

Chapter 10

Classifying Interfaces and Subscribers with the C-Web Interface

This chapter provides information for configuring and using classification scripts with the C-Web interface.

- To use the SRC CLI, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 6, Classifying Interfaces and Subscribers with the SRC CLI*.
- To use SDX Admin, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 7, Classifying Interfaces and Subscribers on a Solaris Platform*.

Topics in this chapter include:

- Overview of Classification Scripts on page 71
- Classifying Interfaces with the C-Web Interface on page 76
- Classifying Subscribers with the C-Web Interface on page 79
- Classifying DHCP Subscribers with the C-Web Interface on page 84
- Selecting DHCP Parameters on page 87
- Creating DHCP Profiles with the C-Web Interface on page 90

Overview of Classification Scripts

The SAE uses classification scripts to determine whether it manages router interfaces, to select default policies, to find subscriber profiles, and to choose DHCP profiles. The SAE has three classification scripts:

- Interface classification script—When a subscriber's IP interface comes up on the router, the router sends the subscriber's login and interface information to the SAE. The SAE runs the interface classification script to determine whether the SAE manages the interface and if so, what default policies to send to the router.
- Subscriber classification script—If the SAE is managing the interface, the SAE uses the login and interface information that the router sends to run the subscriber classification script to determine which subscriber profile to load into memory.

- DHCP classification script—For DHCP subscribers, the SAE runs DHCP classification scripts to choose DHCP profiles.

How Classification Scripts Work

Classification scripts are organized into rules. Each rule has a *target* and one or more match *conditions*.

- A target is the result of the classification script. For example, the result of subscriber classification scripts is an LDAP search string that is used to find a unique subscriber profile. The result of interface classification scripts is a policy group.
- Conditions are match criteria. The script attempts to match conditions in the script with information sent from the router. For example, match conditions for a subscriber classification script might be login type or domain name. Match conditions for an interface classification script could be interface IP address or interface description.

Each script can have multiple targets, and each target can have multiple conditions. When an object needs classification, the script processes the targets in turn. Within each target, the script processes conditions sequentially. When it finds that the classification conditions for a target match, it returns the target to the SAE. If the script does not find any targets that can be matched, the classifier engine returns a no-match message to the SAE.

Because classification scripts examine conditions sequentially as the conditions appear in the script, you should put more specific conditions at the beginning of the script and less specific conditions at the end of the script.

Interface Classification Scripts

When a subscriber's IP interface comes up on the router, the router sends the subscriber's login and interface information to the SAE. For example, the router might send the following information:

```
IP address=0.0.0.0
Virtual router name=default@erx5_ssp58
Interface name=FastEthernet3/1.1
PPP login name (PPP)=pebbles@virneo.net
User IP address (PPP)=192.168.55.5
Interface speed=100000000
Interface description=P3/1.1
Interface alias=1st pppoe int
RADIUS class=null
```

The SAE invokes the interface classification script and provides to the script the information that it received from the router. The script engine matches the information sent from the router to the conditions in the interface classification script. The script examines each condition in sequential order to find a match.

- If it finds a match, the script processing stops, and the target for that condition is returned to the SAE. The target is the path of a policy group. This policy group is the default policy. The SAE installs the policy on the interface and begins managing the interface.
- If it does not find a match, the script sends a no-match message to the SAE. The SAE does not manage the interface; that is, the policies installed through RADIUS or the CLI remain in effect. The SAE does not install policies.

Subscriber Classification Scripts

When the SAE begins managing an interface, it determines whether a subscriber is associated with the interface by running the subscriber classification script. The SAE also runs the subscriber classification script when certain login events occur. See *SRC-PE Subscribers and Subscriptions Guide, Chapter 3, Subscriber Logins and Service Activation* for a description of login event types.

To find the matching subscriber profile, the SAE uses interface information that it received from the router when the interface became operational (for example, virtual router name, interface name, interface alias). It also uses login information that it received from the router or the portal application when the subscriber attempted to log in (for example, subscriber IP address, login name, or login event type).

When the SAE runs the subscriber classification script, the script engine matches the information sent from the router to the conditions in the subscriber classification script. The script examines each condition in sequential order to find a match.

