350/quick/guide/mse_qsgmain.html#wp1057105) and in Appendix A of the *Mobility Services Engine Context Aware Deployment Guide* (http://www.cisco.com/en/US/products/ps9742/products_tech_note09186a00809d1529.shtml#appena).

```
If you choose to enable NTP, the system time will be configured from NTP servers that you
select. Otherwise, you will be prompted to enter the current date and time.

NTP is currently disabled.

Configure NTP related parameters? (Y)es/(S)kip/(U)se default (S)kip: Y

Enter whether or not you would like to set up the Network Time Protocol (NTP) for this
machine.

If you choose to enable NTP, the system time will be configured from NTP servers that you
select. Otherwise, you will be prompted to enter the current date and time.

Enable NTP (yes/no) no : yes
Enter NTP server name or address: <IP address or DNS name of NTP server>
Enter another NTP server IP address (or none) none: none
```

NTP Configuration of the Wireless Control System (WCS) Server

Configuration of the internal clock and the specification of which NTP servers to use for periodic time synchronization must be performed on the WCS server using the time and date capabilities of the WCS host operating system in use (either Windows or Linux).

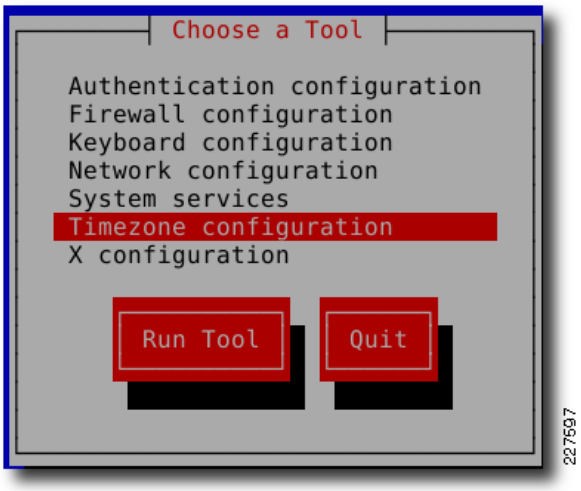
2. UTC may be considered as equivalent to Greenwich Mean Time (GMT) in this case.

RHEL-Based WCS Server

For a Redhat Linux-based WCS server, login to the host OS as root and use the following procedure to synchronize the internal software clock to the NTP server, followed by synchronizing the software clock to the server's hardware clock, and then finally ensuring that synchronization is maintained by starting the ntpd client daemon:

- clock—Displays the current setting of the software clock.
- /etc/init.d/ntpd stop—Stops the ntpd client if it is already running.
- ntpdate <*ntp server name or address*>—Synchronizes the system software clock with the NTP server.
- setup—Brings up a setup utility that allows you to choose to set the time zone (shown in Figure 4-8).
- hwclock --systohc—Writes the software clock settings to the hardware clock.
- /etc/init.d/ntpd start—Starts the ntpd daemon to maintain clock synchronization going forward.[3]

Figure 4-8 RHEL Setup Utility



Note There are various other approaches that can be used to select the time zone on a Linux host system. The reader is encouraged to consult the Redhat documentation for methods involving the use of the TZ variable or symbolic links to the localtime file or a particular time zone file in the system's time zone directory.

