



CHAPTER 19

Managing FlexConfigs

Security Manager provides tools to configure most parameters needed to manage your devices. For those parameters for which there is not a tool for configuration, and for certain customized applications, Security Manager provides the FlexConfig feature. The FlexConfig feature provides a simple way for you to write configuration commands, variables, and scripts and save these as FlexConfig policy objects. A FlexConfig policy object's contents can range from a single simple command string to elaborate CLI command structures that incorporate scripting and variables.

FlexConfig policy objects are reusable, named components that can be referenced by other policy objects and policies. FlexConfig policy objects simplify the distribution and reuse of CLI commands to manage your devices.

FlexConfig policy objects can be contained within a FlexConfig policy. You use the FlexConfig policies, as you would other policies, to define particular aspects of network configuration and to produce various configuration assignment and deployment results.

Understanding policies and objects is central to understanding and using FlexConfig policy objects. For more information on how Security Manager defines and uses policies, see [Chapter 6, “Managing Policies,”](#) and for information on how Security Manager defines and uses objects, see [Chapter 8, “Managing Objects.”](#)

The following topics describe the FlexConfig feature and how to use it:

- [Understanding FlexConfig Policy Objects, page 19-2](#)
- [Understanding FlexConfig Policies, page 19-36](#)
- [A FlexConfig Creation Scenario, page 19-36](#)

- [Configuring FlexConfig Policy Objects, page 19-41](#)

Understanding FlexConfig Policy Objects

FlexConfig policy objects are reusable, named components that can be referenced by other policy objects and policies. You create FlexConfig policy objects by entering configuration commands, either with or without additional scripting language instructions, in the FlexConfig.

Or, you can create a FlexConfig policy object by duplicating and then modifying an existing FlexConfig policy object, either one that you have created, or one of the predefined FlexConfig policy objects that are shipped with Security Manager.

The following topics describe FlexConfig policy objects:

- [CLI Commands, page 19-2](#)
- [Scripting Language Instructions, page 19-3](#)
- [Object Variables, page 19-6](#)
- [FlexConfig Policy Object Example, page 19-7](#)
- [Predefined FlexConfig Policy Objects, page 19-7](#)
- [FlexConfig System Variables, page 19-13](#)

For more information about policy objects in general, see [Chapter 8, “Managing Objects.”](#)

CLI Commands

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 3.1. 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 device type (Cisco Router, PIX Firewall, and so on) for more information.

You can add commands and instructions to the beginning or end of the configurations:

- Prepended commands—Commands placed at the beginning of the configurations. Prepended commands are always replaced when configuration files are deployed.
- Appended commands—Commands placed after all other commands in a configuration file and before the **write mem** command are called appended commands.

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 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.

Scripting Language Instructions

When creating or editing a FlexConfig policy object you have the option to use scripting language instructions. Scripting language instructions are a subset of commands supported in the Velocity Template Engine, a Java-based scripting language, where control flows, such as looping and if/else statements, and variables can be used.

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

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

- [Example 1: Looping, page 19-4](#)
- [Example 2: Looping with Two-Dimensional Arrays, page 19-4](#)
- [Example 3: Looping with If/Else Statements, page 19-5](#)

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
```

Example 2: Looping with Two-Dimensional Arrays

In this example, a set of phone numbers is associated to voice ports, so 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 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
```

Object Variables

There are three types of variables you employ in a FlexConfig:

- **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.
- **System Variables**—Dynamic variables that reference a value during deployment when the CLI is generated. The values are obtained from either the deploying device or policies configured for the deploying device. System variables can be declared 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.
- **Local Variables**—Variables that are local in the looping and assignment derivatives (*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.

You can manually enter variables (denoted with a starting \$ character) in an object. For example:

```
interface $inside
```

FlexConfig Policy Object Example

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

You enter these commands:

```
interface $inside
crypto map $xyz
```

You enter these variable assignments:

```
$inside = "serial0"
$xyz = "my_crypto"
```

When the configuration is generated, the following output is created from the commands and variables you entered:

```
interface serial0
crypto map my_crypto
```

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 permanently set as 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:

- Predefined ASA FlexConfig Policy Objects—[Table 19-2](#)
- Predefined Cisco IOS FlexConfig Policy Objects—[Table 19-2](#)
- Predefined PIX Firewall FlexConfig Policy Objects—[Table 19-3](#)
- Predefined Router FlexConfig Policy Objects—[Table 19-4](#)

Table 19-1 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_command_alias	Creates a command alias named “save” for the copy running-config and copy startup_config commands.
ASA_csd_image	Provides an ASA Cisco Secure Desktop image. Copy csd image from CSM server /CSCOpX/tftpboot/device-hostname to device, then configure 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 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 19-1 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	<p>Uses the ip-audit command to provide the following:</p> <ul style="list-style-type: none"> • 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. Copy svc image from CSM server /CSCOpX/tftpboot/ <i>device-hostname</i> to device, then configure 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 19-1 Predefined ASA FlexConfig Policy Objects (continued)

Name	Description
ASA_sysopt	Uses the sysopt command to provide the following examples: <ul style="list-style-type: none"> • 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 19-2 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 server.
IOS_compress_config	Compresses large Cisco IOS configurations.
IOS_console_AAA_bypass	Provides examples of the following scenarios: <ul style="list-style-type: none"> • Enables the authentication, authorization, and accounting (AAA) access-control model. • Sets AAA at login. • Enables AAA authentication for logins.
IOS_enable_SSL	Enables SSL.