- If it finds a match, the script processing stops, and the target for the matching condition is returned to the SAE. The target is an LDAP query that uniquely identifies a subscriber profile. The SAE loads the subscriber entry and uses the entry to create a subscriber session in memory.
- If it does not find a match, the script sends a no-match message to the SAE. The SAE does not load a subscriber session onto the interface, and services cannot be activated for this session.

DHCP Classification Scripts

- DHCP classification scripts choose DHCP profiles. See *SRC-PE Subscribers and Subscriptions Guide, Chapter 8, Overview of Plug-Ins Included with the SAE* for information about how DHCP classification scripts are used.

Classification Targets

A target is the result of the classification script that gets returned to the SAE. There are two special types of targets:

- No-match targets—Targets that begin with a - (single dash) are interpreted as no match. If the conditions of this target are matched, a no-match message is returned to SAE. You can use this type of target to exclude certain patterns or to shortcut known nonmatches. To speed up processing, use this target to specify interfaces that you do not want the SAE to manage.
- Script targets—The content of the script rule is interpreted when the classifier is initially loaded. The script rule can contain definitions of custom functions, which can be called during the matching process. Because you can insert arbitrary code into a script, you can use classification scripts to perform arbitrary tasks.

Because script targets use * (asterisks), you cannot use * in other types of targets.

Target Expressions

A target can contain expressions. These expressions can refer to an object in the SAE's memory or configuration, to specific matching conditions, or to another function or script.

Suppose the classification object in a subscriber classifier contains a field called `userName`. The classifier target `uniqueId = <- userName ->` is expanded to contain the actual content of the `userName` field before it is returned to the SAE; for example, for `userName = juser`, `uniqueId = juser` is returned.

Target expressions are enclosed in angle brackets and hyphens; for example, `<-retailerDn->`. The classifier expands expressions before it returns the target to the SAE. The expression is interpreted by an embedded Python interpreter and can contain variables and Python operations. In the simplest case an expression can be a single variable that is replaced with its current contents. Available variable names are all fields of the object passed to the classifier and names created with regular expression matching.

Because a scripting interpreter interprets expressions, more complex operations are possible. Examples are:

- Indexing—`var[index]` returns the element index of a sequence. The first element is at index 0.
- Slicing—`var[start : end]` creates a substring of the variable `var` starting at index `start` to, but not including, index `end`; for example, `var = Hello`, `var[2:4] = ll`

Classification Conditions

You can configure multiple classification conditions for a rule. For example:

```
rule rule-2 {
  target /ent/EntDefault;
  condition {
    "pppLoginName=\"\"";
    "&interfaceName!=\"fastEthernet0*\"";
    "&interfaceName!=\"null*\"";
    "&interfaceName!=\"loopback*\"";
  }
}
```

If you prefix a condition with an & (ampersand) character, the condition is examined only if the previous condition matches.

If you prefix a condition with a | (pipe) character, the condition is examined only if the previous conditions have not produced a positive match.

You can use glob or regular expression matching to configure each target's conditions.

Glob Matching

Glob matches are of the form:

```
field = match
or
field != match
```

where match is a pattern similar to UNIX filename matching. Glob matches are case insensitive. "field != match" is true, if field = match is not true.

- *—Matches any substring.
- ?—Matches any single character.
- [range]—Matches a single character in the specified range. Ranges can have the form a-z or abcd.
- [!range]—Matches a single character outside the specified range.
- C—Matches the single character c.

The available field names are described for the specific classifiers. Examples are:

- interfaceName = fastEthernet3/0 # matches the string "fastEthernet3/0" directly.
- interfaceName = fast*3/1 # matches any string that starts with "fast" and ends with "3/1"
- interfaceName = fast*3/1.* # starts with "fast", contains "3/1." arbitrary ending
- interfaceName = fast*3/[2-57] # starts with "fast", contains "3/" followed by 2,3,4,5 or 7

Regular Expression Matching

Regular expression matches are of the form:

```
field =~ re
or
field !~ re
```

where `field !~ re` is true if `field =~ re` is not true. The regular expression is *re*. For a complete description of the syntax, see:

<http://www.python.org/doc/2.0/lib/re-syntax.html>

You can group regular expressions with pairs of parentheses. If such an expression matches, the contents of the groups are made available for target expressions. Group number *n* is available as `G[n]`, where *n* is the number of the opening parenthesis of the group. You can also name groups by using the special notation `(?P<name>...)`.

