Managing FlexConfigs

FlexConfig policies allow you to configure device commands that are not otherwise supported by Security Manager. By using Flexconfigs, you can extend Security Manager's control over a device configuration and take advantage of new device features before upgrading the product.

FlexConfig policies are made up of FlexConfig objects. These objects are essentially subroutines that can include scripting language commands, device commands, and variables. You can configure an object to be processed prior to applying the Security Manager configuration to a device, or you can have it processed after the configuration. Security Manager processes your objects in the order you specify so that you can create objects whose processing depends on the processing of another object. A FlexConfig policy object's contents can range from a single simple command string to elaborate CLI command structures that incorporate scripting and variables.

Note You can configure Cisco Security Manager either to deploy FlexConfigs only once after creation or modification of a FlexConfig, or to deploy FlexConfigs with each deployment. By default, Cisco Security Manager deploys FlexConfigs one time. If you have FlexConfigs that need to be deployed with each deployment, disable the "Deploy only new or modified Flexconfigs" setting on the **Tools > Security Manager Administration > Deployment page**. After changing this setting, you have to manage one-time FlexConfigs by deleting them after they have been deployed. For more information, see Deployment Page, page 11-13.

Understanding policies and objects is central to understanding and using FlexConfig policy objects. For more information on how Security Manager defines and uses polices, see Chapter 5, "Managing Policies" and for information on how Security Manager defines and uses objects, see Chapter 6, "Managing Policy Objects".

Note With "Deploy only new or modified Flexconfigs" setting on **Tools > Security Manager Administration > Deployment page** enabled, if you have an activity open, with changes, and when you attempt to deploy FlexConfigs, Cisco Security Manager considers the FlexConfig changes specific to that activity alone and not those of other activities. On the other hand, if all activities are submitted and no activity is open, then Security Manager considers the FlexConfig changes specific to the lastly submitted activity that was submitted with changes. Therefore, if you need FlexConfig changes for an activity to reflect during deployment, then ensure the changes are done in a single activity, submitted, and deployed.

The following topics describe FlexConfig policies and policy objects and how to use them:

- Understanding FlexConfig Policies and Policy Objects, page 7-2
- Configuring FlexConfig Policies and Policy Objects, page 7-25

Understanding FlexConfig Policies and Policy Objects

FlexConfig policy objects are used in FlexConfig policies. They allow you to configure device features that are not otherwise supported by Security Manager, or to otherwise fine-tune your device configurations. These policy objects include device configuration commands, variables, and optionally, scripting language instructions to control processing. FlexConfig objects are essentially programming routines to add content to the device configurations that Security Manager generates.

You can create FlexConfig policy objects from scratch or you can duplicate one of the objects that are included with Security Manager.

FlexConfig policies are simply an ordered list of FlexConfig policy objects. Your objects are processed in the order that you specify.

The following topics help you understand FlexConfig policy objects and by extension, FlexConfig policies. For more information about policy objects in general, see Chapter 6, "Managing Policy Objects".

- Using CLI Commands in FlexConfig Policy Objects, page 7-2
- Using Scripting Language Instructions, page 7-3
- Understanding FlexConfig Object Variables, page 7-5
- Predefined FlexConfig Policy Objects, page 7-19

Using CLI Commands in FlexConfig Policy Objects

The configuration commands that you enter into the FlexConfig Editor are actual CLI commands used to configure devices, such as PIX Firewalls and Cisco IOS Routers. You can include CLI commands that are not supported in Security Manager. You are responsible for knowing and implementing the command according to the proper syntax for the device type. See the command reference for the particular operating system for more information.

When you create a Flexconfig policy object, you determine whether the commands and instructions should be added to the beginning or end of the configuration that is generated from regular Security Manager policies:

- Prepended objects—FlexConfig objects that are processed at the beginning of the configurations. If Security Manager policies configure any of the same commands included in the object, the prepended commands are replaced when configuration files are deployed.

- Appended objects—FlexConfig objects that are processed at the end of the configurations, after all other commands in the configuration file and before the **write mem** command.

  If the appended commands are already configured on the device, the device generates an error when you try to add them again. To resolve this, two workarounds are available:

  – Enter the command that removes the configuration in question as an appended command. For example, if the command is **xyz**, enter the following two lines:

    ```
    no xyz
    xyz
    ```
  – Change the setting that controls the action that the device will take to "warn." This is set under **Tools > Security Administration > Deployment**.

The setting change will affect the behavior of devices for all commands being deployed, not just those designated as appended commands.

> **Note** If you are deploying to a device, you should remove most appended commands after the initial deployment. This is especially true for object groups, where any unbound object group is replaced in the Ending Command section during command generation, then re-sent each time the configuration is deployed to a device. The device displays an error because the firewall device shows that the object group already exists. If you are deploying to a file or AUS, the appended commands should remain.

Using Scripting Language Instructions

You can use scripting language instructions in a FlexConfig policy object to control how the commands in the object are processed. Scripting language instructions are a subset of commands supported in the Velocity Template Engine, a Java-based scripting language that supports looping, if/else statements, and variables.

Security Manager supports all Velocity Template Engine commands except the **include** and **parse** commands. For information about additional supported commands supported, see the Velocity Template Engine documentation.

The following topics provide examples of the most commonly used functions:

Scripting Language Example 1: Looping

A plain old telephone service (POTS) dial peer enables incoming calls to be received by a telephony device by associating a telephone number to a voice port. The following example enables caller ID for a set of POTS dial peers.

Object Body

```
#foreach ($peer_id in ["2", "3", "4"])
    dial-peer voice $peer_id pots
    caller-id
#end
```

CLI Output

```
dial-peer voice 2 pots
caller-id

dial-peer voice 3 pots
caller-id

dial-peer voice 4 pots
caller-id
```

Scripting Language Example 2: Looping with Two-Dimensional Arrays

In this example, a set of phone numbers is associated to voice ports so that incoming calls can be received at a router.

Object Body

```
#foreach ($phone in [ [ "2000", "15105552000", "1/0/0" ], [ "2100",
"15105552100", "1/0/1" ], [ "2200", "15105552200", "1/0/2" ] ] )
    dial-peer voice $phone.get(0) pots
 destination-pattern $phone.get(1)
 port $phone.get(2)
#end
```

CLI Output

```
dial-peer voice 2000 pots
destination-pattern 15105552000
port 1/0/0

dial-peer voice 2100 pots
destination-pattern 15105552100
port 1/0/1

dial-peer voice 2200 pots
destination-pattern 15105552200
port 1/0/2
```

Example 3: Looping with If/Else Statements

In this example, a set of phone numbers is associated to voice ports so that incoming calls can be received at a router. In addition, another set of phone numbers is associated to IP addresses to enable Voice Over IP outgoing calls from the router.

Object Body

```
#foreach ( $phone in [ [ "2000", "15105552000", "1/0/0", "" ],
[ "2100", "15105552100", "1/0/1", "" ],
[ "2200", "15105552200", "", "ipv4:150.50.55.55"]
[ "2300", "15105552300", "", "ipv4:150.50.55.55"] ] )
    dial-peer voice $phone.get(0) pots
     destination-pattern $phone.get(1)
 #if ( $phone.get(2) == "" )
     session target $phone.get(3)
 #else
     port $phone.get(2)
 #end

#end
```

CLI Output

```
dial-peer voice 2000 pots
    destination-pattern 15105552000
 port 1/0/0

dial-peer voice 2100 pots
    destination-pattern 15105552100
 port 1/0/1
```

```
dial-peer voice 2200 pots
    destination-pattern 15105552000
 session target ipv4:150.50.55.55

dial-peer voice 2300 pots
    destination-pattern 15105552300
 session target ipv4:150.50.55.55
```

Understanding FlexConfig Object Variables

Variables in FlexConfig policy objects start with the $ character. For example, in the following line, $inside is a variable:

```
interface $inside
```

There are three types of variables you can use in a FlexConfig policy object:

- Policy object variables—Static variables that reference a specific property. For example, Text objects are a type of policy object variable. They are a name and value pair, and the value can be a single string, a list of strings, or a table of strings. Their flexibility allows you to enter any type of textual data to be referenced and acted upon by any policy object.

  There are three ways to add policy object variables to a FlexConfig policy object. First, move the cursor to the desired location, and then:

  – Right-click and select **Create Text Object**. This command opens a dialog box where you can create a simple single-value text object and assign it a value. When you click OK, the variable is added to the object, and it is added to the list of defined Text objects in the Policy Object Manager window so that you can use it in other objects or edit its definition. For an example of creating simple text variables, see Example of FlexConfig Policy Object Variables, page 7-6.

  – Right-click and select a policy object type from the **Insert Policy Object** sub-menu. These commands open a selector dialog box where you can select the specific policy object that contains the variable that you want to insert. After selecting the policy object, you are presented with the Property Selector dialog box, where you choose the specific property of the object that you want to use and optionally change the name of the variable associated with the property.

    By using this technique, you can add a property from an existing policy object when you know that the property has the value that you want to use. For example, if you want to insert a variable that specifies the RADIUS protocol from the AAA Server Group policy object named RADIUS, you would right-click, select **Insert Policy Object > AAA Server Group**, select RADIUS in the AAA Server Group Selector dialog box, click OK, and then select Protocol in the Object Property field on the AAA Server Group Property Selector dialog box and click OK. The $protocol variable is inserted at the cursor, and the value for the property as defined in the selected object is added to the variables list.

