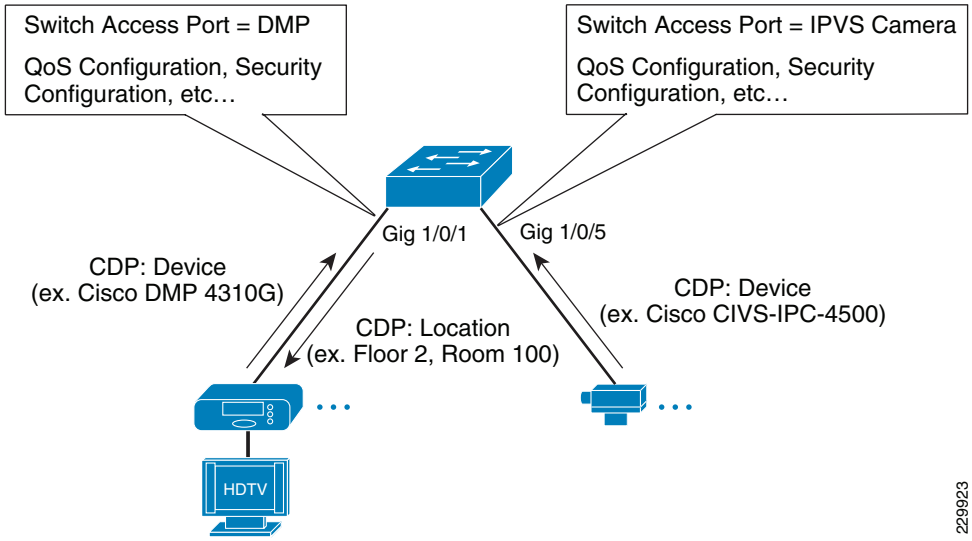
CHAPTER 7

Medianet Auto Configuration

Medianet auto configuration is designed to ease the administrative burden on the network administrator by allowing the network infrastructure to automatically detect a medianet device attached to a Cisco Catalyst switch via the Cisco Medianet Service Interface (MSI) and configure the switch port to support that particular device. Figure 7-1 shows an example with a Cisco digital media player (DMP) and a Cisco IP Video Surveillance (IPVS) camera connected to a Cisco Catalyst switch.

Figure 7-1 Example of Auto Configuration



From an FCAPS perspective, auto configuration is part of configuration management. The current medianet auto configuration functionality includes two features:

- Auto Smartports
- Location Services

Auto Smartports

Auto Smartports (ASP) macros are an extension to Cisco Static Smartports macros. With Static Smartports, either built-in or user-defined macros can be applied manually to an interface by a network administrator. Macros contain multiple interface-level switch commands bundled together under the macro name. For repetitive tasks, such as multiple interfaces which require the same configuration,

Static Smartports can reduce both switch configuration errors and the administrative time required for such configuration. ASP macros extend this concept by allowing the macro to be automatically applied to the interface based upon built-in or user-defined trigger events. The mechanisms for detecting trigger events include the use of Cisco Discovery Protocol (CDP) packets, Link-Level Discovery Protocol (LLDP) packets, packets which include specific MAC addresses or Organizational Unique Identifiers (OUIs), and attribute-value (AV) pairs within a RADIUS response when utilizing ASP macros along with 802.1x/MAB.

Note Triggering an ASP macro by passing a RADIUS AV pair to the Catalyst switch has not been validated at the time this document was written.

Platform Support

Table 7-1 shows Cisco Catalyst switch platforms and IOS software revisions which currently support ASP macros.

Table 7-1 Platform and IOS Revision for Auto Smartports Support

Platform	ASP IOS Revisions	Enhanced ASP IOS Revisions
Catalyst 3750-X Series Switches	12.2(53)SE2	12.2(55)SE
Catalyst 3750, 3560, 3750-E, and 3560-E Series Switches	12.2(50)SE, 12.2(52)SE	12.2(55)SE
Cisco ISR EtherSwitch Modules[1]	12.2(50)SE, 12.2(52)SE	12.2(55)SE
Catalyst 4500 Series Switches	IOS 12.2(54)SG	Future Release
Catalyst 2975 Series Switches	12.2(52)SE	12.2(55)SE
Catalyst 2960-S and 2960 Series Switches	12.2(50)SE, 12.2(52)SE, 12.2(53)SE1	12.2(55)SE

1. This applies to ISR EtherSwitch Modules which run the same code base as Catalyst 3700 Series switches.

There are basically two versions of ASP macros, which are referred to as ASP macros and Enhanced ASP macros within this document. This is due to differences in functionality between ASP macros running on older IOS software revisions and ASP macros running on the latest IOS software revisions. Table 7-2 highlights some of these differences.

Table 7-2 Partial List of Feature Differences Between ASP Macros and Enhanced ASP Macros

Feature	ASP Macros	Enhanced ASP Macros
Macro-of-last-resort	No	Yes
Custom macro	No	Yes
Ability to enable/disable individual device macros	No	Yes
Ability to enable/disable individual detection mechanisms	No	Yes
Built-in ip-camera macro	Yes, without AutoQos	Yes, with AutoQoS

Table 7-2 Partial List of Feature Differences Between ASP Macros and Enhanced ASP Macros

Feature	ASP Macros	Enhanced ASP Macros
Built-in media-player macro	Yes, with MAC-address/OUI trigger	Yes, with CDP trigger or MAC-address/OUI trigger
Built-in phone macro	Yes	Yes
Built-in lightweight access-point macro	Yes	Yes
Built-in access-point macro	Yes	Yes
Built-in router macro	Yes	Yes
Built-in switch macro	Yes	Yes
Built-in detection mechanisms	CDP, LLDP, mac-address, and RADIUS AV pair	CDP, LLDP, mac-address, and RADIUS AV pair

Throughout this document, the term “ASP Macros” is generally used to refer to both the non-enhanced and enhanced Auto Smartports macro functionality. The term “Enhanced ASP Macros” is only used when specific features which are supported by the enhanced Auto Smartports functionality are discussed.

As mentioned above, from a medianet perspective the primary benefit of ASP macros is to ease the administrative burden of provisioning medianet devices onto the IP network infrastructure. Table 7-3 lists the medianet devices currently supported by built-in ASP macros.

Table 7-3 Medianet Devices with Built-in ASP Macros

Device	Models	Software Revisions and Comments
Cisco IPVS Cameras	CIVS-IPC-2400 Series, CIVS-IPC-2500 Series, CIVS-IPC-4300, CIVS-IPC-4500[1]	Revision 1.0.7. CDP detection mechanism only.
Cisco DMPs	Cisco DMP 4305G, Cisco DMP 4400G	Revision 5.2.1. OUI detection mechanism only.
Cisco DMPs	Cisco DMP 4310G	Revision 5.2.2. CDP or OUI detection mechanisms

1. Cisco 5000 Series IPVS cameras currently do not support CDP.

Auto Smartports also has built-in macros for devices which are not specific to a medianet. These devices include routers, switches, access-points, and lightweight (CAPWAP/LWAP enabled) access-points.

Switch Configuration

Auto Smartports macro processing is enabled globally on supported Catalyst switches with the command:

```
macro auto global processing
```

This command also automatically enables ASP macro processing on all switchports. This could lead to unwanted consequences when first enabling ASP macros on a Catalyst switch. For example, the network administrator may not want Auto Smartports to automatically change the configuration of existing

uplink ports connected to infrastructure devices such as switches and routers. Such changes could result in an unintentional service outage when first enabling ASP macros. The network administrator should first disable ASP macro processing on interfaces where it is not desired. The following examples show how to disable ASP macro processing at the interface-level for a single interface and a range of interfaces.

Single Interface	Range of Interfaces
`interface GigabitEthernet1/0/1` `no macro auto processing`	`interface range GigabitEthernet1/0/1 - 48` `no macro auto processing`

Note The **no macro auto processing** interface-level command will currently not appear within the switch configuration—even if it has been typed in—until the **macro auto global processing** global command is entered into the switch configuration. Therefore, the network administrator must manually keep track of which interfaces they have disabled for ASP macro processing before enabling the **macro auto global processing** global command.

The **macro auto global processing** command has one or two optional forms as shown below, depending upon the Catalyst switch platform.

Catalyst Access Switches	Catalyst 4500 Series Switches

`macro auto global processing fallback cdp`	`macro auto global processing fallback cdp`
	Or:
	`macro auto global processing fallback lldp`

These forms of the command may be used when the network administrator has deployed 802.1x or MAB and wishes either CDP packets or LLDP packets to be used for ASP macro trigger detection—after 802.1x/MAB authentication is successful. This functionality may also be enabled per interface with the following interface-level command:

macro auto processing fallback *<fallback method>*

The fallback method can either be CDP or LLDP, depending upon the platform, as discussed above. Security Considerations further describes the use of MAB with CDP fallback.