Windows 2003-Based WCS Server

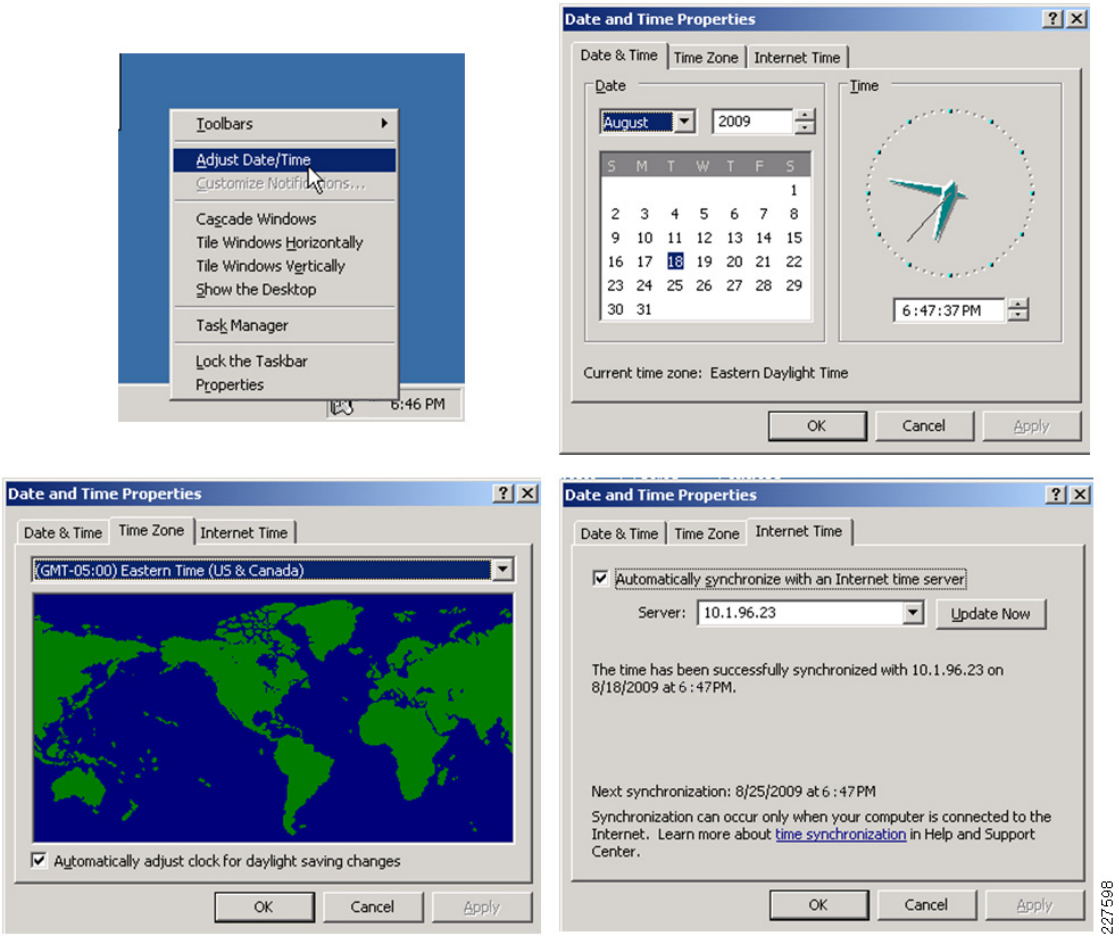
For a WCS server based on the Microsoft Windows 2003 Server OS, use the following procedure to synchronize and maintain the correct system time via the Windows Time service (see Figure 4-9):

1. Check **Settings>Control Panel >Administrative Tools>Services** for the Windows Time service and ensure that it has been started.
2. Right click on the **Task Bar** clock and select **Adjust Date/Time**.

3. If ntpd does not start as part of your system boot script, consider adding it using the command **chkconfig --add ntpd**.

3. Under the **Date & Time** tab, set the current date and clock time to the approximate time of your NTP server.
4. Set the Time Zone and Daylight Savings time selections appropriately.
5. Select the Internet Time tab, check the box to Automatically Synchronize With An Internet Time Server, type in the DNS name or address of your NTP server, and then click **Apply**.

Figure 4-9 Setting Time and NTP Server on Windows 2003



NTP Configuration of Context-Aware Catalyst Ethernet Switches

In order to prevent any issues with certificate authentication and NMSP session initiation, Catalyst Ethernet switches participating in Context-Aware Services should be configured to utilize NTP in order to keep their clocks in synchronization with other context-aware components. NTP is configured similarly among the various switch models discussed in this chapter and the most comprehensive information on how to configure the NTP client within a Catalyst switch can usually be found in the configuration guide for the particular switch model. For example, the Catalyst 3560 NTP configuration is documented in the Configuring NTP section of the *Catalyst 3560 Switch Software Configuration Guide* (http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/swadmin.html#wp1053923).

Similar guidance regarding the use of NTP can be found in the configuration guides for the Catalyst 3750 and 4500 series.

It is best practice to ensure time synchronization of all network components whenever possible. However, from the perspective of Context-Aware Services in the Unified Access design, only those switches that are actually participating in an NMSP session with the MSE require mandatory clock synchronization.

Context-Aware Service Parameters-Tracking

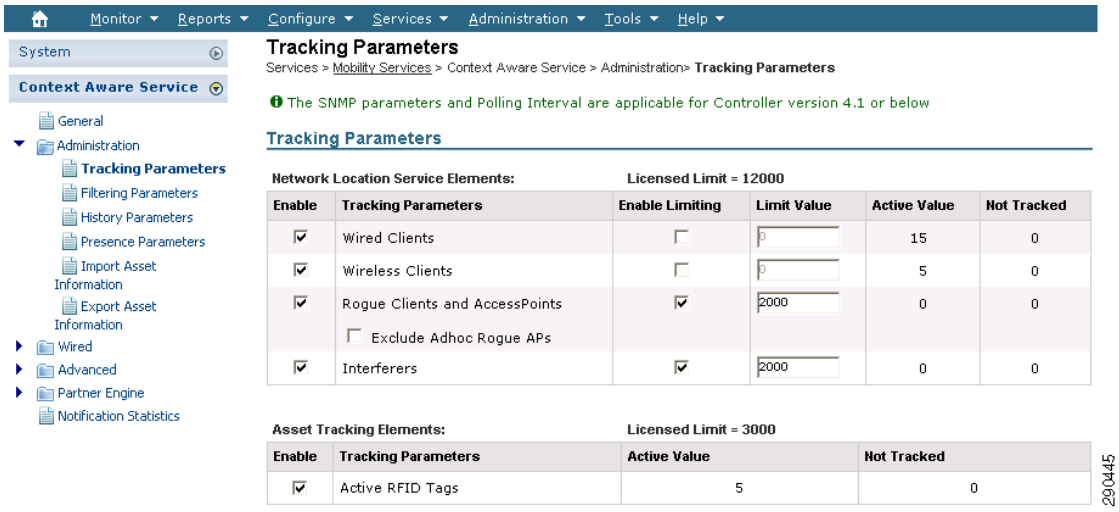
As mentioned earlier, Context-Aware Services can track up to a maximum of 18,000 licensed devices when using the MSE-3350/3355 hardware platform and up to a maximum of 2,000 licensed devices when using the MSE-3310 platform. The absolute limit on the number of devices that can be tracked is determined by the hardware platform used, the presence of any other applications co-residing on the MSE, and the level of licensing purchased. The WCS tracking parameters configuration panel (located at Services > Mobility Services > Context Aware Service > Administration> Tracking Parameters) allows the administrator to pre-determine just how much of the MSE's maximum licensed tracking capacity will be allocated towards the tracking of specific device categories. This is useful in order to allow the tracking of device categories such as rogue access points, rogue clients, or wireless interferers, but also limit these unpredictable categories such that an uncontrolled introduction of rogues or interferers is not allowed to consume all of the device tracking capacity on the MSE.