  – Type in a variable name. If you type in a variable, you cannot assign it a value until you click OK on the Add or Edit FlexConfig dialog box. You will be prompted that a variable is undefined, and given the opportunity to define its value. In the FlexConfig Undefined Variable dialog box, you can select the object type of the policy object that contains the desired value, which will prompt you to select the specific policy object and variable. This is essentially identical to the process for inserting policy object variables described above. The technique you use is a matter of personal preference; the end result is the same.

- System variables—Dynamic variables that reference a value during deployment when the configuration is generated. The values are obtained from either the target device or policies configured for the target device. You can declare system variables to be optional in FlexConfig policy objects, which means that the variables do not need to be assigned a value for it to be deployed to the device.

  To insert a system variable into a FlexConfig policy object, move the cursor to the desired location, right-click, and select the variable from the **Insert System Variable** sub-menus. For a description of the available system variables, see FlexConfig System Variables, page 7-7.

- Local Variables—Variables that are local in the looping and assignment derivatives (the **for each** and **set** statements). Local variables get their values directly from the Velocity Template Engine. There is no need to supply values for the local variables.

  To insert a local variable, simply type it in. When you click OK on the Add or Edit FlexConfig dialog box, you will be asked if you want to define the undefined variable. You can click No, or if you click Yes to define other variables, you can leave the object type of the local variable as Undefined.

Example of FlexConfig Policy Object Variables

Using CLI commands and variables, you can create a FlexConfig policy object to name the inside interface and crypto map on a Cisco router:

```
interface $inside
crypto map $mapname
```

The following example shows how to create a FlexConfig policy object that adds these commands and configures the value of $inside as **serial0** and $mapname as **my_crypto**.

When you add the FlexConfig policy object to a device, and the configuration is generated, the following output is created:

```
interface serial0
crypto map my_crypto
```

Step 1 Select **Manage > Policy Objects** to open the Policy Object Manager (see Policy Object Manager, page 6-4).

Step 2 Select **FlexConfigs** from the table of contents. The table in the right pane lists the existing FlexConfig objects.

Step 3 Right-click in the table and select **New Object**. The Add FlexConfig dialog box appears (see Add or Edit FlexConfig Dialog Box, page 7-30).

Step 4 Enter a name and optionally a description for the object.

> **Tip** You can also enter a group name. Groups help you find FlexConfig objects if you create a lot of them. Either type in a group name, or select an existing one from the drop-down list.

Step 5 Keep **Appended** for Type so that the commands are added at the end of the device configuration.

Step 6 Create the content of the object:

  a. Click in the FlexConfig edit box (the large white box) and type in **interface** followed by a space.

  b. Right-click and select **Create Text Object**.

  c. In the Create Text Object dialog box, enter **inside** as the name and **serial0** as the value. Click **OK** to add the variable.

    d. Press Enter to move to the next line and type **crypto map** followed by a space.

    e. Right-click and select **Create Text Object**.

    f. In the Create Text Object dialog box, enter **mapname** as the name and **my_crypto** as the value. Click **OK** to add the variable.

Step 7 Click the **Validate FlexConfig** icon button above the edit box to check the integrity and deployability of the object. If any errors are identified, fix them.

Step 8 Click **OK** to save the policy object. You can now add the object to a device's local or shared FlexConfig policy.

FlexConfig System Variables

System variables reference values during deployment when commands are generated. Security Manager provides a set of defined system variables for you to use in defining FlexConfig policy objects. The values come from the policies you create for the target devices. The values for these variables are required unless otherwise noted. For information about these variables, see the following tables:

- Device system variables—Table 7-1 on page 7-7. For more information about discovering or configuring devices to obtain values for these variables, see Chapter 3, "Managing the Device Inventory".

- Firewall system variables—Table 7-2 on page 7-9. For more information about firewall policies, see Chapter 46, "Managing Firewall Devices" and Chapter 12, "Introduction to Firewall Services".

- Router platform system variables—Table 7-3 on page 7-13. For more information about router policies, see Chapter 61, "Managing Routers".

- VPN system variables—Table 7-4 on page 7-14. For more information about VPN policies, see Chapter 25, "Managing Site-to-Site VPNs: The Basics".

- Remote access system variables—Table 7-5 on page 7-19. For more information about remote access policies, see Chapter 30, "Managing Remote Access VPNs: The Basics".

Table 7-1   *Device System Variables (Applying to All Device Types)*

Name	Dimension	Description
SYS_DEVICE_IDENTITY	0	Unique device identity for devices managed by a Configuration Engine or Auto Update Server (AUS) as defined on the Tools > Device Properties > General tab. There must be a device identity for devices managed by these servers.
SYS_DOMAIN_NAME	0	DNS domain name as defined on the Tools > Device Properties > General tab. This is not necessarily the same value that is defined in the Platform > Device Admin > Hostname policy.
SYS_FW_OS_MODE	0	Operating system mode of the FWSM or ASA device as defined on the Tools > Device Properties > General tab. Possible values are ROUTER (routed mode), TRANSPARENT, or NOT_APPLICABLE.

Table 7-1   Device System Variables (Applying to All Device Types) (continued)

Name	Dimension	Description
SYS_FW_OS_MULTI	0	Whether the FWSM or ASA is running in single- or multiple-context mode as defined on the Tools > Device Properties > General tab. Possible values are SINGLE, MULTI, or NOT_APPLICABLE.
SYS_HOSTNAME	0	Device hostname as defined on the Tools > Device Properties > General tab. This is not necessarily the same value that is defined in the Platform > Device Admin > Hostname policy.
SYS_IMAGE_NAME	0	Device image name as defined on the Tools > Device Properties > General tab.
SYS_INTERFACE_IP_LIST	1	IP addresses and masks of the interfaces configured in the Interfaces policy. The IP address and mask are in the x.x.x.x/nn format (for example, 10.20.1.2/24). If there are no interfaces defined on the device, no list is returned. Each element in SYS_INTERFACE_NAME_LIST and SYS_INTERFACE_IP_LIST share the same index for the interface. For example, if element 3 in SYS_INTERFACE_NAME_LIST is for Ethernet1, element 3 in SYS_INTERFACE_IP_LIST is the IP address for Ethernet1. If Ethernet1 has no IP address, element 3 in the SYS_INTERFACE_IP_LIST is empty. This variable is optional.
SYS_INTERFACE_NAME_LIST	1	Names of the interfaces on the device configured in the Interfaces policy. If no interfaces are defined on the device, no list is returned. See the explanation above for SYS_INTERFACE_IP_LIST for additional information. This variable is optional.
SYS_MANAGEMENT_IP	0	Management IP address of the device as defined on the Tools > Device Properties > General tab.
SYS_MDF_TYPE	0	Cisco MDF (MetaData Framework) device type, which indicates the device model. This value is displayed on the Tools > Device Properties > General tab, and is determined when you add the device to Security Manager.

Table 7-1   Device System Variables (Applying to All Device Types) (continued)

Name	Dimension	Description
SYS_OS_RUNNING_VERSION	0	Software version of the operating system running on the device as displayed on the Tools > Device Properties > General tab. For example, 12.1, 12.2S, and so on, on an IOS platform. This value is determined when you discover policies from the device.
SYS_OS_TARGET_VERSION	0	Operating system version to be used when generating the device configuration as defined on the Tools > Device Properties > General tab.
SYS_OS_TYPE	0	Operating system for the device as defined on the Tools > Device Properties > General tab. Possible values are IOS, PIX, ASA, FWSM, IPS. You configure this value when you add the device to Security Manager.
SYS_SYS_OID	0	System object ID (SysObjId) of the device, which is determined when you add the device to Security Manager.

Table 7-2   Firewall System Variables

Name	Dimension	Description
SYS_FPM_INPUT_SP	1	FPM policy map names applied on the interface corresponding to the entry in the SYS_FPM_INTERFACE list in the "in" direction. This data is not configured in Security Manager. It is obtained from a router's running configuration and is used by the IOS_FPM FlexConfig.
SYS_FPM_INTERFACE	1	Interface names. This data is not configured in Security Manager. It is obtained from a router's running configuration and is used by the IOS_FPM FlexConfig.
SYS_FPM_OUTPUT_SP	1	FPM policy map names applied on the interface corresponding to the entry in the SYS_FPM_INTERFACE list in the "out" direction. This data is not configured in Security Manager. It is obtained from a router's running configuration and is used by the IOS_FPM FlexConfig.