Note Since none of the medianet devices currently support an 802.1x supplicant, all testing was performed utilizing MAB with CDP fallback only.

By default, all built-in ASP device macros (also referred to as ASP scripts) are enabled when ASP macro processing is enabled on a Catalyst Switch. Table 7-4 shows the built-in device ASP macros, any configurable parameters which can be passed into the macros when they execute, and the default values of those parameters. These can be displayed through the **show macro auto device** command on the Catalyst switch.

Table 7-4 ASP Built-in Device Macros

Macro Name	Cisco Device	Configurable Parameters	Defaults
access-point	Autonomous Access Point	NATIVE_VLAN	VLAN1
ip-camera	Video Surveillance Camera	ACCESS_VLAN	VLAN1
lightweight-ap	CAPWAP / LWAP Access Point	ACCESS_VLAN	VLAN1
media-player	Digital Media Player	Access_VLAN	VLAN1
phone	IP Phone	ACCESS_VLAN, VOICE_VLAN	VLAN1, VLAN2
router	Router	NATIVE_VLAN	VLAN1
switch	Catalyst Switch	NATIVE_VLAN	VLAN1

As listed in Table 7-2, one of the benefits of implementing Enhanced ASP macros is the ability to enable/disable individual built-in device macros. This can be accomplished through the following global switch command:

```
macro auto global control device <list of devices separated by spaces>
```

The list of devices includes one or more of the macro names listed in Table 7-4. For example, in order to enable only the built-in ip-camera and media-player ASP macros, the network administrator would configure the following command on a switch platform which supports Enhanced ASP macros:

```
macro auto global control device ip-camera media-player
```

Built-in device macros can also be enabled/disabled per interface with the following interface-level command:

```
macro auto control device <list of devices separated by spaces>
```

The list of devices includes one or more of the macro names listed in Table 7-4. Security Considerations discusses some potential security reasons why the network administrator may choose to restrict which macros are enabled on a particular switch platform.

With regular ASP macro support, the only way the network administrator can “disable” a built-in macro is to override the macro in such a manner that does nothing. Overriding Built-in Macros discusses this further.

For the most part, the only parameters which can be passed into the built-in ASP macros are VLAN parameters, as shown in Table 7-4. These can be passed using the following global switch configuration command:

```
macro auto device <device> <line>
```

The device is one of the macro names listed in Table 7-4 and line is one of the following forms:

ACCESS_VLAN=*<vlan>*	Used for ip-camera, lightweight-ap, and media-player macros
NATIVE_VLAN=*<vlan>*	Used for access-point, router, and switch macros
ACCESS_VLAN=*<vlan>* **VOICE_VLAN=***<vlan>*	Used for the phone macro

For example, in order to set the access VLAN to VLAN302 for IPVS cameras which use ASP macros, the network administrator would configure the following global switch command:

```
macro auto device ip-camera ACCESS_VLAN=VLAN302
```

From a network design perspective, the ability to set the VLAN for medianet devices is important for two reasons. First, the default macro parameters typically set the access VLAN to VLAN1. Cisco SAFE security best practices have long recommended that network administrators utilize a VLAN other than VLAN1 for devices. Second, the ability to set the VLAN allows different medianet devices to be placed on separate VLANS. This may be beneficial from a traffic isolation perspective, either for QoS or for security purposes. For example, a network administrator may wish to separate all IPVS cameras on a particular Catalyst switch to a VLAN which is separate from normal PC data traffic. The downside of this is that all devices of a particular type are placed into the same VLAN by Auto Smartports. For example, currently there is no ability to place certain DMPs into one VLAN and other DMPs into another VLAN. This may be desirable if two departments within an organization each control their own sets of DMPs and the content to be displayed.

By default, three mechanisms for detecting ASP trigger events are enabled automatically when ASP macro processing is enabled on a Catalyst Switch. These detection mechanisms are shown in Table 7-5.

Table 7-5 ASP Detection Mechanisms

Detection Mechanism Name	Description
cdp	Instructs the switch to look for ASP triggers within CDP packets.
lldp	Instructs the switch to look for ASP triggers within LLDP packets.
mac-address	Instructs the switch to look for either full MAC addresses or the OUI portion of MAC addresses which match in list contained within either a built-in or user-defined MAC-address trigger.

Note The list above does not include the use of an RADIUS AV pair to return a trigger name, which can be used when 802.1x/MAB authentication is enabled as well as ASP macros.

ASP Macro Details details how ASP macros are triggered. With Enhanced ASP macros, the network administrator can disable any of the detection mechanisms via the following global switch configuration command:

```
macro auto global control detection <list of detection mechanism names>
```

The list of detection mechanism names corresponds to one or more of the detection mechanism names in Table 7-5. For example, in order enable only CDP and MAC address detection mechanisms on a given Catalyst switch, the network administrator can configure the following global switch configuration command:

```
macro auto global control detection cdp mac-address
```

Detection mechanisms can also be enabled/disabled per interface with the following interface-level command:

```
macro auto control detection <list of detection mechanism names>
```

From a network design perspective, it may be beneficial to disable unused detection mechanisms if the network administrator knows that there are no devices which will utilize a particular mechanism. This can prevent unexpected switchport configuration changes due to accidental triggering of an ASP macro. For instance, medianet specific devices such as Cisco DMPs and IPVS cameras do not currently support the LLDP protocol. Therefore a network administrator who is interested in using Enhanced ASP macros

to ease the administrative burden of configuring these devices across the network infrastructure may decide to enable only CDP and MAC address detection mechanisms. Finally, note that for regular ASP macros, there is no method of disabling a particular ASP detection mechanism.

One additional command is worth noting. Normally ASP macros are applied to an interface upon detecting a trigger event after link-up and removed upon a link-down event by an anti-macro. Since it is recommended that the interface begin with a default interface configuration (with exceptions when using Location Services, the custom macro, or 802.1x/MAB), the link-down returns the interface to its initial default configuration. The **macro auto sticky global** configuration command causes the macro which is applied upon link-up to remain applied upon link-down. The **macro auto port sticky** interface-level configuration command has the same effect on a port-by-port basis.

The benefit of the **macro auto sticky** and **macro auto port sticky** commands is that the macro is only run once when the medianet device is first seen on the interface, versus every time the interface is transitioned from down to up. The running configuration of the switch always shows the applied macro as well, regardless of whether the device is currently up or down. This may be beneficial from a troubleshooting perspective. The downside is that ASP macros which include the **switchport port-security** command may cause the interface to go into an error-disabled state should another device with a different MAC address be placed onto the switchport.

This document is primarily concerned with the built-in ip-camera and media-player ASP macros, since they relate directly to medianet devices. The built-in access-point and lightweight-ap ASP macros were not evaluated for this document. Future revisions of the Medianet Reference Design may include design guidance regarding wireless connectivity and video. The built-in phone macro was evaluated only from the perspective of its effect on medianet devices such as Cisco TelePresence (CTS) endpoints and desktop video conferencing units which consists of a PC running software daisy-chained to an IP phone.

ASP Macro Details

An understanding of the implementation of ASP will assist in general troubleshooting, customization, and security considerations. The macros are fairly transparent and supported by several useful **show** commands and debugging tools. The logical flow is organized into three distinct functions: detection, policy, and function. Detection is used to determine that an actionable event has occurred and selects an appropriate method to classify the device. Three detection methods are available. These are neighbor discovery using either LLDP or CDP, Mac Address, or 802.1x identity. They can be seen with the IOS exec command:

```
sh macro auto event manager detector all
```

```
No.  Name                 Version   Node        Type
1    identity             01.00     node0/0     RP
2    neighbor-discovery   01.00     node0/0     RP
3    mat                  01.00     node0/0     RP
```

A detail of each detector is also available that lists more information concerning events and variables that are passed back and forth between the IOS event manager and the ASP detector. The details are not explained here. It is only important to know that ASP starts when IOS calls one or more of these detectors and passes information such as interface, mac-address, and link events into the detector. The detectors are associated to a policy that can examine the variables that are passed and make a decision. The policy generates an event trigger. These policy shell scripts do the major work within ASP. The link between detector and policy can be seen with the **show** command:

```
sh macro auto event manager policy registered
```

Typically six policies are registered with the three detectors. Output from the **show** command is summarized in Table 7-6.