Examples:

```
ifAlias =~ "SSP(.*)"
# match a string starting with "SSP". The remainder is stored
# in the variable "G[1]"

ifAlias =~ (?P<dn>name=(?P<name>[^\,]*)).*
# match a string starting with "name=". The whole match is
# stored in the variable "dn". A submatch which does not
# contain any ","-characters and starts after "name="
# is stored in variable "name"
```

Classifying Interfaces with the C-Web Interface

To define interface classification scripts:

1. Click **Configure**, and expand **Shared > Network**.
2. Expand the device for which you want to configure interface classification scripts, and then click **Interface Classifier**.

The Interface Classifier pane appears.

3. From the Create new list, select **Rule**.
4. Type a name for the new rule in the dialog box, and click **OK**.

The rule appears in the side pane and the Rule pane.

5. Enter a script or a target as described in the Help text in the Main pane, and click **Apply**.

6. To configure a condition for a target:
 - a. Expand the rule in the side pane, and click **Condition**.
The Condition pane appears.
 - b. From the Create new list, select **Condition**.
 - c. Type the interface classification condition name as described in Interface Classification Conditions, and click **OK**.

The condition appears in the side pane and the Condition pane.

Interface Classification Conditions

Use the fields in this section to define interface classification conditions.

broadcastAddr

- Interface broadcast address.
- Value—Valid broadcast address format
- Example—broadcastAddr.hostAddress = “255.255.255.255”

ifAlias

- Description of an interface.
- Value—Interface description that is configured on the router. For JUNOSe routers, it is the description configured with the **interface description** command.
- Example—ifAlias = “1st pppoe int”

ifDesc

- Alternate name of the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOSe router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, ifDesc is the same as interfaceName.
- Example—ifDesc = “IP3/1.1”

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JUNOS routers: interfaceName = “fastethernet6/0.1”
 For JUNOS routing platforms: interfaceName = “fe-0/1/0.0”
 For forwarding interface: interfaceName = “FORWARDING_INTERFACE”

ipAddress

- Interface IP address.
- Value—Valid IPv4 IP address format
- Example—ipAddress = “10.10.30.1”

ipMask

- Interface network mask.
- Value—Valid IPv4 IP network mask format
- Example—ipMask = “255.255.255.255”

mtu

- Maximum transfer unit configured on the interface.
- Value—32-integer value
- Example—mtu = “1492”

nasPortId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPortId = “fastEthernet 3/1” (There is a space between fastEthernet and slot number 3/1 in the nasPortId.)

pppLoginName

- Login name for PPP subscribers.
- Value—Login name in the format username@domain
- Example—pppLoginName = “pebbles@virneo.net”

radiusClass

- RADIUS class attribute.
- Value—RADIUS class name
- Example—radiusClass = “Premium”

serviceBundle

- Content of the vendor-specific RADIUS attribute for the service bundle.
- Value—Name of a service bundle

userIpAddress

- Subscriber IP address (PPP only).
- Value—valid IPv4 address
- Example—userIpAddress = “192.168.30.15”

virtualRouterName

- Name of the virtual router or routing instance.
- Value—For JUNOS routers: name of the virtual router in the format `vrname@hostname`
For JUNOS routing platforms: name of the routing instance
- Example—virtualRouterName = “default@erx5”

Classifying Subscribers with the C-Web Interface

To define subscriber classification scripts:

1. Click **Configure**, expand **Shared > SAE**, and then click **Subscriber Classifier**.

The Subscriber Classifier pane appears.

2. From the Create new list, select **Rule**.
3. Type a name for the new rule in the dialog box, and click **OK**.

The rule appears in the side pane and the Rule pane.

4. Enter a script or a target as described in the Help text in the Main pane, and click **OK**.

5. To configure a condition for a target:

- a. Expand the rule in the side pane, and click **Condition**.

The Condition pane appears.

- b. Type the subscriber classification condition name as described in Subscriber Classification Conditions, and click **OK**.

The condition appears in the side pane and the Condition pane.