Table 7-2   Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_ACL_IN_NAME	1	Names of ACLs applied to interfaces for traffic filtering in the inbound direction. Each element has a one-to-one correspondence with the SYS_INTERFACE_NAME_LIST variable for Cisco IOS routers, PIX Firewalls, Firewall Service Modules, and ASA devices.
		Configure firewall access rules to generate values for this variable.
SYS_FW_ACL_OUT_NAME	1	Names of ACLs applied to interfaces for traffic filtering in the outbound direction. Each element of this array has a one-to-one correspondence with SYS_INTERFACE_NAME_LIST variable for Cisco IOS routers, PIX Firewalls, Firewall Service Modules, and ASA devices.
		Configure Access Rules policies to generate values for this variable.
SYS_FW_BRIDGE_INTERFACE_NAMES	1	Names of bridge interfaces.
		This variable applies only to IOS transparent firewalls.
		Configure the Firewall > Transparent Rules policies to generate values for this variable.
SYS_FW_ETHERTYPERULE_ACL_NAMES	1	Names of ethertype access-lists applied to interfaces for traffic filtering coming in or going out. Each element of this array has a one-to-one correspondence with the elements in the SYS_FW_ETHERTYPERULE_INTERFACE_NAMES and SYS_FW_ETHERTYPERULE_DIRECTION_NAMES variables.
		Configure Firewall > Transparent Rules policies to generate values for this variable.
SYS_FW_ETHERTYPERULE_DIRECTION_NAMES	1	Direction that ethertype access-lists are applied. The value is either "in" or "out." Each element has a one-to-one correspondence with the elements in the SYS_FW_ETHERTYPERULE_ACL_NAMES and SYS_FW_ETHERTYPERULE_INTERFACE_NAMES variables.
		Configure Firewall > Transparent Rules policies to generate values for this variable.

Table 7-2   ***Firewall System Variables (continued)***

Name	Dimension	Description
SYS_FW_ETHERTYPERULE_ INTERFACE_NAMES	1	Interface names to which ethertype access-lists are applied. Each element has a one-to-one correspondence with the SYS_FW_ETHERTYPERULE_ACL_NAMES and SYS_FW_ETHERTYPERULE_DIRECTION_NAMES variables. Configure Firewall > Transparent Rules policies to generate values for this variable.
SYS_FW_INSPECT_IN_NAME	1	Names of Inspect Rules applied to Cisco IOS router interfaces in the inbound direction. Each element of this array has a one-to-one correspondence with the SYS_INTERFACE_NAME_LIST variable for Cisco IOS routers. Configure Inspection Rules policies to generate values for this variable. This variable is optional.
SYS_FW_INSPECT_OUT_NAME	1	Names of Inspect rules applied to Cisco IOS router interfaces in the outbound direction. Each element of this array has a one-to-one correspondence with the SYS_INTERFACE_NAME_LIST variable for Cisco IOS routers. Configure Inspection Rules policies as values for this variable. This variable is optional.
SYS_FW_INTERFACE_HARDWARE_ ID_ LIST	1	Hardware IDs for the device. Configure Interface policies on the device to generate values for this variable. This variable is optional.
SYS_FW_INTERFACE_NETWORK_LIST	1	Interface networks on the device. Configure Interface policies on the device to generate values for this variable.
SYS_FW_INTERFACE_SECURITY_ LEVEL_LIST	1	Interface security levels on the device. Configure Interface policies on the device to generate values for this variable.
SYS_FW_INTERFACE_STATE_LIST	1	Interface states on the device. Configure Interface policies on the device to generate values for this variable.
SYS_FW_INTERFACE_VLAN_ID_LIST	0	VLAN IDs on the device. Configure Interface policies on the device to generate values for this variable.

Table 7-2   Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_IPV6_ACL_IN_NAME	1	A list of all IPv6 ACLs in the In direction on the device. Configure IPv6 Access Rules policies in the In direction on the device to generate values for this variable.
SYS_FW_IPV6_ACL_OUT_NAME	1	A list of all IPv6 ACLs in the Out direction on the device. Configure IPv6 Access Rules policies in the Out direction on the device to generate values for this variable.
SYS_FW_MPCRULE_TRAFFICFLOW_ TUNNELGROUPNAME	1	Names of tunnel groups specified in Traffic Flow objects. Traffic Flow objects configure **class-map** commands on PIX/ASA devices, and the names of the tunnel groups listed in Traffic Flow objects populate this variable. This variable is used by the ASA_define_traffic_flow_tunnel_group FlexConfig object to create tunnel groups on PIX/ASA devices. This variable is optional.
SYS_FW_MULTICAST_PIM_ ACCEPT_ REG_ROUTEMAP	0	Route-map name used in the **pim accept-register route-map** command. Enter a name for the route-map (Platform > Multicast > PIM > Request Filter), then configure its features using FlexConfig to generate values for this variable. This variable is optional.
SYS_FW_NAT0_ACL_NAMES	1	Names of ACLs used in the **nat** *interface_name* **0 access-list acl_name** command. This variable is optional.
SYS_FW_OSPF_PROCESS_ID_LIST	1	IDs for OSPF routing processes globally configured on PIX Firewalls, Firewall Service Modules, and ASA devices. Configure Platform > Routing > OSPF policies to generate values for this variable.
SYS_FW_OSPF_REDISTRIBUTION_ ROUTE_MAP_LIST	1	Names for the route maps to apply to the OSPF redistribute commands configured on PIX Firewalls, Firewall Service Modules, and ASA devices. Configure Platform > Routing > OSPF policies to generate values for this variable.

Table 7-2   Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_POLICY_NAT_ACL_NAMES	1	Names of ACLs used in the **policy nat** commands (nat commands with non-0 pool id). Configure NAT (NAT > Translation Rules > Policy NAT) to generate values for this variable. This variable applies to only PIX 6.3(3) and higher, PIX/ASA 7.x, 8.0(x), 8.1(x), and 8.2(x), and FWSM devices. This variable does not apply to Cisco IOS routers. This variable is optional.
SYS_FW_POLICY_STATIC_ACL_NAMES	1	Names of ACLs used in the **policy static** commands that include access lists. Configure NAT 0 (NAT > Translation Rules > Policy NAT) to generate values for this variable. The variable contains the access-list names used by the **nat-0**, **policy nat**, and **policy static** commands. This variable applies to only PIX 6.3(3) and higher, PIX/ASA 7.x, 8.0(x), 8.1(x), and 8.2(x), and FWSM devices. This variable does not apply to Cisco IOS routers. This variable is optional.

Table 7-3   Router Platform System Variables

Name	Dimension	Description
SYS_ROUTER_BGP_AS_NUMBERS_LIST	1	Autonomous system (AS) number of the border gateway protocol (BGP) and exterior gateway protocol (EGP) on the device. Configure Router Platform > Routing > BGP policies to generate values for this variable. This variable is optional.
SYS_ROUTER_EIGRP_AS_NUMBERS_LIST	1	Autonomous system (AS) numbers of the different enhanced Internet gateway routing protocols (EIGRP) and interior gateway protocols (IGP) on the device. Configure Router Platform > Routing > EIGRP policies to generate values for this variable. This variable is optional.

Table 7-3   Router Platform System Variables (continued)

Name	Dimension	Description
SYS_ROUTER_OSPF_PROCESS_ IDS_LIST	1	Open shortest path first (OSPF) interior gateway protocol (IGP) process numbers on the device. Configure Router Platform > Routing > OSPF Process policies to generate values for this variable. This variable is optional.
SYS_ROUTER_QOS_CLASS_ MAP_LIST	1	Names of QoS class maps on the device. Configure Quality of Service policies to generate values for this variable. This variable is optional.
SYS_ROUTER_QOS_POLICY _MAP_LIST	1	Names of the QoS policy-maps on the device. Configure Quality of Service policies to generate values for this variable. This variable is optional.

Table 7-4   VPN System Variables

Name	Dimension	Description
Topology		
Variables related to the VPN in which a device participates. Configure VPNs to generate values for these variables.		
SYS_VPN_TOPOLOGY	1	Virtual private network (VPN) topology type. Possible values are HUB_AND_SPOKE, POINT_TO_POINT, and FULL_MESH.
SYS_VPN_TOPOLOGY_NAME	1	Name of the VPN topology in which the device participates.
SYS_VPN_TOPOLOGY_ROLE	1	Details about the role of the device in the VPN. Possible values are PEER, HUB, and SPOKE.
Devices		
Variables related to devices in the VPN in which a device participates. Configure VPNs to generate values for these variables.		
SYS_VPN_HOST_NAME	1	Device hostname.
SYS_VPN_LOCAL_PREFIXES	2	Interface and network IP addresses of protected networks.
SYS_VPN_PRIVATE_INTERFACES	2	Private interface names.
SYS_VPN_PRIVATE_TUNNEL_ENDPT_IP	1	Interface tunnel IP address.
SYS_VPN_PUBLIC_INTERFACES	2	Public interface names.

Table 7-4   VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_TUNNEL_ENDPT_ INTERFACE_IP	1	IP address of the VPN endpoint. In IPSec, the endpoint is the VPN interface; in GRE, it is the tunnel source.
SYS_VPN_TUNNEL_ENDPT_ INTERFACE_NAME	1	Name of the VPN endpoint. In IPSec, the endpoint is the VPN interface; in GRE, it is the tunnel source.
SYS_VPN_VPNSM_PUBLIC_ IFC	2	Export port names for Catalyst 6000 series switches.
Remote Peers		
Variables related to remote peers in which a device participates. Configure VPNs to generate values for these variables.		
SYS_VPN_REM_PEER_BAK_ LOGICAL_PRIVATE_IP	3	Interface tunnel IP addresses of remote peers of failover hubs. This value is used in DMVPN for next hop resolution protocol (NHRP).
SYS_VPN_REM_PEER_BAK_ PREFIX	3	Protected networks (interface and network IP addresses) of remote peers of failover hubs.
SYS_VPN_REM_PEER_BAK_ PUBLIC_IP	3	Public interface names of remote peers of failover hubs.
SYS_VPN_REM_PEER_BAK_ TUNNEL_SRC	3	IP address of the VPN endpoint of remote peers. In IPSec, the endpoint is the VPN interface; in GRE, it is the tunnel source.
SYS_VPN_REM_PEER_DEVI CE_NAME	2	Device hostnames of remote peers.
SYS_VPN_REM_PEER_LOGI CAL_ PRIVATE_IP	2	Interface tunnel IP addresses of remote peers. This value is used in DMVPN for next hop resolution protocol (NHRP).
SYS_VPN_REM_PEER_PREF IX	3	Protected networks (interface and network IP addresses) of remote peers.
SYS_VPN_REM_PEER_PRIV ATE_IP	2	Private interface names of remote peers.
SYS_VPN_REM_PEER_PUBL IC_IP	2	Public interface names of remote peers.
SYS_VPN_REM_PEER_TUN NEL_SRC	2	Tunnel sources (if included in the interface tunnel of remote peers).
IPSec Proposal		
Variables related to IPSec Proposal policies. For more information, see Configuring IPsec Proposals in Site-to-Site VPNs, page 26-22 and Configuring High Availability in Your VPN Topology, page 25-52.		
Configure the IPSec Proposal policy to generate values for these variables.		
SYS_VPN_CRYPTO_MAP_T YPE	1	Crypto map type. Possible values are STATIC and DYNAMIC.