Table 7-6 Policies Associated With ASP Detectors

	Detector	Policy	Event
1	neighbor-discovery	Mandatory.link.sh	link-event down
2	neighbor-discovery	Mandatory.link2.sh	link-event admindown
3	neighbor-discovery	Mandatory.lldp.sh	lldp update
4	mat	Mandatory.mat.sh	use-mat-db yes hold-down 65.000
5	neighbor-discovery	Mandatory.cdp.sh	cdp update
6	identity	Mandatory.identity.sh	aaa-attribute {auto-smart-port}

As an example, when a link-event down occurs, neighbor discovery will run the script Mandatory.link.sh. Details of the script can be seen with the command:

```
sh macro auto event manager policy registered detailed <policy script>
```

The scripts can be read with a little background in programming. It is possible to register user-generated scripts, although the details of that procedure are not included in this document. There are significant differences in the system scripts packaged in Auto Smartports and those found in Enhanced Auto Smartports. Each script fetches information from the router configuration, such as the current macro description. Based on the calling event, passed variables, and interface configuration, the policy script generates a trigger. Triggers are mapped to shell functions. This mapping can be seen with the command:

```
sh shell trigger
```

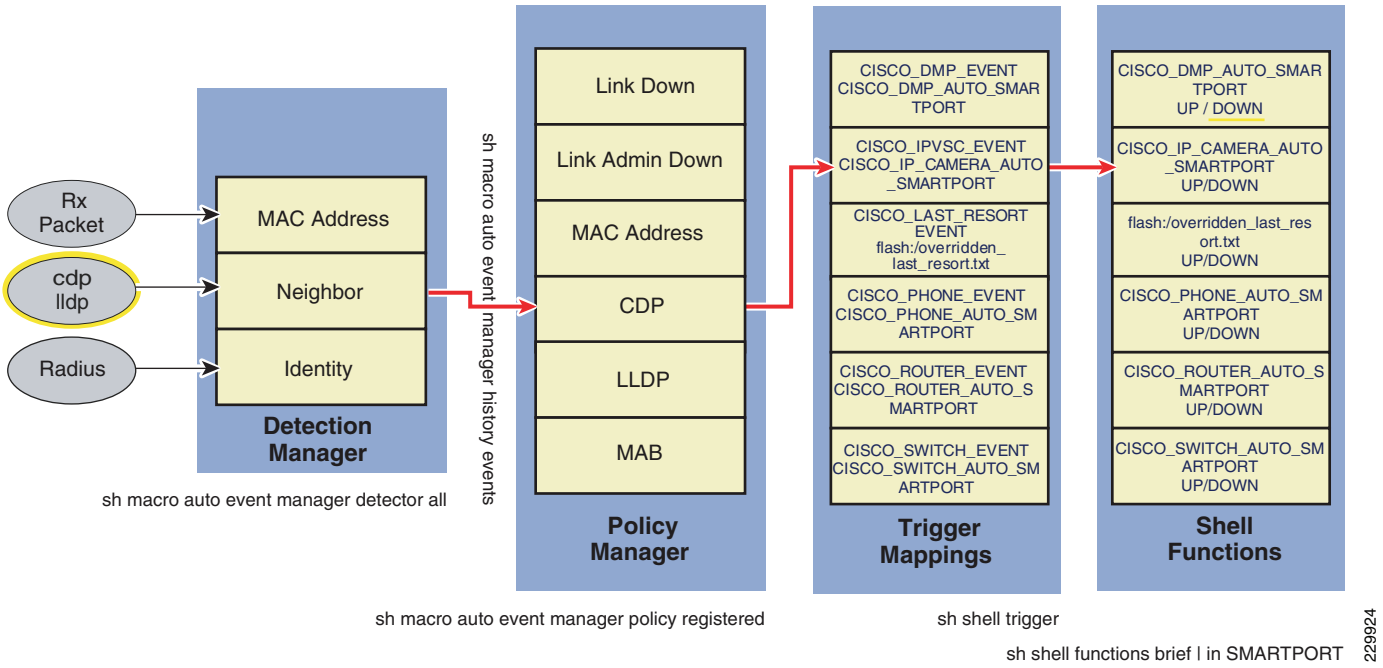
This displays all of the mapped triggers. However ASP is only relevant to those triggers that map to a function that contains AUTO_SMARTPORT in the function name. Arguments are passed into the shell function from the trigger. Common arguments include $LINKUP, $INTERFACE, $TRIGGER, and $ACCESS_VLAN. With this information, the function applies the appropriate configuration to the interface of the switch. The functions can be customized. The shell function details can be seen with the command:

```
show shell function
```

As an example, consider the case were a CDP packet is received on an interface that was previously configured with the appropriate ASP configuration. Neighbor-discovery calls the script Mandatory.cdp.sh. The script first checks to see if CDP detection is available; if so, then the CDP capabilities are checked. If the host bit is set, then the CDP platform type is compared against known types. The previous trigger is noted by pulling the macro description from the interface configuration. Another check is made to see if discovery is enabled for that particular type of device. If so, then the script continues to check the other capabilities bits for Phone, Access Point, Router, or Switch. If the host bit is set in conjunction with the phone bit, then the phone trigger takes precedence. Finally a trigger is generated and mapped to a shell function. Different policies can generate the same trigger. For example, both Mandatory.link.sh and Mandatory.cdp.sh can generate a CISCO_DMP_EVENT trigger, but pass the variable LINKUP with a different value into the shell function. The event policy has the logic to handle various situations, such as the case where the new trigger is the same as the previous trigger that was last used to configure the interface. The event policy also checks to see if the interface is configured with a sticky macro. These are not removed when the link is down. As discussed previously, this could result in an err_disabled state if a different device is attached to a sticky interface with port security. Sticky configurations should not be used if the intent is to dynamically configure the interface based on device discovery when devices move from port to port.

The relationship between the various components is shown in Figure 7-2. The example flow shows the result of a CDP event.

Figure 7-2 Auto Smartports Event Flow



Medianet Devices with Built-in ASP Macros

The following devices are currently supported by built-in ASP macros.

Cisco IPVS Cameras

Cisco IPVS cameras support CDP as the detection mechanism for executing the built-in ip-camera ASP macro. There are slight differences in the built-in ip-camera macro applied depending upon the platform (Catalyst access switch or Catalyst 4500) and upon whether the platform supports Enhanced ASP macros or regular ASP macros. The example in Table 7-7 shows the switchport configuration applied after a link-up event for a Catalyst access switch, both with regular ASP Macros and Enhanced ASP Macros. The configuration assumes the initial switchport configuration was a default configuration (meaning no configuration on the interface).

Table 7-7 Configuration Example 1—Switchport Configuration Resulting from the Built-in IP-Camera Macro

Regular ASP Macro	Enhanced ASP Macro
!	!
interface GigabitEthernet1/0/40	interface GigabitEthernet1/0/40
switchport access vlan 302[1]	switchport access vlan 302
switchport mode access	switchport mode access
switchport block unicast	switchport block unicast
switchport port-security	switchport port-security
mls qos trust dscp	srr-queue bandwidth share 1 30 35 5
macro description CISCO_IPVSC_EVENT	queue-set 2
spanning-tree portfast	priority-queue out
spanning-tree bpduguard enable	mls qos trust device ip-camera
!	mls qos trust dscp
	macro description CISCO_IPVSC_EVENT
	auto qos video ip-camera
	spanning-tree portfast
	spanning-tree bpduguard enable
	!

1. Access VLAN set by **macro auto device ip-camera ACCESS_VLAN=VLAN302** global configuration command.

Brief explanations of the commands are shown in Table 7-8.

Table 7-8 Summary of ASP Commands

Command	Description
switchport access vlan 302	Configures the switchport as a static access port using the access VLAN specified through the following manually configured global command: **macro auto device ip-camera ACCESS_VLAN=302**
switchport mode access	The port is set to access unconditionally and operates as a nontrunking, single VLAN interface that sends and receives nonencapsulated (non-tagged) frames.
switchport block unicast	By default, all traffic with unknown MAC addresses is sent to all ports. This command blocks unicast packets with unknown MAC addresses received by this port from being sent to other ports on the switch. This feature is designed to address the cam table overflow vulnerability, in which the cam table overflows and packets are sent out all ports.
switchport port-security	Enables port security on the interface. Defaults to one secure MAC address. Defaults to set the port in error-disable state upon a security violation. SNMP Trap and Syslog message are also sent.

auto qos video ip-camera	Automatically configures QoS on the port to support a Cisco IPVS camera. Causes the following interface level commands to be added:
	`srr-queue bandwidth share 1 30 35 5`
	`queue-set 2`
	`priority-queue out`
	`mls qos trust device ip-camera`
	`mls qos trust dscp`
	Causes global configuration changes to the switch configuration to occur as well.