Table 19-2 *Predefined Cisco IOS FlexConfig Policy Objects (continued)*

Name	Description
IOS_FPM	Copies traffic class definition files to a router and applies policy-maps.
IOS_set_clock	Sets the clock to the current time on the Security Manager server.
IOS_VOIP_advanced	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.
IOS_config_root_wireless_station	Creates and configures the root radio station for a wireless LAN on Cisco IOS 851 or 871 routers.

Table 19-3 *Predefined PIX 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 19-4 Predefined Router FlexConfig Policy Objects

Name	Description
ROUTER_add_inspect_rules	Loops through and appends inspect rules.
ROUTER_BGP_no_auto_summary	<p>Disables the auto route summary for each BGP process by using the no auto-summary sub-command.</p> <p>This FlexConfig policy object uses the list of border gateway protocol (BGP) numbers from the SYS_ROUTER_BGP_AS_NUMBERS_LIST system variable.</p>
ROUTER_BGP_untrusted_info	<p>Uses the distance bgp 255 255 255 sub-command to make the border gateway protocol (BGP) routing information untrusted for each BGP.</p> <p>This FlexConfig policy object uses the list of BGP numbers from the SYS_ROUTER_BGP_AS_NUMBERS_LIST system variable.</p>
ROUTER_EIGRP_min_cost_routes	<p>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.</p> <p>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.</p>
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	<p>Prevents denial-of-service (DOS) attacks on all device interfaces.</p> <p>This FlexConfig policy object uses the list of interface names from the SYS_INTERFACE_NAME_LIST system variable.</p>

Table 19-4 Predefined Router FlexConfig Policy Objects (continued)

Name	Description
ROUTER_OSPF_router_ID_reset	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.
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.

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 and policies. The values for these variables are required unless otherwise noted. For information about these variables, see the following tables:

- Device system variables—[Table 19-5](#). For more information about discovering or configuring devices to obtain values for these variables, see [Chapter 5, “Managing Devices.”](#)
- Firewall system variables—[Table 19-6](#). For more information about creating Firewall system variables, see [Chapter 15, “Managing Firewall Devices”](#) and [Chapter 12, “Managing Firewall Services.”](#)
- Router platform system variables—[Table 19-7](#). For more information about creating router system variables, see [Chapter 14, “Managing Routers.”](#)
- VPN system variables—[Table 19-8](#). For more information about creating VPN system variables, see [Chapter 9, “Managing Site-to-Site VPNs.”](#)
- Remote access system variables—[Table 19-9](#). For more information about creating remote access system variables, see [Chapter 10, “Managing Remote Access VPNs.”](#)

Table 19-5 *Device System Variables¹*

Name	Dimension	Description
SYS_DOMAIN_NAME	0	<p>The DNS domain name.</p> <p>Discover or configure devices on Security Manager to generate values for this variable.</p>
SYS_FW_OS_MODE	0	<p>OS mode of the FWSM or ASA device. Valid values are ROUTER (routed mode), TRANSPARENT, or NOT_APPLICABLE.</p> <p>Discover or configure device operating system information (Tools > Device Properties > General) to generate values for this variable.</p> <p>This variable applies only to FWSM or ASA devices.</p>
SYS_FW_OS_MULTI	0	<p>Device OS context (single or multi mode). Valid values are SINGLE, MULTI, or NOT_APPLICABLE.</p> <p>This variable applies only to FWSM or ASA devices.</p> <p>Discover or configure device properties (Tools > Device Properties > General) to generate values for this variable.</p>
SYS_HOSTNAME	0	<p>The device's hostname.</p> <p>Discover or configure devices on Security Manager to generate values for this variable.</p>
SYS_IMAGE_NAME	0	<p>The device's image name.</p> <p>Discover or configure devices on Security Manager to generate values for this variable.</p>

Table 19-5 Device System Variables¹ (continued)

Name	Dimension	Description
SYS_INTERFACE_IP_LIST	1	<p>IP addresses and masks of the interfaces configured in the Interface policy.</p> <p>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 will be returned.</p> <p>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.</p> <p>Configure interface policies on the device to generate values for this variable.</p> <p>This variable is optional.</p>

Table 19-5 Device System Variables¹ (continued)

Name	Dimension	Description
SYS_INTERFACE_NAME_LIST	1	<p>Names of the interfaces on the device. If no interfaces are defined on the device, no list is returned.</p> <p>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.</p> <p>Discover or configure interfaces on the device to generate values for this variable.</p> <p>This variable is optional.</p>
SYS_MANAGEMENT_IP	0	<p>Management IP address of the device.</p> <p>Discover or configure device IP addresses (Tools > Device Properties > General) to generate values for this variable.</p>
SYS_MDF_TYPE	0	<p>The Cisco MDF (MetaData Framework) Type of the device. Indicates the device model.</p> <p>Discover or configure devices on Security Manager to generate values for this variable.</p>
SYS_OS_RUNNING_VERSION	0	<p>The software version of the OS running on the device. Version string could be 6.1, 6.2, and so on, on a PIX platform; 12.1, 12.2S, and so on, on an IOS platform; and 3.5, 4.1, and so on in an IDS platform.</p> <p>Discover or configure devices on Security Manager to generate values for this variable.</p>

Table 19-5 Device System Variables¹ (continued)

Name	Dimension	Description
SYS_OS_TARGET_VERSION	0	Indicates the OS version to be used when generating the device configuration. Discover or configure devices on Security Manager to generate values for this variable.
SYS_OS_TYPE	0	Device OS type. Valid values are IOS, PIX, ASA, CATOS, FWSM, IDS. Discover or configure device properties (Tools > Device Properties > General) to generate values for this variable.
SYS_SYS_OID	0	The SysObjId of the device. Discover or configure devices on Security Manager to generate values for this variable.