Subscriber Classification Conditions

Subscriber classification conditions define match criteria that are used to find the subscriber profile. Use the fields in this section to define subscriber classification conditions.

dhcp

- DHCP options. See *Sending DHCP Options to the JUNOS Router* on page 82.

domainName

- Domain name of the subscriber.
- Value—Valid domain name
- Example—domainName = “isp99.com”

ifAlias

- Description of the interface.
- Value—Interface description that is configured on the router. For JUNOS routers, it is the description configured with the **interface description** command
- Example—ifAlias = “dhcp-subscriber12”

ifDesc

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOS router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, ifDesc is the same as interfaceName.
- Example—ifDesc = “IP3/1.1”

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
 - Router for a JUNOS router instance
- Example—For JUNOS routers: interfaceName = “fastEthernet6/0”
For JUNOS routing platforms: interfaceName = “fe-0/1/0.0”
For forwarding interface: interfaceName = “FORWARDING_INTERFACE”

loginName

- Name to be used to create a loginName attribute for a subscriber session for JUNOS interfaces that are not otherwise assigned a loginName when a session starts, such as unauthenticated DHCP addresses, unauthenticated IP interfaces (that are not using PPP connections), or core-facing interfaces.

The loginName can also be used to identify a subscriber session through the SAE CORBA remote API.
- Value—Name in the form subscriber@domain

- < Login name >
- Guideline—The format is not defined. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the operator.
- Example—idp@idp

loginType

- Type of subscriber session to be created.
- Value—One of the following login types:
 - ASSIGNEDIP—For assigned IP subscribers. Triggered when an application accesses a subscriber object for an assigned IP subscriber that is not currently loaded into memory. (Supported on JUNOS routers.)
 - AUTHINTF—For authenticated interface login requests. Triggered when a login Name is reported together with the interface, such as authenticated PPP or autoconfigured ATM interface, by means of the **subscriber** command. (Supported on JUNOS routers.)
 - INTF—For unauthenticated interface login requests. Triggered when an interface comes up and the interface classification script determines that the SAE should manage the interface. (Supported on JUNOS routing platforms and JUNOS routers.)
 - ADDR—For unauthenticated address login requests. Triggered when the DHCP server in the JUNOS router provides an unauthenticated IP address. (Supported on JUNOS routers.)
 - AUTHADDR—For authenticated address login requests. Triggered when the DHCP server in the JUNOS router provides an authenticated IP address. (Supported on JUNOS routers.)
 - PORTAL—Triggered when the portal API is invoked to log in a subscriber. (Supported on JUNOS routing platforms and JUNOS routers.)
- Example—loginType = "AUTHADDR"

macAddress

- String representation of the DHCP subscriber media access control (MAC) address.
- Value—Valid MAC address
- Example—macAddress = "00:11:22:33:44:55"

nasPortId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPortId = "fastEthernet 3/1" (There is a space between fastEthernet and slot number 3/1 in the nasPortId.)

radiusClass

- RADIUS class used for authorization.
- Value—RADIUS class name
- Example—radiusClass = “Premium”

retailerDn

- DN of the retailer object. The object is found when the domain name is mapped to a retailer object in LDAP.
- Value—DN of a retailer

serviceBundle

- Content of the vendor-specific RADIUS attribute for the service bundle.
- Value—Name of a service bundle
- Example—serviceBundle = “goldSubscriber”

unauthenticatedUserDn

- DN of the unauthenticated subscriber profile (usable for target expressions only).
- Value—DN of a subscriber profile

userName

- Name of the subscriber.
- Value—Subscriber name without the domain name
- Example—userName = “peter”

virtualRouterName

- Name of the virtual router or routing instance.
- Value—For JUNOS routers: name of the virtual router in the format `vrname@hostname`
For JUNOS routing platforms: name of the routing instance
- Example—virtualRouterName = “default@e_series5”

Sending DHCP Options to the JUNOS Router

Subscriber classification scripts support DHCP options conveyed through COPS. When COPS reports an address, the JUNOS router sends DHCP options received for DHCP requests for that address. The DHCP options are available in the subscriber classification context for selecting the subscriber profile to load.

The fields in Table 5 are in the classification context of subscriber classification scripts.