Table 7-4   VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_DYNAMIC_CRYPTO_NAME	1	Dynamic crypto map name.
SYS_VPN_DYNAMIC_CRYPTO_NUM	1	Dynamic crypto map number.
SYS_VPN_STATIC_CRYPTO_NAME	1	Static crypto map name.
SYS_VPN_STATIC_CRYPTO_NAME_BAK	1	Static crypto map name of failover hubs.
SYS_VPN_STATIC_CRYPTO_NUM	2	Static crypto map number.
SYS_VPN_STATIC_CRYPTO_NUM_BAK	2	Static crypto map number of failover hubs.

Preshared Keys

Variables related to Preshared Key and IKE Proposal policies. For more information, see Configuring IKEv1 Preshared Key Policies, page 26-48.

Name	Dimension	Description
SYS_VPN_IKE_AUTHENTICATION_MODE	1	Authentication method of the IKE policy. Possible values are pre-share, rsa-sig, rsa-encr, dsa-sig. Configure an IKE Proposal policy to generate values for this variable.
SYS_VPN_IKE_PRIORITY	1	Priority number of the IKE policy Configure an IKE Proposal policy to generate values for this variable.
SYS_VPN_NEGOTIATION_MODE	1	Negotiation method. Possible values are MAIN_ADDRESS, MAIN_HOST, and AGGRESSIVE. Configure a Preshared Key policy to generate values for this variable.

GRE Modes

Variables related to GRE Modes policies. For more information, see Understanding the GRE Modes Page, page 27-1.

Name	Dimension	Description
SYS_VPN_BAK_TUNNEL_IFC	2	Interface tunnel number of remote peers of failover hubs, for example, tunnel0. Configure VPNs to generate values for this variable.
SYS_VPN_SIGP_PROCESS_NUMBER	1	Process number of the interior gateway protocol (IGP). Configure GRE Modes policies to generate values for this variable.

Table 7-4   VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_SIGP_ROUTING_PROTOCOL	1	Type of secured interior gateway protocol (IGP) used. Possible values are STATIC, OSPF, EIGRP, RIPV2, BGP, and ODR. Configure GRE Modes policies to generate values for this variable.
SYS_VPN_SPOKE_TO_SPOKE_ CONN	1	Whether DMVPN is configured for spoke-to-spoke connectivity. Possible values are true or false. Configure GRE Modes policies to generate values for this variable.
SYS_VPN_TUNNEL_IFC	2	Interface tunnel number of remote peers, for example, tunnel0. Configure VPNs to generate values for this variable.

VRF

Variables related to virtual routing and forwarding (VRF). For more information, see Configuring VRF Aware IPsec Settings, page 25-48.

Configure VPN VRF settings to generate values for these variables.

Name	Dimension	Description
SYS_VPN_VRF_AREA_ID	1	Area ID numbers if the OSPF process number was chosen.
SYS_VPN_VRF_MPLS_INTERFACE_IP	1	Multiprotocol label switching (MPLS) interface IP addresses.
SYS_VPN_VRF_MPLS_INTERFACE_NAME	1	Multiprotocol label switching (MPLS) interface names.
SYS_VPN_VRF_NAME	1	VRF names.
SYS_VPN_VRF_PROCESS_NUMBER	1	Interior gateway protocol (IGP) process numbers.
SYS_VPN_VRF_RD	1	RD values.
SYS_VPN_VRF_ROUTING_PROTOCOL	1	Interior gateway protocol (IGP) values. IGP is used for routing the IPSec aggregator toward the Provider Edge (PE)/Multiprotocol Label Switching (MPLS) network. Possible values are STATIC, OSPF, EIGRP, RIPV2, and BGP.
SYS_VPN_VRF_SOLUTION	1	Virtual routing and forwarding (VRF) solution. Possible values are 1BOX and 2BOX.

CA

Variables related to certificate authority policies. For more information, see Configuring IKEv1 Public Key Infrastructure Policies in Site-to-Site VPNs, page 26-54.

Table 7-4   VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_CA_NAME	2	Certificate authority (CA) names. Configure PKI policies to generate values for this variable.
EZVPN		
Variables related to EZVPN. For more information, see Understanding Easy VPN, page 28-1.		
SYS_VPN_EZVPN_GROUP_NAME	2	User group names. Configure User Group policies to generate values for this variable.
Dial Backup		
Variables related to dial backup configurations. For more information, see Configuring Dial Backup, page 25-40.		
SYS_VPN_RTR_WATCH	1	The rtr/watch number. Configure dial backup to generate values for this variable.
GETVPN		
Variables related to Group Encrypted Transport (GET) VPN. For more information, see Understanding Group Encrypted Transport (GET) VPNs, page 29-2.		
SYS_GDOI_GROUP_NAME	1	Name of the Group Domain of Interpretation (GDOI) group. Configure the Group Encryption policy to generate values for this variable (Manage > Site-to-Site VPNs > Group Encryption Policy > Group Settings).
SYS_GM_GET_ENABLED_INTF_NAME	1	VPN-enabled outside interface to the provider edge (PE). Traffic originating or terminating on this interface is evaluated for encryption or decryption, as appropriate. Configure group members to generate values for this variable (Manage > Site-to-Site VPNs > Group Members).
SYS_IPSEC_PROFILE_NAME	1	Name of the profile that defines the parameters to be used for IPsec encryption between two group members. Configure the Group Encryption policy to generate values for this variable (Manage > Site-to-Site VPNs > Group Encryption Policy > Security Associations).

Table 7-4   VPN System Variables (continued)

Name	Dimension	Description
SYS_KS_REG_INTERFACE	0	Interface on the key server assigned to handle group domain of interpretation (GDOI) registrations. If no registration interface is specified, GDOI registrations can occur on any interface. Configure key servers to generate values for this variable (Manage > Site-to-Site VPNs > Key Servers).

Table 7-5   Remote Access System Variables

Name	Dimension	Description
SYS_ASA_RA_TUNNEL_GROUP_NAME	2	Tunnel group name for ASA devices.
SYS_ASA_RA_USER_GROUP_NAME	2	Name of the ASA user group.
SYS_EZVPN_RA_DYNAMIC_CRYPTO_ MAP_NAME	1	Dynamic Crypto map name for EZVPN.
SYS_EZVPN_RA_DYNAMIC_CRYPTO_ MAP_SEQ_NUM	1	Dynamic Crypto map number for EZVPN.
SYS_EZVPN_RA_PUBLIC_INTERFACE_PIX	2	External interface names for EZVPN for PIX firewall and ASA devices only.
SYS_EZVPN_RA_STATIC_CRYPTO_ MAP_NAME	1	Static crypto map names for EZVPN.
SYS_EZVPN_RA_STATIC_CRYPTO_ MAP_SEQ_NUM	1	Static crypto map numbers for EZVPN.
SYS_IOS_RA_CA_NAME	1	Certificate authority (CA) names for Cisco IOS devices.
SYS_IOS_RA_PUBLIC_INTERFACE	1	External interface names for Cisco IOS devices.
SYS_IOS_RA_USER_GROUP	1	User group names for Cisco IOS devices.
SYS_IOS_RA_VRF_NAME	1	Virtual routing and forwarding (VRF) names for Cisco IOS devices.

Predefined FlexConfig Policy Objects

Security Manager provides predefined FlexConfig policy objects for you to use. These policy objects have predefined commands and scripting.