1. Device variables apply to all device types.

Table 19-6 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 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 FPM FlexConfig.

Table 19-6 Firewall System Variables (continued)

Name	Dimension	Description
SYS_FPM_OUTPUT_SP	1	<p>FPM policy map names applied on the interface corresponding to the entry in the SYS_FPM_INTERFACE list in the “out” direction.</p> <p>This data is not configured in Security Manager. It is obtained from a router’s running configuration and is used by the FPM FlexConfig.</p>
SYS_FW_ACL_IN_NAME	1	<p>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.</p> <p>Configure firewall access rules to generate values for this variable.</p>
SYS_FW_ACL_OUT_NAME	1	<p>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.</p> <p>Configure Access Rules policies to generate values for this variable.</p>

Table 19-6 Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_BRIDGE_INTERFACE_NAMES	1	<p>Names of bridge interfaces.</p> <p>This variable applies only to IOS transparent firewalls.</p> <p>Configure Policy Firewall > Settings > Transparent to generate values for this variable.</p>
SYS_FW_ETHERTYPERULE_ACL_NAMES	1	<p>Names of etherstype 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.</p> <p>Configure Firewall transparent rules policies to generate values for this variable.</p>
SYS_FW_ETHERTYPERULE_DIRECTION_NAMES	1	<p>Direction that etherstype 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.</p> <p>Configure Firewall transparent rules policies to generate values for this variable.</p>

Table 19-6 Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_ETHERTYPE_RULE_INTERFACE_NAMES	1	<p>Interface names to which ether type access-lists are applied. Each element has a one-to-one correspondence with the SYS_FW_ETHERTYPE_RULE_ACL_NAMES and SYS_FW_ETHERTYPE_RULE_DIRECTION_NAMES variables.</p> <p>Configure Firewall transparent rules policies to generate values for this variable.</p>
SYS_FW_INSPECT_IN_NAME	1	<p>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.</p> <p>Configure Inspection Rules policies to generate values for this variable.</p> <p>This variable is optional.</p>
SYS_FW_INSPECT_OUT_NAME	1	<p>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.</p> <p>Configure Inspection Rules policies as values for this variable.</p> <p>This variable is optional.</p>

Table 19-6 Firewall System Variables (continued)

Name	Dimension	Description
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_MPCRULE_TRAFFICFLOW_TUNNELGROUPNAME	1	Names of tunnel groups specified in Traffic Flow objects. Traffic Flow objects configure class-map commands on PIX Firewalls, and the names of the tunnel groups listed in Traffic Flow objects populate this variable. This variable is used by the <code>define_traffic_flow_tunnel_group</code> FlexConfig object to create tunnel groups on PIX firewalls. This variable is optional.

Table 19-6 Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_MULTICAST_PIM_ACCEPT_REG_ROUTE_MAP	0	<p>Route-map name used in the pim accept-register route-map command.</p> <p>Enter a name for the route-map (Platform > Multicast > PIM > Request Filter), then configure its features using FlexConfig to generate values for this variable.</p> <p>This variable is optional.</p>
SYS_FW_NAT0_ACL_NAMES	1	<p>Names of ACLs used in the nat interface_name 0 access-list acl_name command.</p> <p>This variable is optional.</p>
SYS_FW_OSPF_PROCESS_ID_LIST	1	<p>IDs for OSPF routing processes globally configured on PIX Firewalls, Firewall Service Modules, and ASA devices.</p> <p>Configure OSPF (Platform > Routing > OSPF) to generate values for this variable.</p>
SYS_FW_OSPF_REDISTRIBUTION_ROUTE_MAP_LIST	1	<p>Names for the route maps to apply to the OSPF redistribute commands configured on PIX Firewalls, Firewall Service Modules, and ASA devices.</p> <p>Configure the OSPF policy to generate values for this variable.</p>

Table 19-6 Firewall System Variables (continued)

Name	Dimension	Description
SYS_FW_POLICY_NAT_ACL_NAMES	1	<p>Names of ACLs used in the policy nat commands (nat commands with non-0 pool id).</p> <p>Configure NAT (NAT > Translation Rules > Policy NAT) to generate values for this variable. This variable applies to only PIX 6.3(3), PIX/ASA 7.0, and FWSM devices. This variable does not apply to Cisco IOS routers.</p> <p>This variable is optional.</p>
SYS_FW_POLICY_STATIC_ACL_NAMES	1	<p>Names of ACLs used in the policy static commands that include access lists.</p> <p>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.</p> <p>This variable applies to only PIX 6.3(3), PIX/ASA 7.0, and FWSM devices. This variable does not apply to Cisco IOS routers.</p> <p>This variable is optional.</p>

Table 19-7 Router Platform System Variables

Name	Dimension	Description
SYS_ROUTER_BGP_AS_NUMBERS_LIST	1	<p>Autonomous system (AS) number of the border gateway protocol (BGP) and exterior gateway protocol (EGP) on the device.</p> <p>Configure BGP policies as values for this variable (Router Platform > Routing > BGP).</p> <p>This variable is optional.</p>
SYS_ROUTER_EIGRP_AS_NUMBERS_LIST	1	<p>Autonomous system (AS) numbers of the different enhanced internet gateway routing protocols (EIGRP) and interior gateway protocols (IGP) on the device.</p> <p>Configure EIGRP policies as values for this variable (Router Platform > Routing > EIGRP).</p> <p>This variable is optional.</p>
SYS_ROUTER_OSPF_PROCESS_IDS_LIST	1	<p>Open shortest path first (OSPF) interior gateway protocol (IGP) process numbers on the device.</p> <p>Configure OSPF Process policies as values for this variable (Router Platform > Routing > OSPF Process).</p> <p>This variable is optional.</p>
SYS_ROUTER_QOS_CLASS_MAP_LIST	1	<p>Names of QoS class maps on the device.</p> <p>Configure Quality of Service policies to generate values for this variable.</p> <p>This variable is optional.</p>
SYS_ROUTER_QOS_POLICY_MAP_LIST	1	<p>Names of the QoS policy-maps on the device.</p> <p>Configure Quality of Service policies to generate values for this variable.</p> <p>This variable is optional.</p>