Table 5: DHCP Options in UserClassificationContext Field

DHCP Option	UserClassificationContext Field	Comments
giAddr	dhcp.giAddr	Relay agent gateway address
Option 82 data	dhcp.getOption(82)	Content is accessible with getSubOptions()
Client ID	dhcp.getOption(61).getString()	
Lease time	dhcp.getOption(51).getInt()	
Client requested parameter list	dhcp.getOption(55).getBytes()	
Domain name sent to client	dhcp.getOption(12).getString() dhcp.getOption(15).getString()	12 = HostName 15 = DomainName
DNS server address(es) sent to client	dhcp.getOption(6).getIpAddresses()	
Subnet mask	dhcp.getOption(1).getIpAddress()	
NetBios name server address(es) sent to client	dhcp.getOption(44).getIpAddresses()	
NetBios node type	dhcp.getOption(46).getBytes()	
Default router address(es) sent to client	dhcp.getOption(3).getIpAddresses()	

The DHCP options are accessible to the subscriber classification script with the following syntax:

```

dhcp.giAddr = "match"

# interpret option 61 as string
dhcp[61].string = "match"

# interpret option 1 (subnet) as dotted decimal IP
dhcp[1].ipAddress = "match"

# option 82, suboption 1, interpreted as string
dhcp[82].subOptions[1].string = "match"

```

The received DHCP options are also stored in the UserSession and are available through the portal API (method User.getDhcpOptions).

Subscriber Classification Targets

The target of the subscriber classification script is an LDAP search string. The search string uses a syntax similar to an LDAP URL (see RFC 2255—The LDAP URL Format (December 1997)). The syntax is:

```
"baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]"
```

- **baseDN**—Distinguished name of object where the LDAP search starts
- **attributes**—Can be used to override attributes in the loaded LDAP object. For example, for static IP subscribers the SAE must learn the IP address assigned to a particular subscriber. This address is defined in the `ipAddress` attribute of the subscriber profile. A target of the form `baseDN?ipAddress = <-function(interfaceName)->` invokes function after the subscriber profile is loaded from LDAP and sets the IP address to the return value of function. The function is defined in the subscriber classification script, and can be used for a variety of things; for example, to query an external database.



NOTE: You can use subscriber classification to override only the `ipAddress`, `loginName`, or `accountingId` attributes. If you configure values to override other attributes, the value is lost when the SAE recovers from a network or server failure.

- **scope**—Scope of search
 - **base**—Is the default, searches the base DN only.
 - **one**—Searches the direct children of the base DN.
 - **sub**—Searches the complete subtree below the base DN.
- **filter**—Is an RFC 2254-style LDAP search filter expression; for example, `(uniqueId = <-userName->)`. See RFC 2254—The String Representation of LDAP Search Filters (December 1997).

With the exception of `baseDN` all the fields are optional.

The result of the LDAP search must be exactly one directory object. If no object or more than one object is found, the subscriber session is terminated.

Classifying DHCP Subscribers with the C-Web Interface

A classification script can contain either a target and a condition or a script. If you do not define a script, the classifier must have both a target and a condition.

To configure DHCP classification scripts:

1. Click **Configure**, expand **Shared > SAE**, and then click **DHCP Classifier**.

The Dhcp Classifier pane appears.

2. From the Create new list, select **Rule**.
3. Type a name for the new rule in the dialog box, and click **OK**.

The rule appears in the side pane and the Rule pane.

4. Enter a script or a target as described in the Help text in the Main pane, and click **OK**.

5. To configure a condition for a target,
 - a. Expand the rule in the side pane, and click Condition.

The Condition pane appears.

- b. Type the DHCP classification condition name as described in DHCP Classification Conditions, and click **OK**.

The condition appears in the side pane and the Condition pane.