We can use the Context-Aware Service Tracking configuration to:

- Entirely enable or disable the tracking of wired or wireless client stations, asset tags, rogue access points, rogue clients, and interferers.
- Set limits on how much MSE tracked device capacity will be allocated to certain device categories. Figure 4-10 provides us with an example of how this can be achieved, where the maximum number of rogue access points/clients and the maximum number of wireless interferers are capped at 2,000 devices each. No limit value has been placed on any other categories, which in effect means that the maximum number of wired and wireless devices tags tracked will be allowed to rise until the licensing limits are reached. But due to the limit values imposed on them, the MSE would never track more than 2,000 wireless rogues or interferers.

Note Any devices that are detected but excluded from tracking due to the enforcement of a tracking limit will be reflected in the "Not Tracked" device count column shown on the right side of the display. This is very useful to both the designer and the network administrator in that it allows for straightforward verification of license sufficiency post-deployment.

Figure 4-10 Mobility Services Engine Tracking Parameters



Catalyst Switch Definition and Synchronization

To allow for proper tracking of the devices that may be registered or attached to them, context-aware Catalyst switches must be defined to WCS and then synchronized with the Mobility Services Engine.

Detailed information about adding Catalyst switch definitions to WCS using the WCS Configure > Add Ethernet Switches menu panel can be found in the *WCS Configuration Guide 7.0* (http://www.cisco.com/en/US/docs/wireless/wcs/7.0/configuration/guide/7_0ctrlcfg.html#wp1089752) .

Detailed information about synchronizing the MSE with context-aware Catalyst switches using the WCS Services >Mobility Services > Synchronize WCS and MSE(s) menu panel can be found in the Synchronizing Mobility Services Engines chapter of the *Context-Aware Service Configuration Guide 7.0* (http://www.cisco.com/en/US/docs/wireless/mse/3350/7.0/CAS/configuration/guide/msecg_ch3_CAS.html#wp998995).

Usage Considerations for Wired Endpoint Device Tracking

Beginning with release 6.0 of Cisco Context-Aware Services, civic and ELIN location attributes can be assigned to devices connected to Cisco Catalyst switch ports, such as the 3560, 3750, 4500, and 4900 series. As participants in Context-Aware Services, switches provide relevant contextual information for all the wired device endpoints that attach to them. These endpoints may include IP phones, PCs, host servers, access points, and so on. The NMSP protocol is used between the switches and MSE to deliver this contextual information to the MSE. Location information may include the physical street location address (also known as the civic address) as well as other information about endpoints such as their IP address, MAC address, port, VLAN, and 802.1x username. If the end device makes use of CDP or LLDP-MED, additional endpoint device information, such as the version number of its operating system and its hardware serial number, can also be sent to the MSE.

In the Unified Access design, redundancy is provided such that should a Catalyst switch stack member or modular component responsible for NMSP communications fail, the NMSP session will automatically recover once communication is re-established using a redundant component. Thus, in our test bed configuration, using the default NMSP parameter settings on the MSE, we observed that the total time

required for our nine member 3750X switch stack to recover from stack master failure and re-establish NMSP communications was under 60 seconds. Typically, about 50-55 seconds of this elapsed time was spent performing switch stack recovery and election of a new stack master, with only a three to seven second additional delay observed before the NMSP session itself was seen to be fully recovered and operational.

Hardware and Software Requirements for Wired Device Tracking

As mentioned previously, at the current time wired device tracking is only performed on Catalyst switch hardware such as the 3560, 3750, 4500, and 4900 series.

Note Readers should note that cryptography-enabled (k9) switch images are mandatory in order to enable NMSP functionality in Catalyst switches.