Table 19-8 VPN System Variables

Name	Dimension	Description
Topology —Variables related to the VPN in which a device participates. For more information, see Creating a VPN Topology, page 9-20 .		
SYS_VPN_TOPOLOGY	1	Virtual private network (VPN) topology type. Valid 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. Configure VPNs to generate values for this variable.
SYS_VPN_TOPOLOGY_ROLE	1	Details about the role of the device in the VPN. Valid values are PEER, HUB, and SPOKE. Configure VPNs to generate values for this variable.
Devices —Variables related to devices in the VPN in which a device participates. For more information, see Creating a VPN Topology, page 9-20 .		
SYS_VPN_HOST_NAME	1	Device host name. Configure VPNs to generate values for this variable.
SYS_VPN_LOCAL_PREFIXES	2	Interface and network IP addresses of protected networks. Configure VPNs to generate values for this variable.

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_PRIVATE_INTERFACES	2	Private interface names. Configure VPNs to generate values for this variable.
SYS_VPN_PRIVATE_TUNNEL_ENDPT_IP	1	Interface tunnel IP address. Configure VPNs to generate values for this variable.
SYS_VPN_PUBLIC_INTERFACES	2	Public interface names. Configure VPNs to generate values for this variable.
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.) Configure VPNs to generate values for this variable.
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.) Configure VPNs to generate values for this variable.
SYS_VPN_VPNISM_PUBLIC_IFC	2	Export port names (for Catalyst 6000 series switches only).

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
Remote Peers —Variables related to remote peers in which a device participates. For more information, see Creating a VPN Topology, page 9-20 .		
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). Configure VPNs to generate values for this variable.
SYS_VPN_REM_PEER_BAK_PREFIX	3	Protected networks (interface and network IP addresses) of remote peers of failover hubs. Configure VPNs to generate values for this variable.
SYS_VPN_REM_PEER_BAK_PUBLIC_IP	3	Public interface names of remote peers of failover hubs. Configure VPNs to generate values for this variable.
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.) Configure VPNs to generate values for this variable.

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_REM_PEER_DEVICE_NAME	2	Device host names of remote peers. Configure VPNs to generate values for this variable.
SYS_VPN_REM_PEER_LOGICAL_PRIVATE_IP	2	Interface tunnel IP addresses of remote peers. This value is used in DMVPN for next hop resolution protocol (NHRP). Configure VPNs to generate values for this variable.
SYS_VPN_REM_PEER_PREFIX	3	Protected networks (interface and network IP addresses) of remote peers. Configure VPNs to generate values for this variable.
SYS_VPN_REM_PEER_PRIVATE_IP	2	Private interface names of remote peers. Configure VPNs to generate values for this variable.
SYS_VPN_REM_PEER_PUBLIC_IP	2	Public interface names of remote peers. Configure VPNs to generate values for this variable.

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_REM_PEER_TUNNEL_SRC	2	Tunnel sources (if included in the interface tunnel of remote peers). Configure VPNs to generate values for this variable.
IPSec Proposal—Variables related to policy IPSec proposals. For more information, see Configuring IPSec Proposals, page 9-77 and Configuring High Availability in Your VPN Topology, page 9-60.		
SYS_VPN_CRYPTOMAP_TYPE	1	Crypto map type. Valid values are STATIC and DYNAMIC. Configure an IPSec proposal policy to generate values for this variable.
SYS_VPN_DYNAMIC_CRYPTOMAP_NAME	1	Dynamic crypto map name. Configure VPNs to generate values for this variable.
SYS_VPN_DYNAMIC_CRYPTOMAP_NUM	1	Dynamic crypto map number. Configure VPNs to generate values for this variable.
SYS_VPN_STATIC_CRYPTOMAP_NAME	1	Static crypto map name. Configure VPNs to generate values for this variable.

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_STATIC_CRYPTONAME_BAK	1	Static crypto map name of failover hubs. Configure VPNs to generate values for this variable.
SYS_VPN_STATIC_CRYPTONUM	2	Static crypto map number. Configure VPNs to generate values for this variable.
SYS_VPN_STATIC_CRYPTONUM_BAK	2	Static crypto map number of failover hubs. Configure VPNs to generate values for this variable.
Preshared Keys—Variables related to preshared key/IKE policies. For more information, see Configuring Preshared Key Policies, page 9-86.		
SYS_VPN_IKE_AUTHENTICATION_MODE	1	Authentication method of IKE policy. Valid 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.