DHCP Classification Conditions

DHCP classification conditions define match criteria that are used to find the DHCP profile. Use the fields in this section to define DHCP classification conditions.

authVirtualRouterName

- Name of JUNOS virtual router that is set by an authorization plug-in through the authorization response.
- Value—Name of the virtual router in the format `vrname@hostname`

dhcp

- DHCP options. See *Setting DHCP Parameters with DHCP Options* on page 87.

dhcpProfileDN

- Search base for DHCP profiles. The DN can be used in target expressions.
- Value—DN of DHCP profile

interfaceName

- Name of the interface where the DHCP discover message was received.
- Value—Name of the interface in your router CLI syntax
- Example—`interfaceName = fastEthernet6/0`

ifAlias

- Description of the interface where the DHCP discover request was received.
- Value—Interface description that is configured on the router. For JUNOS routers, it is the description configured with the **interface description** command
- Example—`ifAlias = "dhcp-subscriber12"`

ifDesc

- Alternate name for the interface where the DHCP discover request was received. This is a system-generated name that is used by SNMP.
- Value
 - On a JUNOSe router, the format of the description is:
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, ifDesc is the same as interfaceName.

macAddress

- MAC address of the DHCP client that appears in DHCP request.
- Value—Valid MAC address
- Example—macAddress = “00:11:22:33:44:55”

nasPortId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPortId = “fastEthernet 3/1” (There is a space between fastEthernet and slot number 3/1 in the nasPortId.)

poolName

- IP address pool name that is set by an authorization plug-in through the authorization response.
- Value—Name of an address pool configured on the JUNOSe router

virtualRouterName

- Name of the virtual router.
- Value—Name of the virtual router in the format vrname@hostname

DHCP Classification Targets

The target of the DHCP classification script uses a syntax similar to an LDAP URL. With the exception of baseDN, all fields are optional. The syntax is:

```
baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]
```

- baseDN—DN of object where search starts.
- attributes—Comma-separated list of properties, in the format attribute = <-value->, that allow you to set specific attributes for directory objects that the script finds; see *DHCP Classification Conditions* on page 85.

You can use the attribute configuration to override attributes in the directory. For example, to override the IP pool name that is stored in the DHCP profile with the pool name that the authorization plug-in sends, use the attribute statement radiusFramedPool = <-poolName->.

- scope—Scope of search in the directory
 - base—Searches the base DN only; default scope
 - one—Searches the direct subordinates of the base DN (one-level search)
 - sub—Searches all objects subordinate to the base DN
- filter—An RFC 2254-style LDAP search filter expression; for example, (uniqueId = <-userName->). See RFC 2254—The String Representation of LDAP Search Filters (December 1997).

Selecting DHCP Parameters

The SAE sends a set of parameters to the DHCP server in the JUNOS router. The DHCP server determines the IP address offered, as well as the options sent to the DHCP client. The parameters comprise IP address authorization parameters, as well as parameters stored in a DHCP profile. Parameters in the DHCP profile override authorization parameters.

For more information about how the SAE handles DHCP subscribers, see:

- *SRC-PE Subscribers and Subscriptions Guide, Chapter 8, Overview of Plug-Ins Included with the SAE*
- *SRC-PE Subscribers and Subscriptions Guide, Chapter 3, Subscriber Logins and Service Activation*

Setting DHCP Parameters with DHCP Options



NOTE: JUNOS routers do not currently support the functionality described in this section. DHCP options and BOOTP options that the SAE sends to the JUNOS router are ignored.

DHCP servers use DHCP options to configure DHCP clients. The DHCP local server in the JUNOS router supports a subset of DHCP options. The SAE supports all DHCP options defined in RFC 2132—DHCP Options and BOOTP Vendor Extensions (March 1997) by name. It also supports other options, but you need to specify them by number and type. The DHCP options allow a flexible definition of parameters offered to DHCP subscribers. For example, they allow integration with cable modems or set-top boxes because you can configure options to control the boot sequence of these devices.

You can configure DHCP options in DHCP profiles and in DHCP classification scripts. Table 6 on page 88 lists the name, number, and type of all supported DHCP options. You can use these fields to configure DHCP options.

The following example shows how to specify an option by number and by type. The two statements identify the same option:

```
dhcp[12]

dhcp['host-name']
```

In SDX software earlier than Release 4.2, you had to include the option type in your option definition. For example:

```
dhcp[12].string = HOST
```

You can now write:

```
dhcp[12] = HOST
```

Note that the earlier method of defining options still works in Release 4.2 and later.