Table 7-8 Summary of ASP Commands

`srr-queue bandwidth share 1 30 35 5`	Sets ratio by which the shaped round robin (SRR) scheduler services each of the four egress queues (Q1 through Q4 respectively) of the interface. Bandwidth is shared, meaning that if sufficient bandwidth exists, each queue can exceed its allocated ratio. Note that the priority-queue out command overrides the bandwidth ratio for Q1.
`queue-set 2`	Maps the port to the 2nd queue set within the switch. Catalyst 3560, 3750, and 2960 Series switches support two queue sets.
`priority-queue out`	Enables egress priority queuing. Automatically nullifies the srr-queue bandwidth share ratio for queue 1 since the priority queue is always serviced first (unlimited bandwidth).
`mls qos trust device ip-camera`	Enables the QoS trust boundary if CDP packets are detected indicating the connection of a IP surveillance camera to the interface.
`mls qos trust dscp`	Classify an ingress packet by using the packet's DSCP value.
`macro description CISCO_IPVSC_EVENT`	Description indicating which built-in macro has been applied to the interface, in this case the built in ip-camera macro.
`spanning-tree portfast`	When the Port Fast feature is enabled, the interface changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes.
`spanning-tree bpduguard enable`	Puts the interface in the error-disabled state when it receives a bridge protocol data unit (BPDU). This should not occur on a port configured for access mode.

The main difference between the Enhanced ASP macro and the regular ASP macro is that the Enhanced ASP macro includes the **auto qos video ip-camera** interface-level command. AutoQoS has been extended in IOS version 12.2(55)SE on the Catalyst access switches to support video devices as well as VoIP. Among other things, the **auto qos video ip-camera** command causes DSCP markings from the device to be trusted when the switchport detects CDP from the attached Cisco IPVS camera. On Catalyst access switches, the **auto qos video ip-camera** command also causes changes to the queue-sets, which globally affect the switch configuration. These global changes—which result from the AutoQoS commands within ASP macros—are not reversed when the anti-macro is run, returning the interface to its default configuration. Instead the global configuration changes remain within the running configuration of the switch. The network administrator may need to manually access the switch in order to save these changes in the running configuration into the startup configuration. Note also that minor disruptions to switch processing may occur the first time the queue-sets are modified. However, this occurs only when the first switchport configured for Enhanced ASP macros detects an IPVS camera. Subsequent switchports which detect an IPVS camera do not cause further changes to the queue-sets, since they have already been modified. For further discussion of the effects of AutoQoS video, see the Medianet Campus QoS Design 4.0 document at:
http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND_40/QoSCampus_40.html

Note Cisco recommends a DSCP setting of CS5 for IPVS cameras. However this is not currently the default value which ships in the firmware. The network administrator may have to manually change the DSCP value to CS5 within the IPVS cameras.

Cisco Digital Media Players (DMPs)

Cisco 4310G DMPs running revision 5.2.2 are the only DMPs which support CDP as the detection mechanism for executing the built-in media-player ASP macro. The CDP detection mechanism for DMPs only works for Enhanced ASP macros as well. However, the MAC address detection mechanism automatically works for Cisco 4305G, 4400G, and 4310G DMPs for both Enhanced ASP macros and regular ASP macros. Catalyst switches which support ASP macros have a built-in MAC address trigger which matches on the OUI values of 00-0F-44 or 00-23-AC, corresponding to Cisco DMPs.

The built-in media-player ASP macro is the same regardless of whether the platform supports Enhanced ASP macros or regular ASP macros. The example in Table 7-9 shows the switchport configuration applied after a link-up event for a Catalyst access switch. The configuration assumes the initial switchport configuration was a default configuration (meaning no configuration on the interface).

Table 7-9 Configuration Example 2—Switchport Configuration Resulting from the Built-in Media-Player Macro

Regular and/or Enhanced ASP Macro
!
interface GigabitEthernet2/0/8
switchport access vlan 282[1]
switchport mode access
switchport block unicast
switchport port-security
priority-queue out
mls qos trust dscp
macro description CISCO_DMP_EVENT
spanning-tree portfast
spanning-tree bpduguard enable
!

1. Access VLAN set by **macro auto device media-player ACCESS_VLAN=VLAN282** global configuration command.

Brief explanations of the commands are shown in Table 7-10.

Table 7-10 Summary of ASP Commands

Command	Description
`switchport access vlan 282`	Configures the switchport as a static access port using the access VLAN specified through the following manually configured global command: **macro auto device media-player ACCESS_VLAN=282**
`switchport mode access`	The port is set to access unconditionally and operates as a nontrunking, single VLAN interface that sends and receives nonencapsulated (non-tagged) frames.
`switchport block unicast`	By default, all traffic with unknown MAC addresses is sent to all ports. This command blocks unicast packets with unknown MAC addresses received by this port from being sent to other ports on the switch. This feature is designed to address the cam table overflow vulnerability, in which the cam table overflows and packets are sent out all ports.

Table 7-10 Summary of ASP Commands

`switchport port-security`	Enables port security on the interface. Defaults to one secure MAC address. Defaults to set the port in error-disable state upon a security violation. SNMP Trap and Syslog message are also sent.
`priority-queue out`	Enables egress priority queuing. Automatically nullifies the srr-queue bandwidth share ratio for queue 1, since the priority queue is always serviced first (unlimited bandwidth).
`mls qos trust dscp`	Classify an ingress packet by using the packet's DSCP value.
`macro description CISCO_DMP_EVENT`	Description indicating which built-in macro has been applied to the interface, in this case the built-in media-player macro.
`spanning-tree portfast`	When the Port Fast feature is enabled, the interface changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes.
`spanning-tree bpduguard enable`	Puts the interface in the error-disabled state when it receives a bridge protocol data unit (BPDU). This should not occur on a port configured for access mode.

The network administrator should note that MAC-address triggers are executed only after a timeout of either CDP or LLDP triggers. The timeout value is roughly 65 seconds. In other words, when deploying DMPs which do not support CDP, or deploying DMPs on Catalyst switch platforms which do not support Enhanced ASP macros, the Catalyst switch listens for CDP or LLDP triggers for approximately one minute. After the timeout, the switch executes the built-in MAC-address trigger corresponding to the DMP.

It is also important for the network administrator to understand the order in which certain services start when devices such as DMPs boot up. When using dynamic IP addressing, CDP should be sent before any DHCP packets are sent. This is because the access VLAN is often passed into the ASP macro. A device which acquires an IP address before the ASP macro has run will acquire an IP address corresponding to the default VLAN (VLAN 1). When the ASP macro subsequently runs, the device is moved onto a different access VLAN. Therefore, the device will need to release the existing IP address and acquire a new IP address. Typically this is done when the device sees the line-protocol transitioned when the VLAN is changed on the switch port. The built-in macros do not transition the link upon VLAN reassignment. Failure to release and renew the IP address results in an unreachable device, since its IP address corresponds to the wrong VLAN. This issue also exists when using the built-in MAC address trigger to execute the built-in media-player ASP macro for DMPs.

Medianet Devices without Built-in ASP Macros

The following devices are not currently supported by built-in ASP macros.

Cisco TelePresence (CTS) Endpoints

Currently there are no built-in ASP macros for Cisco TelePresence (CTS) endpoints within the Catalyst switch software. CTS endpoints consist of one or more codecs and an associated IP phone. As of CTS software version 1.6(5), both the codec and the phone send CDP packets to the Catalyst switch with the phone bit enabled within the capabilities field of CDP packets. Catalyst switchports currently apply the built-in phone ASP macro for attached CTS endpoints, based on the CDP trigger from the combination

of the IP phone and codec, assuming the phone macro is enabled globally on the Catalyst switch. For customers who have both Cisco IP phones and CTS endpoints attached to the same Catalyst switch and who wish to use ASP macros, this is a likely scenario.

The application of the built-in phone ASP macro does not cause CTS endpoints to stop working, provided the network administrator has deployed the TelePresence endpoint to share the voice VLAN with IP phones. However, the configuration is not optimal or recommended for CTS endpoints. The application of the built-in phone ASP macro includes the interface-level **auto qos voip cisco-phone** command. This applies AutoQoS VoIP to both the global configuration of the Catalyst switch as well as the interface. The current AutoQoS VoIP configuration only identifies, marks, and polices EF and CS3 traffic from an IP phone accordingly. Since TelePresence is recommended to be configured to send traffic with a CS4 DSCP marking, the AutoQoS VoIP configuration does not address TelePresence traffic at all. However, the traffic from the TelePresence codec is still trusted at the ingress port. Therefore the TelePresence traffic still crosses the network with a CS4 marking.

A recommended work-around for this situation is to disable ASP macros via the **no macro auto processing** interface-level command for Catalyst switchports which support Cisco TelePresence endpoints. Either manually configure the switchports or use Static Smartports with the recommended configuration to support a CTS endpoint.

Note As of IOS version 12.2(55)SE, Catalyst access switches support AutoQos for CTS endpoints with the **auto qos video cts** command. For more information, see the *Medianet Campus QoS Design 4.0* guide: http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND_40/QoSCampus_40.html.

Other Video Conferencing Equipment

Cisco desktop video conferencing software which consists of a PC daisy chained off of a Cisco IP phone will exhibit similar characteristics as Cisco CTS endpoints when implementing ASP macros. The attached Cisco IP phone will result in the built-in phone ASP macro being executed. The resulting configuration may not be optimal for desktop video conferencing.