Enabling the tracking of wired endpoint devices in a context-aware Unified Access network requires that the Mobility Services Engine be licensed for the expected number of wired devices you anticipate tracking. This is especially important in networks where wired device tracking is being utilized alongside the tracking of wireless devices, rogues, interferers, and RFID tags, as the license represents the upper limit on the total number of devices tracked.

Each context-aware switch that is enabled for wired device tracking in your network establishes one NMSP session to the MSE. Therefore, when enabling numerous switches for context-aware wired device tracking, it is recommended that you plan for the total number of MSEs that may be required to support the total number of NMSP sessions in your network. This is especially relevant in Unified Access network designs, where Context-Aware Services will likely be enabled for both wired Catalyst switches as well as wireless LAN controllers. In very large networks consisting of many switches and WLAN controllers, you may find that more than one MSE is required to accommodate all anticipated NMSP sessions. Although a single MSE is intended to support up to 500 NMSP sessions, Cisco scalability testing has validated this capability to an maximum of 100 simulated NMSP connections.

Enabling Context-Aware Wired Device Tracking

In order to track wired devices on Catalyst switch ports, each switch whose devices we wish to track must be configured to enable NMSP and other important parameters and to contain the appropriate location information for each switch port. In addition, WCS must be configured to be aware of the context-aware switches in the network and to be able to communicate with them. WCS is also used to transmit information about the switches to the Mobility Services Engine and initiate the synchronization process.

A complete, step-by-step guide to configuring Catalyst switches and WCS for wired device tracking can be found in the *Context Aware Service Configuration Guide 7.0* (http://www.cisco.com/en/US/docs/wireless/mse/3350/7.0/CAS/configuration/guide/msecg_ch7_CAS.html#wp1224011).

In addition, the following chapters and documents provide valuable and detailed background information concerning the wired device tracking capability of Catalyst switches:

- Configuring LLDP, LLDP-MED, and Wired Location Service in the *Catalyst 3750 Switch Software Configuration Guide* http://www.cisco.com/en/US/docs/switches/lan/catalyst3750/software/release/12.2_55_se/configuration/guide/swlldp.html

- Configuring LLDP, LLDP-MED, and Wired Location Service in the *Catalyst 3560 Switch Software Configuration Guide*
  http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/swlldp.html
- Configuring LLDP and LLDP-MED in the *Catalyst 4500 Series Switch Software Configuration Guide*
  http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/54sg/configuration/guide/swlldp.html#wp1097119

Readers are reminded that:

- NMSP is disabled on Catalyst switches by default and must be explicitly enabled via the **nmsp enable** global configuration command.
- IP device tracking is disabled by default on Catalyst switches and must be enabled in order for Context-Aware wired device tracking to function properly. It can be enabled by issuing the **ip device tracking** in global configuration mode on the context-aware switch.
- The civic location identifier in the LLDP-MED TLV is limited to 250 bytes or less. To avoid receiving error messages regarding available buffer space during switch configuration, the total length of all civic location information specified for each individual civic-location identifier must not exceed 250 bytes.
- In Release 7.0, all wired device client and switch tracking is available only to the root WCS virtual domain user. Because of this, you may wish to limit the use of context-aware wired device tracking in this release to only those users with whom you are comfortable assigning WCS root virtual domain privileges.

NMSP Attachment Notification Interval

After an NMSP session is established between the MSE and a context-aware Catalyst switch, the MSE transmits an echo response packet to the switch every echo interval time period. The echo interval is specified on the MSE using the WCS menu entitled Services > Mobility Services > System > NMSP Parameters and applies to all NMSP session partners. The echo interval can be set from 1 to 120 seconds, with the default being 15 seconds.

More information about NMSP session parameters can be found in the *Cisco Context-Aware Service Configuration Guide 7.0* at:
http://www.cisco.com/en/US/docs/wireless/mse/3350/7.0/CAS/configuration/guide/msecg_ch4_CAS.html#wp1014368.

In addition to echo responses, the switch will periodically send attachment notifications to the MSE via the NMSP session. Any link-up or link-down events that are detected by the switch are aggregated during a configurable time interval and sent to the MSE via an attachment notification at the conclusion of that time interval. This interval is known as the nmsp attachment notification interval and is configurable on the switch via the **nmsp notification interval attachment** *interval-seconds* global command. The range of values for interval-seconds is from 1 to 30 seconds, with 30 seconds being the default.

The setting chosen for **nmsp notification interval attachment** will impact how quickly changes in device attachment are propagated from switches to the MSE. Shorter settings result in changes in device attachment being reflected faster, but at the cost of increased switch activity and NMSP network traffic. Longer settings result in more efficient aggregation of attachment information in NMSP packets, but changes in device attachment will not be reflected at the WCS as quickly.

In large networks where there are many NMSP sessions active to the MSE and the number of users connecting and disconnecting from each switch is high, configuring **nmsp notification interval attachment** to a very low number can increase the workload on each switch and increase the amount of NMSP traffic generated between switches and the MSE, creating an unnecessary burden on your switches, your network, and the MSE.