Table 6: DHCP Options Supported on the SAE

Option Name	Option Number	Option Type
subnet-mask	1	ip-address
time-offset	2	int32
routers	3	ip-address
time-servers	4	ip-address
ien116-name-servers	5	ip-address
domain-name-servers	6	ip-address
log-servers	7	ip-address
cookie-servers	8	ip-address
lpr-servers	9	ip-address
impress-servers	10	ip-address
resource-location-servers	11	ip-address
host-name	12	string
boot-size	13	int16
merit-dump	14	string
domain-name	15	string
swap-server	16	ip-address
root-path	17	string
extension-path	18	string
ip-forwarding	19	int8
non-local-source-routing	20	int8
policy-filter	21	ip-address
max-dgram-reassembly	22	int16
default-ip-ttl	23	int8
path-mtu-aging-timeout	24	int32
path-mtu-plateau-table	25	int16
interface-mtu	26	int16
all-subnets-local	27	int8
broadcast-address	28	ip-address
perform-mask-discovery	29	int8

Table 6: DHCP Options Supported on the SAE (continued)

Option Name	Option Number	Option Type
mask-supplier	30	int8
router-discovery	31	int8
router-solicitation-address	32	ip-address
static-routes	33	ip-address
trailer-encapsulation	34	int8
arp-cache-timeout	35	int32
ieee802-3-encapsulation	36	int8
default-tcp-ttl	37	int8
tcp-keepalive-interval	38	int32
tcp-keepalive-garbage	39	int8
nis-domain	40	string
nis-servers	41	ip-address
ntp-servers	42	ip-address
netbios-name-servers	44	ip-address
netbios-dd-server	45	ip-address
netbios-node-type	46	int8
netbios-scope	47	string
font-servers	48	ip-address
x-display-manager	49	ip-address
requested-ip-address	50	ip-address
ip-address-lease-time	51	int32
option-overload	52	int8
dhcp-msg-type	53	int8
server-identifier	54	ip-address
parameter-request-list	55	data-string
message	56	string
maximum-dhcp-msg-size	57	int16
renewal-time	58	int32
rebinding-time	59	int32
vendor-class-identifier	60	data-string
client-identifier	61	data-string
nisplus-domain	64	string
nisplus-servers	65	ip-address
tftp-server-name	66	string
bootfile-name	67	string
mobile-ip-home-agent	68	ip-address
smtp-server	69	ip-address
pop-server	70	ip-address

Table 6: DHCP Options Supported on the SAE (continued)

Option Name	Option Number	Option Type
nntp-server	71	ip-address
www-server	72	ip-address
finger-server	73	ip-address
irc-server	74	ip-address
streettalk-server	75	ip-address
streettalk-directory-assistance-server	76	ip-address

Creating DHCP Profiles with the C-Web Interface

When the SAE receives a DHCP discover request from the router, it uses the client's MAC address to find a DHCP profile in cache or in the directory. If it finds a DHCP profile, the SAE uses the information in the profile to create a discover decision that it returns to the router. The discover decision includes information to select an IP address and DHCP options to configure the DHCP client.

When a DHCP subscriber logs in to the SAE through a Web portal, the SAE registers the subscriber's equipment and creates a cached DHCP profile in the *o = AuthCache* directory. These profiles are keyed by the MAC address of the DHCP client device. They are created by the `grantPublicIp` or the `registerEquipment` methods.

You can also create DHCP profiles manually with SDX Admin or by adding DHCP profile entries to the directory. DHCP profiles are stored in the *o = AuthCache* directory in the `dhcpProfile` object class. The `dhcpProfile` object class is subordinate to the `cachedAuthenticationProfiles` object class. Manually created profiles are keyed by the `cn` (common name) attribute.

For more information about how the SAE handles DHCP subscribers, see:

- *SRC-PE Subscribers and Subscriptions Guide, Chapter 8, Overview of Plug-Ins Included with the SAE*
- *SRC-PE Subscribers and Subscriptions Guide, Chapter 3, Subscriber Logins and Service Activation*

To create a DHCP profile:

1. Click **Configure**, expand **Shared**, and then click **Auth Cache**.
The Auth Cache pane appears.
2. From the Create new list, select **Cached Dhcp Profile**.
3. Type a name for the new cached DHCP profile in the dialog box, and click **OK**.
The cached authentication profile appears in the side pane and in the Cached DHCP Profile pane.
4. Enter information as described in the Help text in the main pane, and click **Apply**.