No built-in ASP macros currently exist for Cisco Tandberg video conferencing equipment at the time this document was written. Therefore, it is recommended to either manually configure the switchports or use Static Smartports with the recommended configuration to support Cisco Tandberg video conferencing equipment.

Overriding Built-in Macros

Generally the built-in ASP macros support the requirements of most customers while easing the deployment of medianet devices onto the network infrastructure. However, sometimes there may be reasons why a network administrator may wish to change the functionality of a built-in ASP macro. For example, with regular ASP macros, there is no ability to disable an individual built-in macro, such as the switch or router macros. Since these macros automatically configure the port as a trunk allowing all VLANS, there may be potential security issues with allowing them to run. The network administrator may desire to override the existing macro in such a manner that it is effectively disabled.

Alternatively, the network administrator may wish to only slightly modify the function of an existing built-in ASP macro. For example, as previously mentioned, the deployment of sticky macros in a dynamic environment causes Auto Smartports to be less effective due to the **switchport port-security**

interface-level command within both the built-in ip-camera and media-player macros. An overridden macro may be configured in order to modify or remove port-security for these devices if the network administrator desires to use sticky macros.

Built-in macros can be overridden by creating a new macro with the same name as an existing built-in macro. These overridden macros can be located in one of three places:

- Embedded within the switch configuration
- A standalone macro file within the switch flash
- A standalone macro file accessed remotely by the switch

The partial configuration example in Table 7-11 shows an overridden switch macro embedded within the configuration of a Catalyst access switch.

Table 7-11 Configuration Example 3—Overridden ASP Macro within the Switch Configuration

```
!
macro auto execute CISCO_SWITCH_EVENT {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface  $INTERFACE
                macro description $TRIGGER
                description ROGUE SWITCH
DETECTED - PORT ENABLED
                switchport mode access
                shutdown
            exit
        end
    else
        conf t
            interface  $INTERFACE
                no macro description
                description ROGUE SWITCH
DETECTED - PORT DISABLED
                no switchport mode access
            exit
        end
    fi
}
!
```

The overridden switch macro example above simply causes the interface to be put into a shutdown state when the switchport detects the presence of another switch via the CDP triggering mechanism.

The benefit of embedding an overridden macro directly within the switch configuration is the ability to view the macro directly from the configuration. The downside is that the network administrator may need to duplicate the same overridden macro on every switch which requires it. This can be both time consuming and error prone in large deployments, limiting the overall ability to scale Auto Smartports deployments.

The second method of overriding a built-in macro is to put the overridden macro in a file located within the flash memory of the switch. In order to override a built-in macro from a flash file, the network administrator needs to include the **macro auto execute** <*trigger name*> **remote** <*remote file location*> command within the global configuration of the switch. The example in Table 7-12 shows the command line added to a Catalyst access switch to override the built-in media-player ASP macro and the file contents of the overridden macro itself.

Table 7-12 Configuration Example 4—Overridden Macro within a Flash File on the Switch

Global Configuration Command

```
!
macro auto execute CISCO_DMP_EVENT remote flash:DMP_macro.txt
!
```

Contents of the Flash File Overriding the Built-in Macro

```
me-w-austin-3#more DMP_macro.txt
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface  $INTERFACE
                macro description $TRIGGER
                switchport access vlan $ACCESS_VLAN
                switchport mode access
                switchport block unicast
                mls qos trust dscp
                spanning-tree portfast
                spanning-tree bpduguard enable
                priority-queue out
            exit
        end
    fi
    if [[ $LINKUP -eq NO ]]; then
        conf t
            interface  $INTERFACE
                no macro description
                no switchport access vlan $ACCESS_VLAN
                no switchport block unicast
                no mls qos trust dscp
                no spanning-tree portfast
                no spanning-tree bpduguard enable
                no priority-queue out
                if [[ $AUTH_ENABLED -eq NO ]]; then
                    no switchport mode access
                fi
            exit
        end
    fi
```

The benefit of this method is that a single overridden macro file can be created centrally—perhaps on a management server—and copied to each switch which needs to override the built-in ASP macro. This can help reduce the administrative burden and potential for errors, increasing the scalability of Auto Smartports deployments.

The downside is that there is no method to validate that the overridden macro actually functions correctly when it is typed in a separate text file and subsequently downloaded to the Catalyst switch. It is recommended that the network administrator test any overridden macros—perhaps using a non-production lab or backup switch—before deploying them in order to avoid such errors. Errors in overridden macros will cause macro processing to immediately exit. This can result in un-deterministic results in the configuration of an interface, based upon where in the macro the error occurred. The network administrator should also note that currently no error or warning will be generated to the switch console or syslog when the macro exits due to an error.

A second downside is that there is no method to validate that the overridden macro is correct for the particular model of switch to which it is being downloaded. There are slight command differences between Catalyst access switches and Catalyst 4500 Series switches which could cause a macro written for the wrong switch model to execute incorrectly. This can again result in un-deterministic results in

the configuration of an interface, based upon where the command differences occurred. In order to avoid this potential issue, the network administrator may choose to include the Catalyst switch model within the file name of the overridden macro. This gives the network administrator a quick visual indication if the file being downloaded is correct for the switch model.

The third method of overriding a built-in ASP macro is to put the overridden macro in a file on a remote server and configure the switch to access the file when it needs to run the macro. In order to override a built-in macro from a file on a remote server, the network administrator needs to include the **macro auto execute** <*trigger name*> **remote** <*remote file location*> command again within the global configuration of the switch. However, this time the remote file location includes the protocol, network address or hostname, userid and password, and path to the file on the remote server. The example in Table 7-13 shows the command line added to a Catalyst access switch to override the built-in media-player ASP macro. The contents of the overridden macro itself are the same as that shown in Table 7-12.

Table 7-13 Configuration Example 5—Example Configuration for Overriding a Macro via a File on a Remote Server

Global Configuration
!
macro auto execute CISCO_DMP_EVENT remote ftp://admin:cisco@10.16.133.2/DMP_macro.txt
!

The switch is capable of using the following protocols for download of remote macros: FTP, HTTP, HTTPS, RCP, SCP, and TFTP. In production networks, it is recommended to implement a secure protocol, such as SCP or HTTPS.

The benefit to this approach is that the overridden ASP macro files can again be managed centrally on a management server. This further eases the administrative burden of not having to manually copy the macro file to each switch which requires it. This is particularly useful when changing the behavior of an overridden macro that is already deployed on switches throughout the network infrastructure.

The network administrator should note that the overridden ASP macro file is downloaded to the Catalyst switch every time a link-up or link down event occurs. The switch does not cache the macro file; it simply requests the file every time there is a link-up or link-down event on the port. Testing did not investigate any potential scalability implications for processing on the switch, since multiple ports on the same switch may simultaneously request a file for download in order to apply the macro, particularly in scenarios where the switch has just been reloaded. This method also has potential scalability implications for the remote server, since it may have to process multiple simultaneous downloads from multiple ports on a single switch and from multiple switches throughout the network.

A downside to this method is that if the remote server is unavailable, the overridden ASP macro will not be run and the device will end up with a default configuration on the Catalyst switch. In many cases, the device will not function since it may be on the wrong VLAN. If the interface is already up and configured via the overridden ASP macro when the medianet device is removed, the configuration will remain on the Catalyst switchport if the remote server is unavailable. This is because the anti-macro will not be run to clean-up the switchport configuration. If another device is subsequently connected to the switchport, the resulting switchport configuration could be somewhat un-deterministic. This situation should be avoided. The remote server could be unavailable either due to a network error or a server error. Therefore, it is recommended that the network administrator implement both network-level redundancy as well as server-level redundancy in order to ensure the availability of the remote ASP macro when utilizing this method. Obviously the built-in router Auto Smartport macro should not be used to configure the interface that would route to the FTP server. Extra care will also be needed if the account password is to be changed on a reoccurring basis due to a security policy.

Finally, as with the previous method, there is no mechanism to validate the overridden ASP macro has no errors or is the correct macro for the model of switch to which it will be automatically downloaded. It is again recommended that the network administrator test any overridden macros— perhaps using a non-production lab or backup switch—before making them available for automatic download in order to avoid such errors.

Note CiscoWorks LMS is targeted to add support for managing Auto Smartports macros in an upcoming release. Future updates to this document may include details about LMS as it relates to ASP macros.

Macro-of-Last-Resort

As highlighted in Table 7-2, Enhanced ASP macros support a feature known as the macro-of-last-resort (also referred to as the LAST_RESORT macro). The macro-of-last-resort is a built-in ASP macro which is run if no other trigger event is seen and therefore no other ASP macro (built-in or user-defined) is run. Without the use of the macro-of-last-resort, devices such as end-user PCs—which typically will not trigger any ASP macros—may end up with a default switchport configuration, depending on whether the custom macro has been overridden. This may not be the desired switchport configuration, particularly if the network administrator uses a VLAN other than VLAN1 for the normal data VLAN. The custom macro is discussed in Custom Macro.