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_NEGOTIATION_MODE	1	<p>Negotiation method. Valid values are MAIN_ADDRESS, MAIN_HOST, and AGGRESSIVE.</p> <p>Configure a Preshared Key policy to generate values for this variable.</p>
<p>GRE Modes—Variables related to GRE Modes policies. For more information, see Configuring GRE or GRE Dynamic IP Policies, page 9-99.</p>		
SYS_VPN_BAK_TUNNEL_IFC	2	<p>Interface tunnel number. (Matches the tunnel number of remote peers of failover hubs, for example, tunnel0.)</p> <p>Configure VPNs to generate values for this variable.</p>
SYS_VPN_SIGP_PROCESS_NUMBER	1	<p>Process number of interior gateway protocol (IGP).</p> <p>Configure GRE Modes policies to generate values for this variable.</p>
SYS_VPN_SIGP_ROUTING_PROTOCOL	1	<p>Type of secured interior gateway protocol (IGP) used. Valid values are STATIC, OSPF, EIGRP, RIPV2, BGP, and ODR.</p> <p>Configure GRE Modes policies to generate values for this variable.</p>

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_SPOKE_TO_SPOKE_CONN	1	Indication whether DMVPN is configured for spoke-to-spoke connectivity. Valid values are true or false. Configure GRE Modes policies to generate values for this variable.
SYS_VPN_TUNNEL_IFC	2	Interface tunnel number. (Matches the tunnel number of remote peers, for example, tunnel0.) Configure VPNs to generate values for this variable.
VRF—Variables related to VRF. For more information, see Configuring VRF-Aware IPsec Settings, page 9-56.		
SYS_VPN_VRF_AREA_ID	1	Area ID numbers (if the OSPF process number was chosen). Configure VPNs to generate values for this variable.
SYS_VPN_VRF_MPLS_INTERFACE_IP	1	Multiprotocol label switching (MPLS) interface IPs. Configure VPN VRF settings to generate values for this variable.

Table 19-8 **VPN System Variables (continued)**

Name	Dimension	Description
SYS_VPN_VRF_MPLS_INTERFACE_NAME	1	Multiprotocol label switching (MPLS) interface names. Configure VPN VRF settings to generate values for this variable.
SYS_VPN_VRF_NAME	1	VRF names. Configure VPN VRF settings to generate values for this variable.
SYS_VPN_VRF_PROCESS_NUMBER	1	Interior gateway protocol (IGP) process numbers. Configure VPN VRF settings to generate values for this variable.
SYS_VPN_VRF_RD	1	RD values. Configure VPN VRF settings to generate values for this variable.

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
SYS_VPN_VRF_ROUTING_PROTOCOL	1	<p>Interior gateway protocol (IGP) values. The IGP is used for routing the IPsec aggregator toward the Provider Edge (PE)/Multiprotocol Label Switching (MPLS) network.</p> <p>Valid values are STATIC, OSPF, EIGRP, RIPV2, and BGP.</p> <p>Configure VPN VRF settings to generate values for this variable.</p>
SYS_VPN_VRF_SOLUTION	1	<p>Virtual routing and forwarding (VRF) solution. Valid values are 1BOX and 2BOX.</p> <p>Configure VPN VRF settings to generate values for this variable.</p>
<p>CA—Variables related to CA policies. For more information, see Configuring Public Key Infrastructure Policies, page 9-92.</p>		
SYS_VPN_CA_NAME	2	<p>Certificate authority (CA) names.</p> <p>Configure PKI policies to generate values for this variable.</p>
<p>EZVPN—Variables related to EZVPN. For more information, see Overview of Configuring Easy VPN, page 9-112.</p>		

Table 19-8 VPN System Variables (continued)

Name	Dimension	Description
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 9-39.		
SYS_VPN_RTR_WATCH	1	Rtr/watch number. Configure dial backup to generate values for this variable.

Table 19-9 Remote Access System Variables

Name	Dimension	Description
SYS_EZVPN_RA_DYNAMIC_CRYPTOMAP_NAME	1	Dynamic Crypto map name
SYS_EZVPN_RA_DYNAMIC_CRYPTOMAP_SEQ_NUM	1	Dynamic Crypto map number
SYS_EZVPN_RA_PUBLIC_INTERFACE_PIX	2	External interface names (PIX Firewall and ASA devices only).
SYS_EZVPN_RA_STATIC_CRYPTOMAP_NAME	1	Static crypto map names.
SYS_EZVPN_RA_STATIC_CRYPTOMAP_SEQ_NUM	1	Static crypto map numbers.
SYS_IOS_RA_CA_NAME	1	Certificate authority (CA) names (Cisco IOS routers only).
SYS_IOS_RA_PUBLIC_INTERFACE	1	External interface names (Cisco IOS routers only)

Table 19-9 Remote Access System Variables (continued)

Name	Dimension	Description
SYS_IOS_RA_USER_GROUP	1	User group names (Cisco IOS routers only).
SYS_IOS_RA_VRF_NAME	1	Virtual routing and forwarding (VRF) names (Cisco IOS routers only).

Understanding FlexConfig Policies

You can assign FlexConfig policies to devices using either Policy view or Device view. Then, you can deploy configurations containing these policies as you would deploy any configuration generated by Security Manager. For more information about working with policies in general, see [Chapter 6, “Managing Policies.”](#) 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 19-36.](#)

A FlexConfig Creation Scenario

This scenario takes you through the steps to set up Media Gateway Control Protocol (MGCP) for a PIX Firewall 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 a FlexConfig policy object can be used to provide a configuration. This illustrates how the FlexConfig feature enables you to customize, for your network, what is not natively 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 FlexConfig policy objects.

Before You Begin

- Add two PIX Firewalls to Security Manager for this scenario.

Procedure

- Step 1** Duplicate the FlexConfig policy object by doing the following:
- a. Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window](#), page F-3.
 - b. Select **FlexConfigs** from the Policy Object Type selector. The FlexConfig Objects page appears. For more information, see [FlexConfigs Objects Page](#), page P-10
 - c. Right-click ASA_MGCP FlexConfig and select **Create Duplicate**. The Add FlexConfig dialog box appears. For more information, see [FlexConfig Editor Dialog Box](#), page P-11.
 - d. Enter a new name for the new FlexConfig object, for this example MyASA_MGCP.
 - e. Enter a new group name and a description.