Civic Address Configuration

The information contained in the *Context Aware Service Configuration Guide 7.0* (http://www.cisco.com/en/US/docs/wireless/mse/3350/7.0/CAS/configuration/guide/msecg_ch7_CAS.html#wp1224011) provides the guidance necessary to configure context-aware switches and the WCS for wired device tracking.

As can be seen in the *Context Aware Service Configuration Guide 7.0*, all configuration of civic and ELIN location information is performed on the Catalyst switch. This is normally done either directly using the switch CLI or by uploading a switch configuration text file that has been created offline. Once a switch is configured with the desired civic and ELIN location information, the switch will share all of the configured port information with the MSE when the NMSP session is initially established and will periodically update the MSE if any location updates are performed.

Readers should find the following IETF RFC documents helpful in better understanding the types of values that should be specified for the various civic location fields:

- RFC 4776 (http://www.ietf.org/rfc/rfc4776.txt)
- RFC 4589 (http://www.ietf.org/rfc/rfc4589.txt)
- RFC 5139 (http://www.ietf.org/rfc/rfc5139.txt).

During the course of lab testing, we noted the following behaviors that should prove useful to the network designer or network administrator planning to deploy Context-Aware Services and wired endpoint location:

- Civic and ELIN location configuration scope—In both code trains, civic location and ELIN information may be defined at a global level and then assigned to each switch interface using the appropriate civic or ELIN location identifier. The following example illustrates the process to define both a civic and ELIN location identifier and then assign both to an interface:

```
Switch(config)# location civic-location identifier bldg_17
Switch(config-civic)# number 3550
Switch(config-civic)# primary-road-name "Cisco Way"
Switch(config-civic)# city "San Jose"
Switch(config-civic)# state CA
Switch(config-civic)# building 19
Switch(config-civic)# room C6
Switch(config-civic)# county "Santa Clara"
Switch(config-civic)# country US
Switch(config-civic)# end
Switch(config-civic)# exit
Switch(config)# location elin-location 2122315596 identifier 1
Switch(config)# interface gigabitEthernet 1/0/1
Switch(config-if)# location civic-location-id bldg_17
Switch(config-if)# location elin-location-id 1
Switch(config-if)# exit
Switch(config)#exit
Switch#
```

If more than one switch port shares the same civic location or ELIN, then the same globally specified civic and ELIN location identifiers can be assigned. One such case where this would make sense might be where a user requires three separate switch ports to be terminated at the same physical location in a

building. In this case, a common civic-location identifier can be used, since all three ports will be present at the same building address, floor, room, and seat. Since each termination will bear a unique wiring jack identifier, this information can be uniquely applied to each interface using the interface location parameter **additional-location-information**. Any information specified using **additional-location-information** will appear in the "Address Line 2" area under the Civic Address panel for the device. Example 4-1 is an excerpt of a switch configuration illustrating how a common civic-location identifier and **additional-location-identifier** can be used:

Example 4-1 Common civic-location Identifier Shared among Multiple Interfaces with additional-location-information

```
location civic-location identifier bldg_300
 building "Building 300"
 city Alpharetta
 country US
 county Fulton
 name "ABC Corporation"
 postal-code 30022
 state Georgia
 street-group "Kimball Bridge Road"
 number 9300
 floor 2
 room 212
 type-of-place "Shipping & Receiving"
!
location elin-location 19789363021 identifier 1009
location elin-location 19789363022 identifier 1010
location elin-location 19789363023 identifier 1011
.
.
interface GigabitEthernet1/0/9
 location civic-location-id bldg_300
 location elin-location-id 1009
 location additional-location-information "A10-Seat 5 Jack 1"
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/10
 location civic-location-id bldg_300
 location elin-location-id 1010
 location additional-location-information "A10-Seat 5 Jack 2"
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/11
 location civic-location-id bldg_300
 location elin-location-id 1011
 location additional-location-information "A10-Seat 5 Jack 3"
 switchport access vlan 106
 switchport mode access
```

In some cases, the switch might contain ports for which the assigned civic location information is largely unique. One approach to handling this would be to define unique civic-location-identifier definitions globally and apply those definitions to ports on a 1:1 basis. This technique was introduced in version 6.0, remains largely unchanged, and functions consistently across all supported Catalyst switching products. It is recommended when civic location identifiers differ greatly, such as when some ports are located in a building with a different name, address, organization, type, and so on. This is illustrated in

Example 4-2, where a switch port terminates in an neighboring outbuilding (Building 301), as well as a building that is just across the adjacent state highway but technically in a different city and county (Building 400).