Note The use of a VLAN other than the default (VLAN 1) for the data VLAN is consistent with Cisco SAFE security guidelines.

The built-in macro-of-last-resort is enabled on Catalyst switches which support Enhanced ASP macros via the following global configuration command:

```
macro auto global control trigger last-resort
```

The macro-of-last-resort can also be enabled per interface with the following interface-level command:

```
macro auto control trigger last-resort
```

The only parameter which can be passed into the built-in ASP macro-of-last-resort is the access VLAN. This can be passed using the following global switch configuration command:

```
macro auto execute CISCO_LAST_RESORT_EVENT built-in CISCO_LAST_RESORT_SMARTPORT
ACCESS_VLAN=<vlan>
```

For example, in order to set the access VLAN to VLAN100 for the macro-of-last-resort, the network administrator would configure the following global switch command:

```
macro auto execute CISCO_LAST_RESORT_EVENT built-in CISCO_LAST_RESORT_SMARTPORT
ACCESS_VLAN=100
```

The example in Table 7-14 shows the switchport macro-of-last-resort configuration applied after a link-up event for a Catalyst access switch. The configuration assumes the initial switchport configuration was a default configuration (meaning no configuration on the interface).

Table 7-14 Configuration Example 6—Switchport Configuration Resulting from the Built-in Macro-of-Last-Resort

```
!
interface GigabitEthernet1/0/7
 switchport access vlan 100[1]
 switchport mode access
 load-interval 60
 macro description CISCO_LAST_RESORT_EVENT
 spanning-tree portfast
 spanning-tree bpdufilter enable
!
```

1. Access VLAN set by **macro auto execute CISCO_LAST_RESORT_EVENT built-in CISCO_LAST_RESORT_SMARTPORT ACCESS_VLAN=100** global configuration command.

Brief explanations of the commands are shown in Table 7-15.

Table 7-15 Summary of ASP Commands

Command	Description
`switchport access vlan 100`	Configures the switchport as a static access port using the access VLAN specified through the following manually configured global command: **macro auto execute CISCO_LAST_RESORT_EVENT built-in CISCO_LAST_RESORT_SMARTPORT ACCESS_VLAN=100**
`switchport mode access`	The port is set to access unconditionally and operates as a nontrunking, single VLAN interface that sends and receives nonencapsulated (non-tagged) frames.
`load-interval 60`	Sets the interval over which interface statistics are collected to average over 60 seconds.
`macro description CISCO_LAST_RESORT_EVENT`	Description indicating which built-in macro has been applied to the interface, in this case the built in last-resort macro.
`spanning-tree portfast`	When the Port Fast feature is enabled, the interface changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes.
`spanning-tree bpduguard enable`	Puts the interface in the error-disabled state when it receives a bridge protocol data unit (BPDU). This should not occur on a port configured for access mode.

When the device is removed from the interface, the anti-macro will return the switchport to a default interface configuration. The macro-of-last-resort can also be overridden. This allows the network administrator to implement a completely custom default switchport configuration for devices which do not match any built-in or user-defined ASP macros.

Since the macro-of-last resort executes if no other triggering events are seen, including MAC-address trigger events, there could be a delay of over one minute between the time the switchport interface becomes active and the execution of the macro-of-last-resort. During this time period, the device will be active on the default VLAN (VLAN 1)—unless it was left on a different VLAN by the custom macro.

An end-user PC which uses DHCP could obtain an IP address before the switch moves the VLAN configuration to that specified by the macro-of-last-resort if the default VLAN contains a DHCP server. When the switchport is subsequently moved to the new VLAN, the end-user PC should release and renew the DHCP lease based on the line protocol transition of the switch. The network administrator may wish to test the PC hardware and operating systems deployed within his/her network to ensure they function properly before deploying Enhanced Auto Smartports. An alternative is to simply not provision a DHCP server on the default VLAN. Typically most DHCP clients will try to DISCOVER a server for more than the macro-of-last-resort timeout, however it is possible the end-user PC will timeout when attempting to obtain a DHCP address. In this case, the end-user may need to manually re-activate DHCP again after the macro-of-last-resort has moved the PC to the correct VLAN in order to obtain an IP address corresponding to the correct VLAN.

Note Testing with the macro-of-last resort did not include the use of 802.1x/MAB on the end-user PC. Therefore, no design guidance around the interaction of 802.1x/MAB and the macro-of-last-resort is provided in this document at this time.

Custom Macro

The custom macro is a built-in Enhanced ASP macro which is automatically executed upon an interface link down event. The following example output from the **show shell function CISCO_CUSTOM_AUTOSMARTPORT** exec-level command shows the built-in custom macro.

```
me-w-austin-3>show shell function CISCO_CUSTOM_AUTOSMARTPORT
function CISCO_CUSTOM_AUTOSMARTPORT () {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface  $INTERFACE
            exit
        end
    fi
    if [[ $LINKUP -eq NO ]]; then
        conf t
            interface $INTERFACE
            exit
        end
    fi
}
```

By default, the custom macro does nothing at all unless it is overridden by the network administrator. The network administrator may choose to override the custom macro to provide functionality, such as a VLAN configuration other than the default VLAN (VLAN 1) to a port when there is no device connected to it. The following two examples illustrate possible uses of the custom macro.

Example Scenario #1

The network administrator has pre-configured all unused ports to be on the data VLAN, instead of the default VLAN. If a DMP device is connected to a switchport configured for Enhanced ASP macros, it will be recognized as a DMP and moved into the VLAN specified by the network administrator through the built-media-player Enhanced ASP macro. For example, the port may be moved to a DMP VLAN. If the DMP device is then removed, the DMP anti-macro executes, removing the switchport from the DMP VLAN (which places it into the default VLAN). The custom macro will then execute, moving the switchport into the VLAN specified within the overridden custom macro. This may correspond to the data VLAN again. If a normal PC device is subsequently placed onto the same switchport, the PC will immediately come up within the data VLAN. It will remain there since it will not trigger any other

built-in Enhanced ASP macros. This scenario assumes the macro-of-last-resort has not been enabled. Therefore, in this example, the custom macro provides the network administrator an alternative method of placing devices which do not trigger any built-in Enhanced ASP macros (such as normal PCs) onto a VLAN other than the default VLAN.

The advantage of the custom macro in this scenario is that the device does not have to wait until the macro-of-last resort is executed to be moved into the correct VLAN. This may help minimize issues regarding PCs acquiring an incorrect DHCP addresses because they were moved to another VLAN by the macro-of-last-resort. However, the network administrator should be careful of medianet devices such as DMPs and IPVS cameras accidently getting the wrong IP addresses, since they initially come up within the data VLAN as well. Finally, the network administrator may have to manually pre-configure all unused switchports be within the data VLAN initially. The custom macro will not be run until another macro has been run on the port and the device has subsequently been removed.

Example Scenario #2

The network administrator has pre-configured all unused ports to be on an unused or isolated VLAN, instead of the default VLAN. If a DMP device is connected to a switchport configured for Enhanced ASP macros, it will be recognized as a DMP and moved into the VLAN specified by the network administrator through the built-media-player Enhanced ASP macro. For example, the port may be moved to a DMP VLAN. If the DMP device is then removed, the DMP anti-macro executes, removing the switchport from the DMP VLAN (which places it into the default VLAN). The custom macro will then execute, moving the switchport into the VLAN specified within the overridden custom macro. This may correspond to the unused or isolated VLAN in this scenario. If a normal PC device is subsequently placed onto the same switchport, the PC will immediately come up within the unused or isolated VLAN. If the macro-of-last-resort has been enabled, it will trigger, moving the device into another VLAN, such as the normal data VLAN. If the PC is then removed from the switchport, its anti-macro will execute, removing switchport from the data VLAN (which places it into the default VLAN). Then the custom macro will again execute, moving the switchport back into the unused or isolated VLAN.

In this scenario, the custom macro provides the network administrator a method of placing unused ports into an unused or isolated VLAN—which is more consistent with Cisco SAFE guidelines. If the unused or isolated VLAN has no DHCP server, then devices will not accidently get the wrong IP address before they are subsequently moved into their correct VLANs by the Enhanced ASP macros. However, PCs may have to wait longer until the macro-of-last-resort executes in order to become active on the network. Finally, the network administrator may have to manually pre-configure all unused switchports to be within the unused or isolated VLAN initially. The custom macro will not be run until another macro has been run on the port and the device has subsequently been removed.

Overridden Custom Macro

Table 7-16 shows an example of an overridden custom macro.