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 of FlexConfigs.

- Step 2** Duplicate and edit the \$callAgentList text object by doing the following:

The original ASA_MGCP FlexConfig object uses Policy Object Variable \$callAgentList, 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. From the Policy Object Manager, select **Text Objects** from the Objects list. The Text Objects window appears.
- b. Right-click **callAgentList** and select **Create Duplicate**. 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, then 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, then enter the port number for a call agent in your network. For this example, change the value to 105.
- f. Repeat Steps d and e to change the 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. From the Policy Object Manager, select **FlexConfigs** from the Objects list. The FlexConfigs page appears.
- 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 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.



Note You can select multiple policy objects at one time by holding either the Control (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 it 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 MGCP_Configuration 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 PIX Firewall devices that require MGCP, you can share this policy with them by doing the following:

- a. In Device view, right-click FlexConfigs in the Policy selector, then select **Share Policy**.
The Share Policy dialog box appears.
- b. Enter a name in the Policy Name field and click **OK**. For this example, enter My Shared Policies.
- c. On the main toolbar, click the Policy View button.
- d. From the Policy Types selector, select FlexConfigs. Note the policy type (FlexConfigs) and policy name (My Shared Policies) appear at the top of the page.
- e. Click the **Assignments** tab.
- f. From the Devices selector, navigate to the desired device. For this example, navigate to the other PIX Firewall.
- g. Select the device and click >>.

h. Click **Save**.

Step 7 Deploy the configurations to the devices. For information about deploying configurations, see [Working with Deployment, page 18-36](#).

Configuring FlexConfig Policy Objects

You work with FlexConfig policy objects in much the same manner as other objects in Security Manager. For general information on handling objects, see [Guidelines for Managing Objects, page 8-4](#).

Due to their complexity and interdependency, FlexConfig policy objects are described with FlexConfig policies. For more information, see [Understanding FlexConfig Policy Objects, page 19-2](#). For more information about working with policies in general, see [Chapter 6, “Managing Policies.”](#)

To better understand the steps involved with working with FlexConfig policy objects, from creation through to deployment, see [A FlexConfig Creation Scenario, page 19-36](#).

The following topics describe how to work with FlexConfig policy objects:

- [Creating FlexConfig Policy Objects, page 19-42](#)
- [Duplicating FlexConfig Policy Objects, page 19-43](#)
- [Editing FlexConfig Policy Objects, page 19-45](#)
- [Viewing FlexConfig Policy Objects, page 19-47](#)
- [Generating Usage Reports for FlexConfig Policy Objects, page 19-47](#)
- [Deleting FlexConfig Policy Objects, page 19-49](#)
- [Adding FlexConfig Policy Objects to a Device, page 19-50](#)
- [Removing FlexConfig Policy Objects from a Device, page 19-51](#)
- [Reordering FlexConfig Policy Objects, page 19-52](#)
- [Previewing FlexConfig Policy Objects, page 19-52](#)
- [Deleting FlexConfig Object Variables, page 19-53](#)

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 FlexConfigs, see [Chapter 19, “Managing FlexConfigs.”](#)

**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 8-202](#).

This procedure describes how to create FlexConfig policy objects.

Before You Begin

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

**Note**

Do not use beginning and ending commands to configure interfaces.

Procedure

- Step 1** Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window, page F-3](#).
- Step 2** Select **FlexConfigs** from the Policy Object Type selector. The Policy Object Manager window appears.
- Step 3** Right-click inside the work area, then click **New Object**.
The Add FlexConfig Object dialog box appears. See [Table P-6 on page P-15](#) for a description of the fields in this dialog box.
- Step 4** Enter a name for the new FlexConfig object.
- Step 5** Enter a description for the new FlexConfig object.
- Step 6** (Optional) Assign the new FlexConfig object to a category by selecting an existing group name or by entering a new group name.
- Step 7** 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 configurations.

- Step 8** (Optional) If this FlexConfig object is designed to negate another, enter in the Negate for field the name of the FlexConfig object whose commands are undone by the new FlexConfig object.
- Step 9** In the object body area, enter the commands and instructions to produce the desired configuration file output. You can right-click in the object body field to use the following:
- a. Create Text Object—Allows you to create a variable definition for the FlexConfig object you are creating. For a description of the dialog box that appears, see [Create Text Object Dialog Box, page P-15](#).
 - b. Insert Policy Object—Allows you to choose a policy object type, then select from a list of previously created policy objects.
 - c. Insert System Variable—Allows you to choose a system variable type (Firewall, Remote Access VPN, Router, VPN), then select from a list of predefined variables.
- Step 10** (Optional) Click **Validate FlexConfig** to check the integrity and deployability of the new FlexConfig object.
- Step 11** Click **OK** to save the new FlexConfig object.



Note By default, Security Manager displays a warning if you define an object that matches an existing object. For more information, see [Defining Policy Object Settings, page 2-91](#).

Related Topics

- [FlexConfig Editor Dialog Box, page P-11](#)
- [Understanding the Policy Object Manager Window, page 8-5](#)

Duplicating FlexConfig Policy Objects

You can create policy objects by duplicating an existing object. The new object contains all attributes of the copied object and a default name. You can then modify the name and all attributes as required.

Duplicating is particularly useful for creating objects that are based on predefined objects that cannot be edited.

This procedure describes how to duplicate a FlexConfig object.

Before You Begin

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



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



Note 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.