Example 4-2 Unique civic-location-identifier Assigned to Each Port Interface

```
location civic-location identifier bldg_300
 building "Building 300"
 city Alpharetta
 country US
 county Fulton
 name "ABC Corporation"
 postal-code 30022
 state Georgia
 street-group "Kimball Bridge Road"
 number 9300
 floor 2
 room 212
 type-of-place "Shipping & Receiving"
!
location civic-location identifier bldg_301
 building "Building 301"
 city Alpharetta
 country US
 county Fulton
 name "ABC Corporation"
 postal-code 30023
 state Georgia
 street-group "Waters Road"
 number 310
 floor 1
 room 000
 type-of-place "Receiving Overflow"
 additional-code ""B13-Seat 22 Jack 9"
!
location civic-location identifier bldg_400
 building "Building 400"
 city Milton
 country US
 county Milton
 name "ABC Express"
 postal-code 30004
 state Georgia
 street-group "Freeman Road"
 number 55
 floor 4
 room 411
 type-of-place "Cust Pickup"
 additional-location-information "K64-Seat 7 Jack 13"
!
!
location elin-location 19789363024 identifier 1012
location elin-location 19789363025 identifier 1013
location elin-location 19789363026 identifier 1014
.
.
interface GigabitEthernet1/0/12
 location civic-location-id bldg_300
 location elin-location-id 1012
 location additional-location-information "A12-Seat15 Jack 1"
 switchport access vlan 106
 switchport mode access
```

```
 switchport mode access
 .
 .
interface GigabitEthernet1/0/13
 location civic-location-id bldg_301
 location elin-location-id 1013
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/14
 location civic-location-id bldg_400
 location elin-location-id 1014
 switchport access vlan 107
 switchport mode access
 .
 .
```

Note the use of **additional-location-information** embedded within the **civic-location-identifier** for Building 400. It is specified here instead of as a standalone statement at the GigabitEthernet1/0/14 interface. Also, note the use of **additional-code** within the **civic-location-identifier** for Building 301. We use this field in this example instead of **additional-location-information** simply to demonstrate its use. Whereas information specified for **additional-location-information** will appear in the device's "Address 2" field on the WCS Civic-Address panel, information specified for **additional-code** will appear in the "additional code" field under the WCS Advanced information panel.

Note **Additional-code** cannot be specified as a standalone interface location command in the same manner as **additional-location-information**.

But what about the case where all ports in a high-density Catalyst switch terminate at a location where the civic location information remains largely the same, except for a few parameters such as floor, room, or seat[4]? Or the case where an entire switch is used to service a single floor of a building, where the only civic location parameters that vary are room and seat? Indeed, both of these are very common occurrences in modern enterprise office complexes. What is the most efficient manner to address this without creating inordinately long switch configurations or requiring a massive amount of redundant information to be entered?

Civic Address Port-Location

Until the availability of IOS code trains supporting the **port-location** interface subcommand set, there were just two approaches to this situation. The choice of which approach to use depended on whether we required information to appear in their labeled areas on the Civic Address and Advanced information screens for each device. If we wanted to see the floor number display under the "Floor" heading, the room number display under the "Room" heading, and so on, then our only option was to specify a unique, fully-qualified civic-location-identifier for each and every port (as used in Example 4-2). The downside to this approach is the amount of labor required to create large switch configurations for high density, context-aware Catalyst switches, with a unique civic-location-identifier global definition for each interface.

4. It is assumed that "seat" can be used to specify a seat in a large open area, a cubicle, an office, or some type of combined location code such as pole number/cubicle (e.g., 3C/002). For equipment such as servers, seat could be used to indicate rack and slot number, although **additional-location-information** is probably better suited for that purpose.

If, however, we could suffice with the port-specific information being placed into a format that would fit within the **additional-location-information** field, then we could use as little as one civic-location identifier common to all ports in the switch and specify the floor/room/seat/jack in a concatenated fashion (e.g., 02-212-A45-01) using the **additional-location-information** field at the interface level. This approach was shown in Example 4-1.

While both of these solutions are capable of fulfilling the task at hand, clearly neither is ideal. In our validation, however, we observed that the code trains tested contained a new **port-location** subcommand set within the **location civic-location** interface command:

location {**additional-location-information** *word* | **civic-location-id** *id* [**port-location**] | **elin-location-id** *id*}

Port-location is optional and is used to specify one or more port-specific location attributes. After entering the **location civic-location-id** *id* **port-location** command, the user is placed into the civic location port subcommand configuration mode. In this mode, the user can enter additional location attributes for every port using the same location parameters that were available under the global civic-location identifier.