Table 7-16 Configuration Example 7—Overridden Custom Macro Within the Switch Configuration

```
!
macro auto execute CISCO_CUSTOM_EVENT
ACCESS_VLAN=402  {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface  $INTERFACE
            exit
        end
    fi
    if [[ $LINKUP -eq NO ]]; then
        conf t
            interface $INTERFACE
            switchport access vlan
$ACCESS_VLAN
            exit
        end
    fi
}
!
```

The overridden macro example simply places the switchport into VLAN 402 when it goes into a link down state. Note that the VLAN can either be hardcoded into the overridden macro or passed in via a variable declaration as shown in this example.

Security Considerations

CDP and LLDP are not considered to be secure protocols. They do not authenticate neighbors, nor make any attempt to conceal information via encryption. The only difficulty in crafting a CDP packet is that the checksum is calculated with a non-standard algorithm. Even this has been reversed engineered and published in public forums. As a result, CDP and LLDP offer an attractive vulnerability to users with mal-intent. For example, by simply sending in a CDP packet with the "S" bit (otherwise referred to as the switch bit) set in the capabilities TLV, the switch can be tricked into configuring a trunk port that will pass all VLANs and accept 802.1d BPDUs from the hacker. This could be used in man-in-the-middle (MIM) attacks on any VLAN in the switch. Below is an example of a CDP spoofing device that has set the switch bit. Notice that the platform is set to a DMP. An obvious give-away in this example is the host name, CDP Tool1, which was chosen to be obvious. Normally the host name would have been selected to make the device appear to be a legitimate DMP device.

```
me-w-austin-3#sh cdp neigh g1/0/39
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
CDP Tool1        Gig 1/0/39        30                 S    DMP 4305G eth0
```

Because the switch policy ignores the platform, this field can be used to make the entry appear to be legitimate while still tricking the switch to configure a trunk, as shown below.

```
!
interface GigabitEthernet1/0/39
 location civic-location-id 1 port-location
  floor 2
  room Broken_Spoke
```

```
 switchport mode trunk
 srr-queue bandwidth share 1 30 35 5
 queue-set 2
 priority-queue out
 mls qos trust cos
 macro description CISCO_SWITCH_EVENT
 macro auto port sticky
 auto qos trust
end
```

Fortunately with Enhanced ASP macros, the user is allowed to disable specific scripts. The recommendation is to only enable host type macros. Switches, routers, and access-points are rarely attached to a network in a dynamic fashion. Therefore, ASP macros corresponding to these devices should be disabled where possible.

As discussed previously, ASP allows the use of remote servers to provide macros. Secure sessions such as HTTPS should be used. If the MIM above is used in conjunction with an unsecured remote configuration, the network administrator has released full control of the device to the hacker.

Authenticating Medianet Devices

Device authentication has been an ongoing concern since the early days of wireless access. The topic is the subject of several books. A common approach is to enable 802.1x. This authentication method employs a supplicant located on the client device. If a device does not have a supplicant, as is the case with many printers, then the device can be allowed to bypass authentication based on its MAC address. This is known as MAC-Authentication-Bypass or MAB. As the name implies, this is not authentication, but a controlled way to bypass that requirement. Currently all ASP medianet devices must use MAB if device authentication is in use, since these devices do not support an 802.1x supplicant. With MAB the client's MAC address is passed to a RADIUS server. The server authenticates the devices based solely on its MAC address and can pass back policy information to the switch. Administrators should recognize that MAC addresses are not privileged information. A user can assign a locally-administered MAC address. Another key point is that MAB and ASP can happen independently of one another. A device may use MAB to get through authentication and then use CDP to trigger and ASP event. Security policy must also consider each independently. A user could hijack the MAC address from a clientless IP phone, then spoof CDP to trigger the SWITCH_EVENT macro. The risk is greatly reduced by following the recommendation to turn off ASP support for static devices such as switches and routers.

MAB with ASP can be configured as shown in the example in Table 7-17.

Table 7-17 Configuration Example 7—MAB with ASP

```
!
interface GigabitEthernet1/0/7
 description me-austin-c1040 (new_home)
 switchport mode access
 authentication event fail action authorize vlan 301
 authentication event no-response action authorize vlan 301
 authentication host-mode multi-domain
 authentication order dot1x mab
 authentication priority dot1x mab
 authentication port-control auto
 mab eap
end
!
```

If the built-in macros have been overridden by the user, care should be taken to ensure they do not interfere with the MAB configuration. This includes the anti-macro section that removes the applied macro.

CDP Fallback

This feature is used to provide an alternate trigger method when RADIUS does not return an event trigger. If this feature is not used, then an authenticated device that does not include a RADIUS trigger will execute the LAST_RESORT Macro if enabled. The network administrator may want to disable CDP fallback to prevent CDP spoofing tools from hijacking a MAC-Address known to be authenticated by MAB. This does not prevent the device from being authenticated, but it does prevent the device from assuming capabilities beyond those of the true MAC address. While there is an incremental security gain from this approach, there are service availability concerns if the RADIUS server does not provide a recognized trigger event. As noted previously, this has not been fully validated at the time of this writing.

Guest VLANs and LAST_RESORT Macro

With MAB enabled, the MAC address is sent to a RADIUS server for authentication. If the MAC address is unknown, MAB may direct the interface to join a Guest VLAN if the switch is configured to do so. This is independent of any action invoked via ASP. As a result, there could be inconsistencies in VLAN assignment between MAB and ASP. In this case, the MAB result takes precedence, as shown in Table 7-18.

Table 7-18 Precedence Between ASP and MAB

ASP recognized device	MAB Authenticated	Result
NO	NO	GUEST VLAN
NO	YES	LAST RESORT VLAN
YES	NO	GUEST VLAN
YES	YES	ASP ASSIGNED VLAN

The LAST RESORT VLAN corresponds to the access VLAN configured for the macro-of-last-resort, assuming the network administrator has enabled its use. The final VLAN assignment may not be the initial VLAN that was configured on the interface when line protocol initially came up. The timing is important. If the client's DHCP stack successfully obtains an IP address prior to the final VLAN assignment, the client may become unreachable. In this case, the client should be reconfigured to use static addressing. In most situations, MAB and ASP will complete the VLAN assignment prior to DHCP completion. One area of concern arises when CDP packets are not sent by the client. In this case, a MAC-address-based ASP will wait 65 second prior to executing a trigger. The client may have completed DHCP and will not be aware that a VLAN change has occurred. If MAB was also enabled, an unknown client will be in the placed in the GUEST_VLAN. VLAN reassignments as a result of ASP are transparent to the client's stack. This is also the case if a VLAN is manually changed on an enabled interface. Manual VLAN changes are accompanied by shutting and no shutting the interface. ASP does not do this for the built-in system macros.

Verifying the VLAN Assignment on an Interface

The best method to determine if an ASP has executed correctly is to validate the interface configuration. The macro description can be used to determine which macro has executed. The administrator should also review the configuration settings put in place by the macro. However, when MAB and ASP are running concurrently, the configuration cannot be used to determine the state of the interface. Instead the **show interface switchport** command may be used. The following example shows that the interface has executed the LAST_RESORT macro and therefore could be in VLAN 100 or VLAN 301, depending on the authentication result.

```
!
interface GigabitEthernet1/0/39
 description Overridden Macro-of-Last-Resort (Port Active)
 switchport access vlan 100
 switchport mode access
 authentication event fail action authorize vlan 301
 authentication event no-response action authorize vlan 301
 authentication port-control auto
 mab eap
 macro description CISCO_LAST_RESORT_EVENT
end
```

The **show** command below indicates that the device was not authenticated and it currently is in VLAN 301:

```
me-w-austin-3#sh int g1/0/39 swi
Name: Gi1/0/39
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 301 (VLAN0301)
! <additional lines omitted >
!
```

ASP with Multiple Attached CDP Devices

In some situations, there may be two CDP devices on a single interface. A common case is seen with Cisco TelePresence. In this situation both the CTS codec and IP phone appear as CDP neighbors on a single port of the switch. There are other situations that could also arise, such as a downstream hub with multiple LLDP or CDP devices, although in a practical sense this is quite uncommon. Another case may be a CDP spoofing tool. In any case, the script will make an initial determination based on the first trigger selected. Once the macro has configured the interface with a macro description, no further configuration changes will be made. If a user incorrectly removes the macro description, the interface will be reconfigured on the next trigger event. Because only the first trigger is significant, there may be some concern as to which script will run when multiple devices are present. In the case of the CTS, the phone script will be triggered regardless of whether the codec or phone presents its CDP packet first. This is because the phone bit is set in both the CTS codec and its associated IP phone in the capabilities TLV and the script will override any host trigger with a phone trigger. Even if the codec presents a CDP packet first, the phone trigger will execute.

If a hub is attached to an ASP port, several built-in macro scripts include port security that would likely err_disable the switch interface. In the academic situation where two different classes of CDP or LLDP devices may be attached to a hub, where port security is not being used and where each different type is a known ASP class device, then the first CDP packet seen would set the port configuration. Subsequent

CDP packets from adjacent devices will not cause the interface to change configurations. Hubs are rarely seen in today's networks. Even small four port devices are typically switching. Medianet devices would not typically be found attached via a hub, therefore the LAST_RESORT macro would likely be applied to any switchport supporting Enhanced ASP.