Procedure

-
- Step 1** Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window, page F-3](#).
- Step 2** Select **FlexConfigs** from the Policy Object Type selector. The Policy Object Manager dialog box appears.
- Step 3** In the work area, right-click the object you want to duplicate, then select **Create Duplicate**.
- The FlexConfig Editor dialog box appears. For a description of the fields in this dialog box, see [FlexConfig Editor Dialog Box, page P-11](#).
- Step 4** Click **OK** to save your changes.



Note By default, Security Manager displays a warning if you define an object that matches an existing object. For more information, see [Defining Policy Object Settings, page 2-91](#).

Related Topics

- [FlexConfig Editor Dialog Box, page P-11](#)
- [Understanding FlexConfig Policy Objects, page 19-2](#)

Editing FlexConfig Policy Objects

You can edit any user-defined FlexConfig object as required. Changes that you make to the object are reflected in all policies that use the object.

**Tip**

You can also edit FlexConfig policy objects when you define policies or objects that use this object type. For more information, see [Selecting Objects for Policies, page 8-202](#).

This procedure describes how to edit a FlexConfig object.

Before You Begin

- Generate a usage report to determine if the object is being used and which policies, objects, and devices would be affected by the changes. See [Generating Usage Reports for FlexConfig Policy Objects, page 19-47](#).
- Ensure that your commands do not conflict in any way with the VPN or firewall configuration on the devices.

**Note**

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

- When editing FlexConfigs involving route-maps (for example, OSPF route-maps, multicast route-maps, and others), the corresponding access control lists (ACLs) must be defined *before* the route-maps. This is a device requirement. If you do not define ACLs *before* route-maps, a deployment error results.

Procedure



Note

The predefined FlexConfig policy objects that are shipped with Security Manager cannot be edited. You can duplicate and rename predefined FlexConfig policy objects and then edit the duplicate. For more information, see [Duplicating FlexConfig Policy Objects, page 19-43](#)

-
- Step 1** Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window, page F-3](#).
- Step 2** Select **FlexConfigs** from the Objects selector. The Policy Object Manager dialog box appears.
- Step 3** In the work area, right-click the object you want to edit, then select **Edit Object**. The FlexConfig Editor dialog box appears. For a description of the fields in this dialog box, see [FlexConfig Editor Dialog Box, page P-11](#).



Tip

You can navigate to the FlexConfig Editor dialog box from a device that contains the FlexConfig object you want to edit. Do this by selecting the device in device view, clicking **FlexConfigs**, selecting a FlexConfig object in the work area, and then clicking **Edit**.

- Step 4** Edit the parameters and body of the FlexConfig policy object, as required for your purpose.
- Step 5** Click **OK** to save your changes.



Note

By default, Security Manager displays a warning if you define an object that matches an existing object. For more information, see [Defining Policy Object Settings, page 2-91](#).

Related Topics

- [FlexConfig Editor Dialog Box, page P-11](#)
- [Understanding FlexConfig Policy Objects, page 19-2](#)

Viewing FlexConfig Policy Objects

You can view detailed object information in read-only mode, even when the object is locked by another activity. This is useful when you need to view complete configuration details for complex objects whose definitions cannot be fully displayed in the Policy Object Manager window, or when your user privileges allow you only to view object information.

**Note**

You can display object details without opening an activity.

This procedure describes how to display complete configuration details for a selected object in read-only mode.

Procedure

-
- Step 1** Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window, page F-3](#).
- Step 2** Select **FlexConfigs** from the Objects selector. The Policy Object Manager dialog box appears.
- Step 3** In the work area, right-click the object that you want to view configuration details for, then select **View Object**.

The FlexConfig Editor dialog box appears in read-only mode. For a description of the fields in this dialog box, see [FlexConfig Editor Dialog Box, page P-11](#).

Related Topics

- [Understanding FlexConfig Policy Objects, page 19-2, page 19-1](#)

Generating Usage Reports for FlexConfig Policy Objects

Before you make any changes, you should determine whether the FlexConfig object is referenced and which policies and devices would be affected by any changes. You can do this by generating a usage report that shows which policies,

objects, and devices are using the selected object. Usage reports contain any references to the selected object in your current activity as well as references found in the data committed to the Security Manager database.

This procedure describes how to generate a usage report.

Procedure

- Step 1** Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window, page F-3](#).
- Step 2** Select **FlexConfigs** from the Objects selector. The Policy Object Manager dialog box appears.
- Step 3** In the work area, right-click the object for which you want to generate a report, then select **Find Usage**.

The usage report appears, displaying all references to the object.



Tip Click a column header to sort the table according to the contents of that column. Click the column header again to sort the table in reverse order.

- Step 4** (Optional) Filter the information displayed in the usage reports by deselecting one or more of the following check boxes:
- Devices
 - Policies
 - Other Objects

The deselected entries are removed from the report.

Related Topics

- [Object Usage Window, page F-564](#)
- [Understanding FlexConfig Policy Objects, page 19-2, page 19-1](#)

Deleting FlexConfig Policy Objects

You can delete only the FlexConfig policy objects that you or others define; you cannot delete the predefined FlexConfig policy objects that are shipped with Security Manager. In addition, you can delete objects only when they are not being referenced by policies or other objects, and when you have the correct permissions.



Note

You might be prevented from deleting an unreferenced object from the database, for example, if you replace a local policy that used the object with a shared policy that does not. If object deletion fails, submit or discard all pending changes (in Workflow mode, submit or discard all pending activities), then try again to delete the objects. Or, you can leave unreferenced objects in the database, because they will not affect Security Manager operation.

This procedure describes how to delete FlexConfig policy objects.

Before You Begin

- Generate a usage report to determine whether the object is referenced and which policies, objects, or devices would be affected by the deletion. See [Generating Usage Reports for FlexConfig Policy Objects, page 19-47](#).
- You need to remove all references to the object before you can delete it.