The CLI command help feature (?) lists details regarding the location attributes that can be configured in this mode.

```
cr22-3750s-LB#conf t
cr22-3750s-LB(config)#interface gigabitEthernet 1/0/9
cr22-3750s-LB(config-if)#location civic-location-id bldg_300 port-location
cr22-3750s-LB(config-if-port)#?
Civic location configuration mode:
  additional-codeSet additional code, CA Type 32
  additional-location-information  Set additional location info, CA Type 22
  branch-road-name                 Set branch road name, CA Type 36
  building                       Set building information, CA Type 25
  city                             Set city name, CA Type 3
  country                          Set the country id
  county                           Set county name, CA Type 2
  division                         Set city division name, CA Type 4
  floor                            Set the floor number, CA Type 27
  landmark                         Set landmark information, CA Type 21
  leading-street-dir               Set leading street direction, CA Type 16
  name                             Set resident name, CA Type 23
  neighborhood                     Set neighborhood information, CA Type 5
  number                           Set the street number, CA Type 19
  post-office-box                  Set post office box, CA Type 31
  postal-code                      Set postal code, CA Type 24
  postal-community-name            Set the postal community name, CA Type 30
  primary-road-name                Set primary road name, CA Type 34
  road-section                     Set road section, CA Type 35
  room                             Set room information, CA Type 28
  seat                             Set seat information, CA Type 33
  state                            Set state name, CA Type 1
  street-group                     Set street group, CA Type 6
  street-name-postmodifier         Set street name postmodifier, CA Type 39
  street-name-premodifier          Set street name premodifier, CA Type 38
  street-number-suffix             Set street number suffix, CA Type 20
  street-suffix                    Set the street suffix, CA Type 18
  sub-branch-road-name             Set sub branch road name, CA Type 37
  trailing-street-suffix           Set trailing street suffix, CA Type 17
  type-of-place                    Set type of place, CA Type 29
  unit                            Set unit, CA Type 26
```

If a civic-location attribute is configured globally as well as on the interface using **port-location**, the **port-location** configuration takes precedence. In this way, **port-location** could be thought of as a method with which to “override” the attributes specified by the global civic-location identifier.

Example 4-3 illustrates the usefulness of **port-location** when configuring a high-density switch stack where all ports in the stack are used to service the same building. In this case, only the floor, room, and seat civic location parameters vary from port to port, with all other civic location attributes remaining constant. **Additional-location-information** is used at the port level to specify information relating to Ethernet jack numbering.

Example 4-3 Use of Common civic-location Identifier with port-location

```
location civic-location identifier bldg_405
 building "Building 405"
 city Milton
 country US
 county Milton
 name "ABC Corporation"
 postal-code 30004
 state Georgia
 street-group "Bethany Rd"
 number 693
 type-of-place "Test Facility"
!
!
location elin-location 19789363027 identifier 1015
location elin-location 19789363028 identifier 1016
location elin-location 19789363029 identifier 1017
location elin-location 19789363030 identifier 1018
location elin-location 19789363031 identifier 1019
.
.
interface GigabitEthernet1/0/15
 location civic-location-id bldg_405 port-location
   floor 1
   room 117
   seat 1A/027
 location elin-location-id 1015
 location additional-location-information "1A/117/J1"
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/16
 location civic-location-id bldg_405 port-location
   floor 2
   room 222
   seat 1G/028
 location elin-location-id 1016
 location additional-location-information "1G/222/J2"
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/17
 location civic-location-id bldg_405 port-location
   floor 3
   room 314
   seat 3F/007
 location elin-location-id 1017
 location additional-location-information "3F/314/J1"
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/18
```

```
 location civic-location-id bldg_405 port-location
   floor 4
   room 431
   seat 2Y/107
 location elin-location-id 1018
 location additional-location-information "2Y/107/J3"
 switchport access vlan 106
 switchport mode access
 .
 .
interface GigabitEthernet1/0/19
 location civic-location-id bldg_405 port-location
   floor 5
   room 516
   seat 5E/044
 location elin-location-id 1019
 location additional-location-information "5E/516/J1"
 switchport access vlan 106
 switchport mode access
 .
 .
```

The utility of the **port-location** subcommand mode should be obvious from the preceding example. Using **port-location**, minor changes to global civic-location information can be easily made at the interface level. Common civic location parameters can now be used across the switch where necessary and they can be expanded on or overridden in a very straightforward fashion.

Civic Location Caveats

During validation, we made note of a few minor caveats concerning the use of **port-location** with the access switches in the test bed:

- Civic-location information added at the interface level using the **port-location** subcommand set is not propagated from the switch to the Mobility Services Engine unless:

  1. A change is made to any of the global civic-location identifiers.

     Or:

  2. The Catalyst switch is unassigned and re-assigned to the MSE in WCS and then re-synchronized.

     Or:

  3. The Catalyst switch is rebooted.

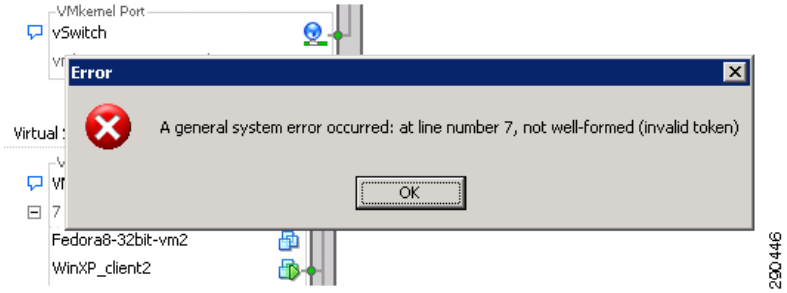
  Since we consider the **port-location** capability to be a significant usability enhancement to the configuration of civic location information in Catalyst switches, we recommend #2 above as a temporary workaround. This resynchronization process needs to occur only once after all **port-location** changes have been made to the switch configuration. This workaround has the advantage of being non-disruptive to any data plane traffic to or from the switch.