Deployment Considerations

When deploying Auto Smartports, the network administrator does not necessarily have to enable ASP macros across the entire switch. Instead the network administrator may wish to consider initially enabling ASP macros only on a range of interfaces. This method of incremental deployment may facilitate a smoother transition from the paradigm of manual configuration to that of auto configuration. For example, if the network administrator is only beginning the transition toward a medianet by deploying digital signage and IP video surveillance cameras over a converged IP infrastructure, he/she may choose to set aside the first several ports on access switches for DMPs and/or IP cameras. ASP macro processing would only need to be enabled for these "reserved" switchports. All end-user PCs and uplinks to other switches, routers, or access points would still be done via either Static Smartports or manual configuration. This methodology works best if the medianet devices (DMPs and IPVS cameras) are placed on a separate VLAN or VLANs from the data VLAN. The macro-of-last resort can be used to simply "quarantine" the medianet device to an unused VLAN if the built-in ASP macro failed to trigger. With this method, the network administrator can still gain the administrative advantages of auto configuration for what may be hundreds or thousands of medianet specific devices, such as IP cameras and DMPs across the network infrastructure. Normal change control mechanisms can be maintained for uplink ports and infrastructure devices such as routers, switches, and access points, since they do not utilize ASP macros in the initial phased rollout of auto configuration. The network administrator can disable the unused built-in ASP macros for these devices, as well as unused detection mechanisms. As the network administrator becomes more familiar with the use of ASP macros, the deployment can then be extended to include other devices as well as infrastructure connections if desired.

Location Services

Location Services is another feature of the Medianet Service Interface (MSI) that provides the ability for the Catalyst switch to send location information to a device via CDP or LLDP-MED. Future benefits of a medianet device learning its location from the network infrastructure may be the ability to customize the configuration of the device based upon its location or the ability to automatically display content based on its learned location.

Catalyst access switches support the ability to pass either civic location information or emergency location information (ELIN) to devices via CDP or LLDP-MED in IOS revision 12.2(55)SE. Catalyst 4500 Series switches support the ability to pass either civic location information or ELIN to devices via LLDP-MED only in IOS revision 12.2(54)SG. This document will only address civic location information.

Civic location is discussed under various IETF proposed standards, including RFCs 4119 and 5139. Civic location information can be configured on a global basis (for location elements which pertain to the entire switch) and on an interface-level basis (for location elements which pertain to the specific switchport). The configuration example in Table 7-19 shows and example of civic location information configured both globally and on a switchport.

Table 7-19 Configuration Example 8—Example Civic Location Configuration

```
!
location civic-location identifier 1
 building 2
 city Austin
 country US
 postal-code 33301
 primary-road-name Research_Blvd
 state Texas
 number 12515
!
!
interface GigabitEthernet1/0/39
 location civic-location-id 1 port-location
  floor 2
  room Broken_Spoke
!
```

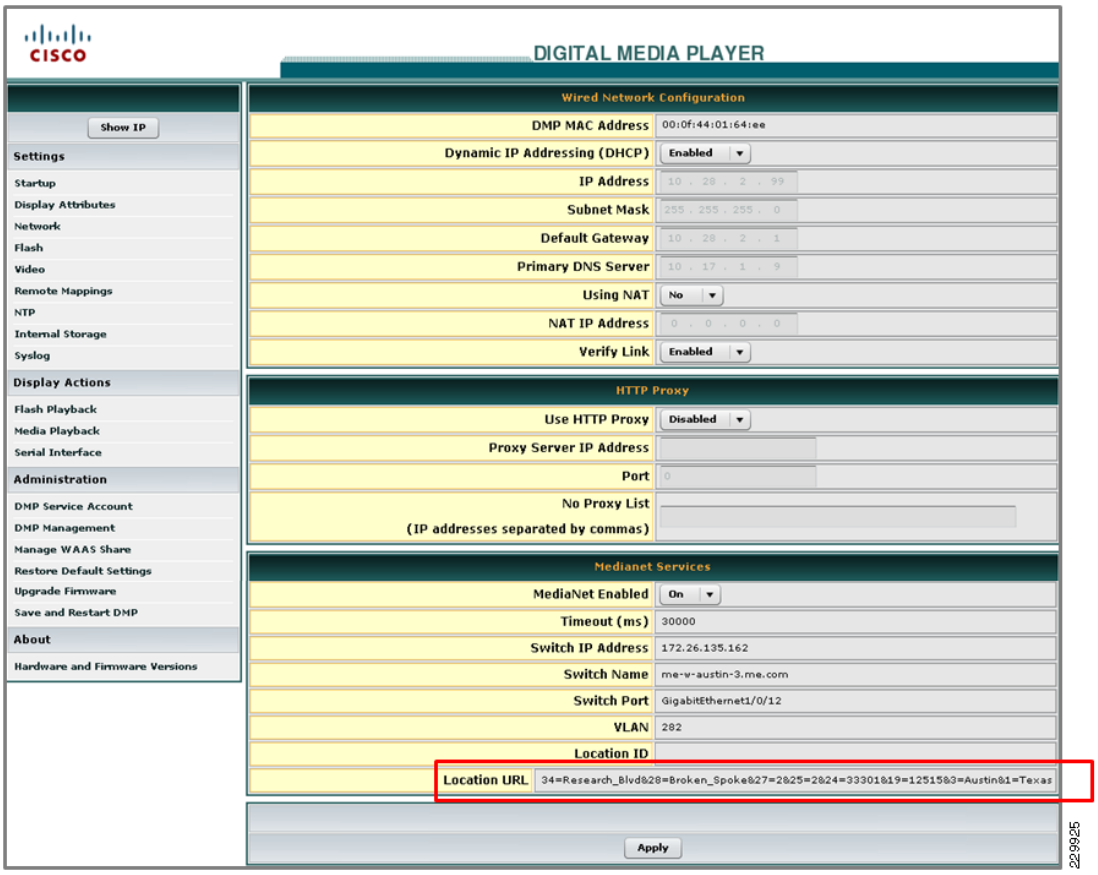
The location of the switch—identified via the **location civic-location identifier 1** global command—corresponds to the following hypothetical address: 12515 Research_Blvd, building 2, Austin, Texas, US, 33301. The location of the switchport extends the civic location via the location **civic-location identifier 1 port-location** interface-level command to identify the device as being in the Broken_Spoke room on floor two. The use of civic location in this manner does require the network administrator to manually keep accurate records as to which switchports are wired to which rooms within the facility.

Note There are limitations regarding the total size of the location information which can be sent via CDP and LLDP. The network administrator should keep the location information size under 255 bytes.

The network administrator can enable or disable the sending of location information via CDP on all ports for the entire switch with the **cdp tlv location** or **no cdp tlv location** global commands. For individual switchports, the network administrator can enable or disable the sending of location information via CDP with the **cdp tlv location** or **no cdp tlv location** interface-level commands. The network administrator can enable or disable the sending of location information via LLDP-MED for individual switchports with the **lldp-med-tlv-select location** or **no lldp-med-tlv-select location** interface-level commands.

Currently the only medianet specific device which supports Location Services is the Cisco 4310G DMP running revision 5.2.2 software. Figure 7-3 shows the configuration of a Cisco 4310G DMP with location information which has been passed to the DMP via CDP from an attached Catalyst 2960-S switch.

Figure 7-3 GUI Interface of a Cisco 4310G DMP Showing Location Passed via CDP



Note CiscoWorks LMS is targeted to add support for Location Services in an upcoming release. Future updates to this document may include details around LMS as it relates to Location Services.

Summary

Auto configuration can help facilitate the transition of the network infrastructure towards a medianet by easing the administrative burden of having to manually configure multiple switchports for devices such as digital media players (DMPs) and IP video surveillance (IPVS) cameras. The Auto Smartports (ASP) feature allows the network infrastructure to automatically detect a medianet device attached to a Cisco Catalyst switch via the Cisco Medianet Service Interface (MSI) and configure the switchport to support that particular device. Additionally, Location Services allow the switchport to send civic location information to the medianet device. Such location information may be used in the future for functionality such as customizing the configuration of the device based upon its location or automatically displaying content based upon the learned location of the medianet device.

References

- Medianet Campus QoS Design 4.0: http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND_40/QoSCampus_40.html
- Auto Smartports Configuration Guide, Release 12.2(55)SE: http://www.cisco.com/en/US/docs/switches/lan/auto_smartports/12.2_55_se/configuration/guide/asp_cg.html
- Configuring LLDP, LLDP-MED, and Wired Location Service: http://www.cisco.com/en/US/docs/switches/lan/catalyst3750x_3560x/software/release/12.2_55_se/configuration/guide/swlldp.html