Predefined FlexConfig policy objects are read-only objects. To edit these predefined FlexConfig policy objects, duplicate the desired object, make changes to the copy, and save it with a new name. This way, the original predefined FlexConfigs remain unchanged. For lists of these predefined policy objects and further information on each, see the following tables:

Table 7-6   Predefined ASA FlexConfig Policy Objects

Name	Description
ASA_add_ACEs	Adds an access control entry (ACE) to all access control lists on the device.
ASA_add_EtherType_ACL_remark	Loops through a list of ethertype access-list names and adds ACEs or remarks to them. The ethertype access list is the same as Transparent Rules for Firewalls in Security Manager. The remarks set by the CLI in this FlexConfig will be shown in the description field of a transparent rule.
ASA_add_IPv6_ACEs	Loops through a list of IPv6 access lists and adds a **deny ip any any log** entry to the end of the ACL.
ASA_command_alias	Creates a command alias named "save" for the **copy running-config** and **copy startup_config** commands.
ASA_copy_image	Copies an image package from a TFTP server to flash.
ASA_csd_image	Provides an ASA Cisco Secure Desktop image. It copies the CSD image from /CSCOpx/tftpboot/*device-hostname* on the Cisco Security Manager server to the device, then configures the CSD image path. Make sure you fill out the device's hostname in Device Properties. If the image name is different than the default, you can override it in Device Properties > Policy Object Overrides > Text Objects > AsaCsdImageName. Unassign this FlexConfig from the device after the image has been copied and configured.
ASA_define_traffic_flow_tunnel _group	Defines site-to-site VPN tunnel groups listed in the SYS_FW_MPCRULE_TRAFFICFLOW_TUNNELGROUPNAME system variable. This variable is populated with tunnel group names defined in Traffic Flow objects.
ASA_established	Permits return access for outbound connections through the security appliance. This command works with an original connection that is outbound from a network and protected by the security appliance and a return connection that is inbound between the same two devices on an external host. Uses the **established** command to specify the destination port that is used for connection lookups, which gives you more control over the command and supports protocols where the destination port is known, but the source port is unknown. The **permitto** and **permitfrom** keywords define the return inbound connection.
ASA_FTP_mode_passive	Sets the FTP mode to passive.

Table 7-6   Predefined ASA FlexConfig Policy Objects (continued)

Name	Description
ASA_generate_route_map	Generates a route map to be used by the **pim accept-register route-map** command configured under Platform > Multicast > PIM > Request Filter. Security Manager exports the route-map name used in the **pim** command so that you can configure it as desired.
ASA_IP_audit	Uses the **ip-audit** command to provide the following: • Sets the default actions (alarm, drop, reset) for packets that match an attack signature. • Sets the default actions (alarm, drop, reset) for packets that match an informational signature. • Creates a named audit policy that identifies the actions to take (alarm, drop, reset) when a packet matches a defined attack signature or an informational signature. • Disables a signature for an audit policy. • Assigns an audit policy to an interface.
ASA_MGCP	Identifies a specific map for defining the parameters for Media Gateway Control Protocol (MGCP) inspection.
ASA_no_router_Id	Removes the router ID for each OSPF process.
ASA_no_shut_Intf	Loops through and enables all interfaces on a device.
ASA_privilege	Sets the privilege levels for the **configuration**, **show** and **clear** commands.
ASA_route_map	Defines a route map for each OSPF process redistribution route map name.
ASA_RSA_KeyPair_generation	Resets and generates RSA key pairs for certificates.
ASA_svc_image	Provides an ASA SSL VPN Client image. It copies the SVC image from /CSCOpx/tftpboot/*device-hostname* on the Cisco Security Manager server to the device, then configures the SVC image path. Make sure you fill out the device's hostname in Device Properties. If the image name is different than the default, you can override it in Device Properties > Policy Object Overrides > Text Objects > AsaSvcImageName. Unassign this FlexConfig from the device after the image has been copied and configured.

Table 7-6   Predefined ASA FlexConfig Policy Objects (continued)

Name	Description
ASA_sysopt	Uses the **sysopt** command to provide the following examples: • Ensures that the maximum TCP segment size does not exceed the value you set or that the minimum is not less than a specified size. • Forces each TCP connection to remain in a shortened TIME_WAIT state of at least 15 seconds after the final normal TCP close-down sequence. • Disables DNS inspection that alters the DNS A record address. • Ignores the authentication key in RADIUS accounting responses. • Enables the web browser to supply a username and password from its cache when it reauthenticates with the virtual HTTP server on the security appliance.
ASA_virtual	Configures virtual HTTP and Telnet servers.

Table 7-7   Predefined Catalyst 6500/7600 FlexConfig Policy Objects

Name	Description
Cat6K_ECLB_algorithm	Sets the Ether Channel load balance algorithm for modules.
Cat6K_ECLB_port_mode	Applies an Ether Channel to the Catalyst trunk ports where IPS sensors are plugged in. Make sure the ports are configure in trunk mode.
Cat6K_ECLB_portchannel	Sets the port channel to trunk mode and add trunk-allowed VLANs.
Cat6K_firewall_multiple_vlan_interfaces	Sets multiple VLAN interfaces mode if multiple SVIs need to be provisioned.

Table 7-8   Predefined Cisco IOS FlexConfig Policy Objects

Name	Description
IOS_add_bridge_interface_desc	Loops through a list of bridge interfaces and adds the description "this is a bridge interface."
IOS_CA_server	Configures a certificate authority server.
IOS_compress_config	Compresses large Cisco IOS configurations.
IOS_config_root_wireless_station	Creates and configures the root radio station for a wireless LAN on Cisco IOS routers such as the 851 and 871.
IOS_console_AAA_bypass	Provides examples of the following scenarios: • Enables the authentication, authorization, and accounting (AAA) access-control model. • Sets AAA at login. • Enables AAA authentication for logins.

Table 7-8   Predefined Cisco IOS FlexConfig Policy Objects (continued)

Name	Description
IOS_Copy_Image	Copies the an SVC image from the Security Manager server to the device, then configures the SVC image path. Unassign this FlexConfig from the device after the image has been copied and configured.
IOS_enable_SSL	Enables SSL.
IOS_FPM	Copies traffic class definition files to a router and applies policy-maps.
IOS_IPS_PUBLIC_KEY	Defines public keys on an IOS IPS device. Public keys are required for Security Manager to perform signature updates.
IOS_IPS_SIGNATURE_CATEGORY	Retires all signatures except those in the ios_ips basic category.
IOS_PKI_with_AAA	Configures a PKI AAA authorization using the entire subject name.
IOS_set_clock	Sets the clock to the current time on the Security Manager server.
IOS_VOIP_advance	Loops through and associates a POTS port number to a telephone number and port or IP address number.
IOS_VOIP_simple	Associates a POTS port number to a telephone number and port number.
IOS_VPN_config_gre_tunnel	Uses VPN variables to configure the GRE tunnel for each VPN in which the device participates.
IOS_VPN_set_interface_desc	Using VPN variables, updates the description of the public interface for each VPN in which the device participates.
IOS_VPN_shutdown_inside_interface	Using VPN variables, shuts down all inside interfaces for each VPN in which the device participates.
IOS_VRF_on_vFW	Configures virtual routing and forwarding (VRF) on virtual firewall interfaces.

Table 7-9   Predefined PIX 6.3 Firewall FlexConfig Policy Objects

Name	Description
PIX6.3_nat0_acl_compiled	Generates a compiled access list for NAT 0 access-control lists.
PIX6.3_policy_nat_acl_compiled	Generates a compiled access list for Policy NAT ACLs
PIX6.3_policy_static_acl_compiled	Generates a compiled access list for Policy Static ACLs.
PIX_VPDN	Configures a virtual private dialup network (VPDN).

Table 7-10   Predefined Router FlexConfig Policy Objects

Name	Description
ROUTER_add_inspect_rules	Loops through and appends inspect rules.
ROUTER_BGP_no_auto_summary	Disables the auto route summary for each BGP process by using the **no auto-summary** sub-command.
	This FlexConfig policy object uses the list of border gateway protocol (BGP) numbers from the SYS_ROUTER_BGP_AS_NUMBERS_LIST system variable.
ROUTER_BGP_untrusted_info	Uses the **distance bgp 255 255 255** sub-command to make the border gateway protocol (BGP) routing information untrusted for each BGP.
	This FlexConfig policy object uses the list of BGP numbers from the SYS_ROUTER_BGP_AS_NUMBERS_LIST system variable.
ROUTER_EIGRP_min_cost_routes	Configures traffic to use minimum cost routes when multiple routes have different cost routes to the same destination network. This is done using multi-interface load splitting on different interfaces with equal cost paths.
	This FlexConfig policy object uses the list of router enhanced interior gateway routing protocol (EIGRP) numbers from the SYS_ROUTER_EIGRP_AS_NUMBERS_LIST system variable.
Router_EIGRP_no_auto_summary	Disables the auto route summary for each router enhanced interior gateway routing protocol (EIGRP) processes by using the **no auto-summary** sub-command. This FlexConfig policy object uses the list of EIGRP numbers from the SYS_ROUTER_EIGRP_AS_NUMBERS_LIST system variable.
ROUTER_interface_prevent_dos _attacks	Prevents denial-of-service (DOS) attacks on all device interfaces.
	This FlexConfig policy object uses the list of interface names from the SYS_INTERFACE_NAME_LIST system variable.
ROUTER_OSPF_no_router_Id	Removes the router OSPF ID for each OSPF process.
	This FlexConfig policy uses the list of OSPF IDs from the SYS_ROUTER_OSPF_PROCESS_IDS_LIST system variable.
ROUTER_QoS_Class_Map _description	Sets QoS class map descriptions.
	This FlexConfig policy object uses the list of router QoS class names from the SYS_ROUTER_QOS_CLASS_MAP_LIST system variable.

Table 7-10   Predefined Router FlexConfig Policy Objects (continued)

Name	Description
ROUTER_QoS_Policy_Map _description	Sets QoS policy descriptions.
	This FlexConfig policy object uses the list of router QoS policy names from the SYS_ROUTER_QOS_POLICY_MAP_LIST system variable.