- Although it is possible to specify “additional-location-information” as a **port-location** subcommand, we noticed during validation that doing so does not result in propagation of the information being added to the MSE. We identified a workaround to this caveat as simply being the use of the standalone **additional-location-information** interface level command instead.

Other Considerations and Caveats

- If you are using Location MAC Filtering (Services > Mobility Services > Context Aware Service > Administration> Filtering Parameters) to specifically limit or block tracked wireless clients and tags, be advised that these address filters apply to wired device clients as well. Make sure that any filtering specifications that you set using Location MAC Filtering are flexible enough to allow tracking of not only your wireless clients and tags, but wired devices as well. Any devices that have been blocked from location tracking as a result of a defined filter will be viewable under the "Blocked MACs" listing on the Filtering Parameters page. Detailed information regarding how to configure Location MAC filtering can be found in the Modifying Filtering Parameters section of the *Cisco Context-Aware Service Configuration Guide 7.0* (http://www.cisco.com/en/US/docs/wireless/mse/3350/7.0/CAS/configuration/guide/msecg_ch7_CAS.html#wp1100062).
- In order to ensure that serial number information collected by Catalyst switches from attached Cisco IP phones is forwarded to the MSE, CDP should be disabled on the switch interface as a temporary workaround. This behavior was observed in testing with both 9971 and 7975 IP phones.
- ESX server 3.5 and above is installed with Cisco CDP enabled by default on the virtual switches. This can cause unreliable detection of virtual machines and errors at the ESX server (shown in Figure 4-11)

Figure 4-11 VMWare ESX Error with CDP Enabled



- In order to avoid such errors and allow reliable detection of virtual machine attachment and disconnection, CDP should be disabled. There are two approaches to accomplishing this:
  - Disabling CDP at the switch interface using the **no cdp enable** IOS interface configuration command.
  - Disabling CDP on the ESX server via the following procedure, replacing *yourVSwitch* by the name of the virtual switch installed on ESX server. Connect to the service console of the ESX server and enter these commands for each virtual switch:
    - **esxcfg-vswitch -B listen** *yourVSwitch*
    - **esxcfg-vswitch -B down** *yourVSwitch*
- If a change is made to the value specified for an ELIN, one of the following actions should be performed in order to propagate this change to the MSE:
  1. A change is made to any of the global civic-location identifiers.

     Or:
  2. (Recommended) The Catalyst switch is unassigned and re-assigned to the MSE in WCS, and re-synchronized.

Or:

3. The Catalyst switch is rebooted.

Since it has zero impact on other users that may be passing data through the switch, #2 workaround is recommended.

Hardware and Software Releases

Table 4-5 Hardware and Software Releases Tested

Component	Comments
Wireless Control System, Release 7.0	For licensing and part number information, see http://www.cisco.com/en/US/prod/collateral/wireless/ps5755/ps6301/ps6305/product_data_sheet0900aecd804b4646.html
Mobility Services Engine 3300, Release 7.0	For licensing information, see http://www.cisco.com/en/US/prod/collateral/wireless/ps9733/ps9742/data_sheet_c07-473865.html
Catalyst 4500	Access switch; must use crypto (K9) image if Context-Aware Services for wired devices is desired
Catalyst 3750E Switch Stack	Access switch; must use crypto (K9) image if Context-Aware Services for wired devices is desired
Catalyst 3750E, X	Access switch; must use crypto (K9) image if Context-Aware Services for wired devices is desired
Catalyst 3560E, X	Access switch; must use crypto (K9) image if Context-Aware Services for wired devices is desired

Context-Aware Services—General References

The following are recommended references with regard to general best practice deployment recommendations for Cisco Unified Networks and the use of Context-Aware Services release 7.0:

- Context-Aware Solution Deployment Guide
  http://www.cisco.com/en/US/products/ps9742/products_tech_note09186a00809d1529.shtml
- Context-Aware Services Configuration Guide, Release 7.0
  http://www.cisco.com/en/US/docs/wireless/mse/3350/7.0/CAS/configuration/guide/CAS_70.html
- Wireless Control System Configuration Guide, Release 7.0
  http://www.cisco.com/en/US/docs/wireless/wcs/7.0/configuration/guide/WCS70cg.html
- Catalyst 3560 Switch Software Configuration Guide
  http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_55_se/configuration/guide/3560_scg.html
- Catalyst 3750 Switch Software Configuration Guide
  http://www.cisco.com/en/US/docs/switches/lan/catalyst3750/software/release/12.2_55_se/configuration/guide/scg3750.html

- Catalyst 4500 Series Switch Software Configuration Guide
http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/54sg/configuration/guide/config.html