Procedure

- Step 1** Select **Tools > Policy Object Manager**.
- Step 2** Select **FlexConfigs** from the Objects selector. The Policy Object Manager dialog box appears.
- Step 3** In the work area, right-click the user-defined object, then select **Delete Object**.



Note

You can select multiple objects by pressing **Ctrl** and clicking the desired objects.

- Step 4** When prompted, click **Yes** to confirm the deletion.

- Step 5** To verify that the object was deleted, select **Tools > Audit Report** and view the generated report.
-

Related Topics

- [Understanding FlexConfig Policy Objects, page 19-2, page 19-1](#)
- [Generating the Audit Report, page 20-9](#)

Adding FlexConfig Policy Objects to a Device

This procedure describes how to add existing FlexConfig policy objects to a device and assumes that you are doing so from the Device view.

Before You Begin

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



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

- When creating FlexConfig policy objects involving route-maps (for example, OSPF route-maps, multicast route-maps, and others), the corresponding access control lists (ACLs) must be defined *before* the route-maps. This is a device requirement. If you do not define ACLs *before* route-maps, a deployment error results.

Procedure

Step 1 Select the desired device and click **FlexConfig**. The FlexConfigs page appears.

Step 2 Click **Add**.

The FlexConfigs Selector dialog box appears.



Note To add a new FlexConfig policy object, click **Add**. For details, see [FlexConfig Editor Dialog Box, page P-11](#).

Step 3 Select one or more of the available FlexConfigs and click >> . For descriptions of predefined FlexConfig policy objects, see [Table 19-4](#).

The FlexConfigs appear in the Selected FlexConfigs column.

Step 4 Click **OK**.

The FlexConfigs policy page appears with the FlexConfigs in the prepended or appended field depending on the type defined for each FlexConfig.

Removing FlexConfig Policy Objects from a Device

You might want to remove a FlexConfig policy object from a device if it is no longer used. This procedure describes how to remove FlexConfig policy objects and assumes that you are doing so from the Device view.

For information on deleting a FlexConfig policy object from Security Manager, see [Deleting FlexConfig Policy Objects, page 19-49](#)

Procedure

Step 1 Select the desired device and click **FlexConfig**. The FlexConfigs page appears.

Step 2 Select the FlexConfig policy object you want to remove.

Step 3 Click **Remove**.

Step 4 Click **Yes**.

The FlexConfigs policy page appears with the selected FlexConfigs policy objects removed from the prepended or appended fields.

Reordering FlexConfig Policy Objects

The order of FlexConfig policy objects within the FlexConfig policy affects the way CLI commands are deployed to devices. First prepended FlexConfig policy objects are deployed, then all other policy objects, and finally appended FlexConfig policy objects. In addition, the FlexConfig policy objects in the prepended and appended fields are deployed sequentially.

The order of CLI commands can affect the results that are deployed and implemented. Therefore, make sure to order the FlexConfig policy objects based on dependencies. That is, the one that is used by most FlexConfig policy objects should be put on the top of the list.

This procedure describes how to reorder FlexConfig policy objects and assumes that you are doing so from Device view.

Before You Begin

- When reordering FlexConfigs involving route-maps (for example, OSPF route-maps, multicast route-maps, and others), the corresponding access control lists (ACLs) must be defined *before* the route-maps. This is a device requirement. If you do not define ACLs *before* route-maps, a deployment error results.

Procedure

- Step 1** Select the desired device and click **FlexConfig**. The FlexConfigs page appears.
- Step 2** Select the FlexConfig policy object you want to move.
- Step 3** Click the up arrow or down arrow to move the FlexConfig policy object accordingly.
- Step 4** Click **Save**.
-

Previewing FlexConfig Policy Objects

You can display the CLI commands to be generated by a FlexConfig policy. 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.

This procedure describes how to preview FlexConfig policy objects and assumes that you are doing so from the Device view.

Procedure

-
- Step 1** Select the desired device and click **FlexConfig**. The FlexConfigs page appears.
 - Step 2** Select the FlexConfig policy object you want to preview.
 - Step 3** Click **Preview**. The CLI generated from the selected FlexConfig policy object is displayed.
 - Step 4** Click **Close** when you are done viewing the CLI command.
-

Deleting FlexConfig Object Variables

If you no longer need a FlexConfig object variable, you can remove it from Security Manager.

This procedure show you how to delete a FlexConfig object variable.

Before You Begin

- Determine whether the object is being used and which policies, objects, and devices would be affected by the changes. You can generate a usage report for this purpose. See [Generating Usage Reports for FlexConfig Policy Objects, page 19-47](#).

Procedure

- Step 1** Delete the object variable.
- Select **Tools > Policy Object Manager**. The Policy Object Manager window appears. For more information, see [Policy Object Manager Window, page F-3](#).
 - Select **FlexConfigs** from the Objects selector. The Policy Object Manager dialog box appears.
 - In the work area, right-click the object that contains the variable you want to delete, then select **Edit Object**.
 - The FlexConfig Editor dialog box appears. For a description of the fields in this dialog box, see [FlexConfig Editor Dialog Box, page P-11](#).
 - In the object body, highlight the variable and click the **Delete** key.
 - Click **OK** to save your changes.



Note By default, Security Manager displays a warning if you define an object that matches an existing object. For more information, see [Defining Policy Object Settings, page 2-91](#).

- Step 2** Validate the FlexConfig object.
- From Device view, select the device and click **FlexConfigs** from the Policy selector.
 - Select the FlexConfig object from which you removed the variable.
 - Click **Values**. The Values Assignment dialog box appears.
 - Click **Validate**.
 - Click **OK**.
-