Configuring FlexConfig Policies and Policy Objects

You create and manage FlexConfig policy objects in the same way that you create other policy objects. The following topics describe how to create FlexConfig policies and policy objects. For information on other tasks you can perform with FlexConfig policy objects (such as deleting them), see Working with Policy Objects—Basic Procedures, page 6-9.

- A FlexConfig Creation Scenario, page 7-25
- Creating FlexConfig Policy Objects, page 7-28
- Editing FlexConfig Policies, page 7-35

A FlexConfig Creation Scenario

This scenario takes you through the steps to set up Media Gateway Control Protocol (MGCP) for an ASA device using one of the predefined FlexConfig policy objects that are shipped with Security Manager. MGCP is used by the call agent application to control media gateways (devices that convert telephone circuit audio to data packets). Security Manager does not support MGCP configuration, but you can use a FlexConfig policy object to provide a configuration. This illustrates how FlexConfigs enable you to customize, for your network, what is not otherwise supported in Security Manager.

In this scenario, you do the following:

1. Create a policy object by duplicating an existing policy object.
2. Assign the policy object to a device.
3. Preview the configuration to verify that it is correct.
4. Share the policy object with another device.
5. Deploy the configuration to the devices.

You can use this scenario as an example to implement other features by creating copies of and modifying predefined FlexConfig policy objects or by creating your own objects.

Before You Begin

Add two ASA devices to Security Manager for this scenario.

Step 1   Duplicate the FlexConfig policy object by doing the following:

   a.   Select **Manage > Policy Objects** to open the Policy Object Manager (see Policy Object Manager, page 6-4).

b. Select **FlexConfigs** from the table of contents. The table in the right pane lists the existing FlexConfig objects.

c. Right-click ASA_MGCP FlexConfig and select **Clone Object**. The Add FlexConfig dialog box appears (see Add or Edit FlexConfig Dialog Box, page 7-30).

d. Enter a name for the new FlexConfig object, for this example, MyASA_MGCP.

e. Enter a new group name and a description of the object.

Tip The group name and description are optional. We recommend you establish descriptions and groups for objects you create.

f. Click **OK**. The new FlexConfig object appears in the list.

Step 2 Duplicate and edit the $callAgentList text object.

The original ASA_MGCP FlexConfig object uses the variable $callAgentList, which is a text object. The text object is read-only and cannot be edited. Duplicating the text object enables you to edit the duplicate object to apply to your network settings.

a. Select **Text Objects** from the table of contents.

b. Right-click **callAgentList** and select **Clone Object**. The Add Text Object dialog box appears.

c. Edit the name of the text object. For this example change it to mycallAgentList.

d. Double-click the first value in column A and enter the IP address for a call agent in your network. For this example, change the value to 10.10.10.10.

e. Double-click the first value in column B and enter the port number for a call agent in your network. For this example, change the value to 105.

f. Change the IP address and port number values for another call agent. For this example, change the IP address to 20.20.20.20 and the port number to 106. Or, if you have only one call agent in your network, you could remove the second row in the table by decreasing the number in the Number of Rows field. Similarly, if you have *more* than two call agents, you can add rows by increasing the number in this field.

This concept is similar for increasing and decreasing the number of columns by increasing or decreasing the Number of Columns field.

g. Click **OK**. The new text object appears in the list of text objects.

Step 3 Edit the new FlexConfig policy object to use the new variable by doing the following:

a. Select **FlexConfigs** from the table of contents.

b. Double-click MyASA_MGCP. The Edit FlexConfig dialog box appears.

c. Edit $callAgentList to read $mycallAgentList.

d. Click **OK**.

A warning appears that reads: "The following variables are undefined: mycallAgentList Define them now?"

e. Click **Yes** to the warning.

The FlexConfig Undefined Variables dialog box appears with mycallAgentList listed in the Variable Name column.

f. From the Object Type list, select **Text Objects**. The Text Objects window appears.

g. Select **mycallAgentList** from the Available Text Objects list and click **OK**.

h. In the FlexConfig Undefined Variables window, click **OK**.

The mycallAgentList variable appears in the Variables list of the Edit FlexConfig dialog box.

i. In the Edit FlexConfig dialog box, click **OK**.

j. Close the Policy Object Manager window.

Step 4 Assign the new FlexConfig policy object to a device by doing the following:

a. From the Device view, select the device for which you want to set up MGCP.

b. Select **FlexConfigs** from the Policy selector. The FlexConfigs Policy page appears.

c. Click the **Add** button. The FlexConfigs Selector dialog box appears.

d. Select the new MyASA_MGCP FlexConfig policy object and click **>>** to add the policy object to the Selected FlexConfigs column.

You can select multiple policy objects at one time by holding either the Ctrl (for multiple selections) or Shift (for multiple continuous selections) keys while selecting.

e. Click **OK**.

The MyASA_MGCP policy object is added to the Appended FlexConfigs table, because the object is set to be appended to the configuration. You configure FlexConfig policy objects that you want added to the beginning of the configuration as prepended policy objects.

f. Click **Save**.

Step 5 Preview the commands before they are generated and sent to the device by doing the following:

a. From the FlexConfigs Policy page, select the MyASA_MGCP policy object.

b. Click **Preview**.

The commands that are generated with this FlexConfig policy object and the values assigned to the selected device appear. Note the changed values:

```
class-map sj_mgcp_class
  match access-list mgcp_list
  exit

mgcp-map inbound_mgcp
       call-agent 10.10.10.10 105
       call-agent 20.20.20.20 106

       gateway 10.10.10.115 101
       gateway 10.10.10.116 102

  command-queue 150
  exit

policy-map inbound_policy
  class sj_mgcp_class
    inspect mgcp inbound_mgcp
    exit
exit

service-policy inbound_policy interface outside
```

Step 6 If you have additional ASA devices that require MGCP, you can share this policy with them by doing the following:

a. Right-click **FlexConfigs** in the Policy selector and select **Share Policy**.

The Share Policy dialog box appears.

b. Enter a name for the policy and click **OK**. For this example, enter MyShared_ASA_MGCP.

The banner above the FlexConfigs policy now shows that the device is using a shared policy and displays the name of the policy.

c. In the FlexConfigs banner, click the link in the Assigned To field. In this example, the link should be labeled **1 Device**, which indicates that this shared policy is assigned to one device (the device you are viewing).

Clicking the link opens the Shared Policy Assignments dialog box. Using this dialog box, you can select the other devices that should use this policy in the Available Devices list, and click **>>** to add them to the list of devices that are assigned the policy.

d. Click **OK**. The Shared Policy Assignments dialog box closes, and the additional devices you selected are configured to use the shared policy. The link in the banner changes to indicate the number of devices that now use this policy (in this example, **2 Devices**).

<div style="margin-left:2em">

🔎

Tip   You can also share policies from Policy view. Select **View > Policy View**, select FlexConfigs in the policy type selector, select the MyShared_ASA_MGCP policy, click the Assignments tab, select the devices to which you want to assign the policy, click**>>**, and then **Save**.

</div>

Step 7   Submit your changes and deploy the configurations to the devices. For information about deploying configurations, see Working with Deployment and the Configuration Archive, page 8-25.

Creating FlexConfig Policy Objects

You can create FlexConfig policy objects to configure features on devices that are not supported by Security Manager. For more information about FlexConfig objects, see Understanding FlexConfig Policies and Policy Objects, page 7-2.

🔎

Tip   You can also create FlexConfig policy objects when defining policies or objects that use this object type. For more information, see Selecting Objects for Policies, page 6-2.

Before You Begin

Ensure that your commands do not conflict in any way with the VPN or firewall configuration on the devices.

Keep the following in mind:

- Security Manager does not manipulate or validate your commands; it simply deploys them to the devices.

- If there is more than one set of commands for an interface, only the last set of commands is deployed. Therefore, we recommend you not use beginning and ending commands to configure interfaces.

- When editing FlexConfig objects that involve route-maps (for example, OSPF or multicast route-maps), you must define the corresponding access control lists (ACLs) before the route-maps. This is a device requirement. If you do not define ACLs before route-maps, you will get a deployment error.

Related Topics

- A FlexConfig Creation Scenario, page 7-25
- Working with Policy Objects—Basic Procedures, page 6-9
- Creating Policy Objects, page 6-9
- Chapter 5, "Managing Policies"

Step 1 Select **Manage > Policy Objects** to open the Policy Object Manager window (see Policy Object Manager, page 6-4).

Step 2 Select **FlexConfigs** from the Policy Object Type selector.

Step 3 Right-click inside the work area and select **New Object**.

The Add FlexConfig Object dialog box appears (see Add or Edit FlexConfig Dialog Box, page 7-30).

Step 4 Enter a name for the object and optionally a description. Other optional informational fields include:

- Group—Select an existing group name or type in a new one. These names can help you identify the use of an object.
- Negate For—If this FlexConfig object is designed to negate another, enter the name of the FlexConfig object whose commands are undone by this object.

Step 5 In the Type field, select whether commands in the object are to be prepended (put at the beginning) or appended (put at the end) of the configurations generated from Security Manager policies.

Step 6 In the object body area, enter the commands and instructions to produce the desired configuration file output. You can type in the following types of data:

- Scripting commands to control processing. For more information, see Using Scripting Language Instructions, page 7-3.
- CLI commands that are supported by the operating system running on the devices to which you will deploy the FlexConfig policy object. For more information, see Using CLI Commands in FlexConfig Policy Objects, page 7-2.
- Variables. You can insert variables using the right-click menu, which allows you to create simple single-value text variables (**Create Text Object**), select variables from existing policy objects (**Insert Policy Object**), or select system variables (**Insert System Variable**). For more information, see Understanding FlexConfig Object Variables, page 7-5.

🔍

Tip   If you want to remove a variable, select it in the object body and click the Cut button or press the Backspace or Delete key. When you click **OK** to save your changes, the variable is removed from the list of variables.

Step 7 Click the **Validate FlexConfig** icon button above the object body to check the integrity and deployability of the object.

Step 8 Click **OK** to save the object.

Add or Edit FlexConfig Dialog Box

Use the Add or Edit FlexConfig dialog box to create or edit FlexConfig policy objects. FlexConfig objects are small programs that allow you to add configuration commands before or after the configurations generated from Security Manager policies, so that you can extend the abilities of the product to configure your devices. You use these policy objects in FlexConfig device or shared policies.

Before creating FlexConfig policy objects, read the sections in Understanding FlexConfig Policies and Policy Objects, page 7-2.

Navigation Path

Select **Manage > Policy Objects**, then select **FlexConfigs** from the Object Type Selector. Right-click inside the work area and select **New Object** or right-click a row and select **Edit Object**.

Related Topics

- Creating FlexConfig Policy Objects, page 7-28
- Editing FlexConfig Policies, page 7-35
- Chapter 7, "Managing FlexConfigs"

Field Reference

Table 7-11   FlexConfigs Editor Dialog Box

Element	Description
Name	The object name, which can be up to 128 characters. Object names are not case-sensitive. For more information, see Creating Policy Objects, page 6-9.
Description	An optional description of the object.
Group	The name of the group of FlexConfig objects to which this object belongs, if any. You can type in a name, or select an existing name from the list. This field is for informational purposes only, and can help you find a FlexConfig object in the FlexConfig Objects page in the Policy Object Manager.
Type	Whether the commands in the object are prepended (put at the beginning) or appended (put at the end) of configurations.
Negate For	The name of the FlexConfig object whose commands are undone in this FlexConfig object. This field is for informational purposes only and does not affect the processing of the object.
	For example, if FlexConfig A has the command **banner login**, and FlexConfig B has the command **no banner login**, FlexConfig B negates the configuration for FlexConfig A.

Table 7-11   FlexConfigs Editor Dialog Box (continued)

Element	Description
FlexConfig Object Body	
Object Body edit box	The commands and instructions to produce the desired configuration file output. You can type in the following types of data:
	• Scripting commands to control processing. For more information, see Using Scripting Language Instructions, page 7-3.
	• CLI commands that are supported by the operating system running on the devices to which you will deploy the FlexConfig policy object. For more information, see Using CLI Commands in FlexConfig Policy Objects, page 7-2.
	• Variables. You can insert variables using the right-click menu, which allows you to create simple single-value text variables (**Create Text Object**), select variables from existing policy objects (**Insert Policy Object**), or select system variables (**Insert System Variable**). For more information, see Understanding FlexConfig Object Variables, page 7-5.
Undo button	Deletes the previous action.
Redo button	Performs the previously undone action.
Cut button	Deletes the highlighted text and copies it to the clipboard.
Copy button	Copies the highlighted text to the clipboard.
Paste button	Pastes previously cut or copied text.
Find button	Locates the specified text string in the object body.
Validate FlexConfig button	Checks the integrity and deployability of the FlexConfig object.
FlexConfig Object Variables	
This table lists the variables that are used in the FlexConfig object.	
Name	The name of the variable. Click the cell to edit the name, which also changes the name in the FlexConfig object body.
Default Value	The value to use when one is not provided. Click the cell to edit the value for user-defined variables. You cannot edit system-defined variables.
	Note Except for optional variables, if a default value is not provided, you must provide a value for the variable.
Object Property	The property of the object. The object property name is in the following format:
	type.*name*.data.*property*
	where
	• *Type*—The type of object, for example Text, Network, AAA Server, and so on.
	• *Name*—The name of the object.
	• Data—Indicates that the property of the object is data.
	• *Property*—The property of the data.

Table 7-11   FlexConfigs Editor Dialog Box (continued)

Element	Description
Dimension	The structure of the data in the variable. Possible values are: • 0—scaler (a single string) • 1—one-dimensional array (a list of strings) • 2—two-dimensional table (a table of strings)
Optional	Whether the variable is required to have a value.
Description	A description of the contents of the object. Click the cell to edit the description.

Create Text Object Dialog Box

Use the Create Text Object dialog box as a shortcut to create text objects of dimension 0, which are single-value variables, for use in FlexConfig policy objects. Enter the name of the variable and the value to assign to it. When you click **OK**, the variable is added to the FlexConfig object at the cursor location and it is added to the list of variables for the object.

Navigation Path

In the Add or Edit FlexConfig Dialog Box, right-click in the object body field and select **Create Text Object**.

$\mathcal{Q}$

Tip   If you want to create a multiple-value text object, right-click and select **Insert Policy Object > Text Objects**, and click the **Add** button under the available objects list. For more information, see Add or Edit Text Object Dialog Box, page 7-32.

Add or Edit Text Object Dialog Box

Use the Add or Edit Text Object dialog box to create, edit, duplicate, and view text objects. Create a text object if you need textual data to be referenced and acted upon by another policy object that accepts text objects.

Text objects are a type of policy object variable. They are a name and value pair, where the value can be a single string, a list of strings, or a table of strings. You can enter any type of textual data to be referenced and acted upon by FlexConfig policies. For more information about FlexConfigs, see Chapter 7, "Managing FlexConfigs".

Create the variable by first selecting the dimension: a simple single-value variable (dimension 0), a list of variables (dimension 1) or a table or variables (dimension 2). After you create the desired grid by selecting the dimension and if applicable, the number of rows and columns, enter the data into each cell by first clicking in the cell.

Navigation Path

Select **Manage > Policy Objects**, then select **Text Objects** from the Object Type Selector. Right-click inside the work area and select **New Object** or right-click a row and select **Edit Object**.

Field Reference

Table 7-12   Text Object Dialog Box

Element	Description
Name	The object name, which can be up to 128 characters. Object names are not case-sensitive. For more information, see Creating Policy Objects, page 6-9.
Description	An optional description of the object (up to 1024 characters).
Dimension	The structure of the data in the variable: • 0—scalar (a single string) • 1—one-dimensional array (a list of strings) • 2—two-dimensional table (a table of strings)
Number of Rows	The number of data rows in the variable if the dimension is 1 or 2.
Number of Columns	The number of data columns in the variable if the dimension is 2.
[text field]	The content of the text object. Click the cell and enter the data.
Category	The category assigned to the object. Categories help you organize and identify rules and objects. See Using Category Objects, page 6-13.
Allow Value Override per Device Overrides Edit button	Whether to allow the object definition to be changed at the device level. For more information, see Allowing a Policy Object to Be Overridden, page 6-18 and Understanding Policy Object Overrides for Individual Devices, page 6-18. If you allow device overrides, you can click the **Edit** button to create, edit, and view the overrides. The **Overrides** field indicates the number of devices that have overrides for this object.

FlexConfig Undefined Variables Dialog Box

Use the FlexConfig Undefined Variables dialog box to define variables used in the FlexConfig object that have not yet been defined. You can choose from a list of policy object types or add a new policy object to use.

Each row in the table represents a single undefined variable.

Tip   You do not need to define local variables, those used by the scripting language for processing control. For more information about variables, see Understanding FlexConfig Object Variables, page 7-5.

Navigation Path

In the Add or Edit FlexConfig Dialog Box, if you enter a variable name but do not define its values, when you click **OK**, Security Manager displays a warning and asks if you want to define the variables. If you click **Yes**, this dialog box is opened.

Field Reference

Table 7-13   FlexConfig Undefined Variables Dialog Box

Element	Description
Variable Name	The name of the undefined variable that you used in the FlexConfig object.
Object Type	The type of policy object that contains the value you want to assign to the variable. For local variables, use the Undefined object type.
	For variables you want to define, you must select the specific policy object and value within that object to assign to the selected variable.
	You start by selecting the type of policy object from this list. You are then prompted to select the specific policy object. When you click **OK**, you are prompted to select the specific property within that object that contains the desired value (see Property Selector Dialog Box, page 7-34). When you select the value on the Property Selector dialog box and click **OK**, the value is assigned to the variable.
Object Property	The property of the object. For a detailed explanation, see Add or Edit FlexConfig Dialog Box.
Optional	Whether the variable is required to have a value.

Property Selector Dialog Box

Use the Property Selector dialog box to select the specific property within a selected policy object that you want to assign to a variable within a FlexConfig policy object. The title of the dialog box indicates the type of policy object that you selected (for example, AAA Server Groups Property Selector).

For more information on variables, see Understanding FlexConfig Object Variables, page 7-5.

Navigation Path

- In the Add or Edit FlexConfig Dialog Box, right-click and select a specific policy object group type from the **Insert Policy Object** menu, select a specific policy object when prompted, and click **OK**.

- In the FlexConfig Undefined Variables Dialog Box, select a policy object type from the Object Type field, select a specific policy object when prompted, and click **OK**.

Field Reference

Table 7-14   Property Selector Dialog Box

Element	Description
Object Property	The property of the object that contains the value you want to assign to the variable. For specific information on the properties, see topics related to the configuration of those objects.
Name	The name of variable. This field is not available when you are defining undefined variables.
Description	An optional description of the variable. This field is not available when you are defining undefined variables.

Editing FlexConfig Policies

You can assign FlexConfig policies to devices using either Device view or Policy view (for shared policies) by selecting **FlexConfigs** from the policy selector. You can deploy configurations containing these policies as you would deploy any configuration generated by Security Manager. For a scenario that takes you through setting up a FlexConfig policy object and creating a shared FlexConfig policy, see A FlexConfig Creation Scenario, page 7-25.

> ✎
> **Note**   With "Deploy only new or modified Flexconfigs" setting on **Tools > Security Manager Administration > Deployment page** enabled, if you have an activity open, with changes, and when you attempt to deploy FlexConfigs, Cisco Security Manager considers the FlexConfig changes specific to that activity alone and not those of other activities. On the other hand, if all activities are submitted and no activity is open, then Security Manager considers the FlexConfig changes specific to the lastly submitted activity that was submitted with changes. Therefore, if you need FlexConfig changes for an activity to reflect during deployment, then ensure the changes are done in a single activity, submitted, and deployed.

When you edit a FlexConfig policy, you can perform the following actions:

- **Add FlexConfig objects**—To add a FlexConfig object to a policy, click the Add icon button and select the desired object. You can also create new objects from the object selector dialog box. The objects are added to the prepended or appended list depending on how the objects themselves are defined.

- **Remove FlexConfig objects**—If you no longer want to include an object in the policy, select it and click the Remove icon button. This action removes the object from the policy, but it does not delete the object from Security Manager. For information on deleting objects, see Deleting Objects, page 6-16.

- **Change the order of the objects**—Objects are processed in the order you specify. If an object depends on the processing of another object, it is important that you order them correctly. Select the object whose order you want to change and click the Up or Down arrow buttons until the object is in the desired location.

  When changing the order of FlexConfig objects that involve route-maps (for example, OSPF or multicast route-maps), make sure that the corresponding access control lists (ACLs) are defined before the route-maps. This is a device requirement. If you do not define ACLs before route-maps, you will get a deployment error.

- **Change the values assigned to the variables used in a policy object**—If you want to configure a variable with a different value for a particular device, creating a device-level override for the object, select the object and click **Values**. In the Values Assignment dialog box, click in the Values cell to change the value. For more information, see Values Assignment Dialog Box, page 7-37.

- **Preview the CLI that will be generated for a policy object**—In Device view, you can view the CLI that will be generated for a policy object by selecting the object and clicking **Preview**. This is especially useful for checking that the CLI commands generated are what you intend to implement on the device.

  > ✎
  > **Note**   During deployment, when the FlexConfig policy objects are compiled on the Security Manager server, the correct system variable values and settings are used to generate commands. However, because the Preview function does not have access to these values the way it normally would during deployment, it might not display some CLI commands. In addition, because the Preview function generates CLI commands on the client, some macros used in FlexConfig policy objects reflect client settings instead of server settings.

Related Topics

- Understanding FlexConfig Policies and Policy Objects, page 7-2
- Creating FlexConfig Policy Objects, page 7-28
- Chapter 5, "Managing Policies"
- Chapter 8, "Managing Deployment"

FlexConfig Policy Page

Use the FlexConfig Policy page to create FlexConfig policies. FlexConfig policies contain ordered lists of FlexConfig policy objects, which are subroutines that allow you to extend the ability of Security Manager to configure your devices. For more information on FlexConfig policy objects, see Understanding FlexConfig Policies and Policy Objects, page 7-2.

Navigation Path

- (Device view) Select **FlexConfigs** from the Policy selector.
- (Policy view) Select **FlexConfigs** from the Policy Type selector and select an existing policy or click the Create a Policy button to create a new one.

Related Topics

- Creating FlexConfig Policy Objects, page 7-28
- Chapter 7, "Managing FlexConfigs"

Field Reference

Table 7-15   FlexConfigs Policy Page

Element	Description
Prepended FlexConfigs	The FlexConfig policy objects that are added to the beginning of the configuration. The objects are processed in the order shown.
Appended FlexConfigs	The FlexConfig policy objects that are added to the end of the configuration. The objects are processed in the order shown.
Values button	Click this button to view, modify, or validate the values assigned to the variables used in the selected FlexConfig policy object using the Values Assignment Dialog Box, page 7-37.
Preview button (Device view only.)	Click this button to view the CLI commands that will be generated for the selected FlexConfig policy object. In Policy view, you can preview CLI by first clicking **Values**, selecting a device in the Values Assignment dialog box, and clicking **Preview**.
Up/Down arrow buttons	Click these buttons to move the selected object up or down in the list. The objects are processed in the displayed order, so it is important that an object whose processing depends on the processing of another object comes after the object it depends on.
Add button	Click this button to add a FlexConfig policy object to the policy. The object itself defines whether it will be added to the prepended or appended list. You can create new FlexConfig objects or select existing ones.

Table 7-15   FlexConfigs Policy Page (continued)

Element	Description
Edit button	Click this button to edit the selected FlexConfig policy object. Your changes affect all devices that use the edited object; your changes are not local policy object overrides for the device. **Note** If you selected a predefined FlexConfig policy object packaged with Security Manager, or an object for which you do not have edit permission, you are allowed only to view the object definition.
Remove button	Click this button to remove the selected object from the policy The object is not deleted from Security Manager; it is simply removed from the FlexConfig policy.

Values Assignment Dialog Box

Use the Values Assignment dialog box to view the variables used in a FlexConfig policy object, validate the object, or preview the CLI generated from the object. For more information, see Understanding FlexConfig Object Variables, page 7-5.

Navigation Path

Select an object and click **Values** from the FlexConfig Policy Page, page 7-36.

Field Reference

Table 7-16   Values Assignment Dialog Box

Element	Description
Assigned Devices (Policy view only)	The devices to which the shared FlexConfig policy has been assigned. Select the device for which you want to display variable values.
Name	The name of the variable.
Value	The value to use for the variable. To change the value, double-click the cell. When you change this value, Security Manager creates a device-level override for the policy object. If the policy object is configured so that its values cannot be overridden, you cannot edit the value. If there is no default value for the variable, you must provide a value unless it is an optional variable.
Default Value	The value assigned to the variable in the policy object. Double-click this cell to view the definition of the policy object that defines the variable's value.
Override	Whether you can override the value of the variable. You can edit the value of only those variables that have a checkmark in this column.
Object Property	The property of the object. For a detailed explanation, see Add or Edit FlexConfig Dialog Box.

Table 7-16   Values Assignment Dialog Box (continued)

Element	Description
Dimension	The structure of the data in the variable: • 0—scalar (a single string) • 1—one-dimensional array (a list of strings) • 2—two-dimensional table (a table of strings)
Optional	Whether the variable value can be empty.
Description	A description of the variable.
Validate button	Click this button to validate the Velocity Template Language syntax and make sure that all required variables have values, that variables do not start with SYS_, and that referenced policy objects exist.
Preview button	Click this button to display the generated CLI commands for the selected FlexConfig policy object.

FlexConfig Preview Dialog Box

Use the FlexConfig Preview dialog box to view the generated CLI commands based on the variables of the selected object defined in the FlexConfig policy.

Navigation Path

To open the FlexConfig Policy Preview dialog box, do one of the following:

- In the Values Assignment Dialog Box, click **Preview**. In Policy view, you must first select a device.
- (Device view) Select a device and click **FlexConfig** (see FlexConfig Policy Page, page 7-36). Select an object in the FlexConfig policy and click **Preview**.

Troubleshooting FlexConfigs

Problem: When adding a FlexConfig using the Cisco Security Manager client, you might receive the following error message:

```
Syntax Error: Failed to setup Velocity Engine to validate syntax.
```

This problem is due to administrator privilege rights on Microsoft Windows. In Microsoft Windows Vista and Microsoft Windows 7, Security Manager requires administrator privileges in order to use the FlexConfig feature.

Solution: To resolve this issue, launch the Security Manager client with administrator privileges in either of the following ways:

- To launch the Security Manager client with administrator privileges, right-click the **Configuration Manager** shortcut and select **Run as administrator**.
- To permanently enable administrator privileges for the Security Manager client, right-click the **Configuration Manager** shortcut and select **Properties**. On the Compatibility tab, select **Run this program as an administrator**, and then select **OK**.

Problem: When using FlexConfigs to deploy to an ASA firewall the following two commands in one job: **reload in x noconfirm**, **reload cancel**, you will receive the following error message:

```
An error response from the device prevented successful completion of this
operation. The device provided the following description: reload cancel No
reload is scheduled
```

Unfortunately, deployment always fails due to fact that both commands are pushed too fast, such that the **reload cancel** is sent before the reload schedule is activated on the device.

Solution: To work around this problem, the commands must be sent in two separate deployments created manually.

Problem: When a FlexConfig is assigned and deployed to a device, the FlexConfig is sometimes shown in subsequent Full configuration previews even after the FlexConfig is deleted from a device.

Solution: No workaround is needed. The FlexConfig will not be included in the deployment because only the delta configuration is pushed to the device during deployment.