# ACS Ruledef Configuration Mode Commands

**Important** In 14.1 and earlier releases, up to 10 rule expressions can be configured in one ruledef. In 15.0 and later releases, up to 32 rule expressions can be configured in one ruledef.

## Command Modes

The ACS Ruledef Configuration Mode is used to create and manage rule expressions in individual rule definitions (ruledefs).

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Important** The commands or keywords/variables that are available are dependent on platform type, product version, and installed license(s).


- bearer 3gpp apn, on page 8
- bearer 3gpp imsi, on page 9
- bearer 3gpp rat-type, on page 10
- bearer 3gpp sgsn-address, on page 11
- bearer 3gpp2 bsid, on page 12
- bearer 3gpp2 service-option, on page 14
- bearer apn, on page 15
- bearer imsi, on page 16
- bearer rat-type, on page 17
- bearer sgsn-address, on page 18
- bearer traffic-group, on page 19
- cca quota-state, on page 20
- cca redirect-indicator, on page 21
- copy-packet-to-log, on page 22
- description, on page 23
- dns answer-name, on page 23

- dns any-match, on page 25
- dns previous-state, on page 26
- dns query-name, on page 27
- dns query-type, on page 28
- dns return-code, on page 29
- dns state, on page 30
- dns tid, on page 31
- email, on page 32
- end, on page 34
- exit, on page 35
- file-transfer any-match, on page 35
- file-transfer chunk-number, on page 36
- file-transfer current-chunk-length, on page 37
- file-transfer declared-chunk-length, on page 38
- file-transfer declared-file-size, on page 39
- file-transfer filename, on page 40
- file-transfer previous-state, on page 41
- file-transfer state, on page 42
- file-transfer transferred-file-size, on page 43
- ftp any-match, on page 44
- ftp client-ip-address, on page 45
- ftp client-port, on page 46
- ftp command args, on page 47
- ftp command id, on page 48
- ftp command name, on page 49
- ftp connection-type, on page 51
- ftp data-any-match, on page 52
- ftp filename, on page 53
- ftp pdu-length, on page 54
- ftp pdu-type, on page 55
- ftp previous-state, on page 56
- ftp reply code, on page 57
- ftp server-ip-address, on page 58
- ftp server-port, on page 59
- ftp session-length, on page 60
- ftp state, on page 61
- ftp url, on page 62
- ftp user, on page 63
- http accept, on page 64
- http any-match, on page 65
- http attribute-in-data, on page 66
- http attribute-in-url, on page 67
- http content disposition, on page 68
- http content length, on page 70
- http content range, on page 71
- http content type, on page 71

- http cookie, on page 72
- http domain, on page 74
- http error, on page 75
- http first-request-packet, on page 76
- http header-length, on page 77
- http host, on page 78
- http payload-length, on page 81
- http pdu-length, on page 82
- http previous-state, on page 83
- http referer, on page 84
- http reply code, on page 87
- http reply payload, on page 88
- http request method, on page 88
- http session-length, on page 90
- http state, on page 91
- http transaction-length, on page 92
- http transfer-encoding, on page 93
- http uri, on page 94
- http url, on page 97
- http user-agent, on page 100
- http version, on page 101
- http x-header, on page 102
- icmp any-match, on page 103
- icmp code, on page 104
- icmp type, on page 105
- icmpv6 any-match, on page 106
- icmpv6 code, on page 107
- icmpv6 type, on page 108
- if-protocol, on page 109
- imap any-match, on page 110
- imap cc, on page 111
- imap command, on page 113
- imap content class, on page 114
- imap content type, on page 116
- imap date, on page 117
- imap final-reply, on page 118
- imap from, on page 119
- imap mail-size, on page 120
- imap mailbox-size, on page 121
- imap message-type, on page 122
- imap previous-state, on page 123
- imap session-length, on page 124
- imap session-previous-state, on page 125
- imap session-state, on page 126
- imap state, on page 127
- imap subject, on page 128

- imap to, on page 129
- ip any-match, on page 130
- ip dscp, on page 131
- ip downlink, on page 132
- ip dst-address, on page 133
- ip error, on page 135
- ip protocol, on page 136
- ip server-domain-name, on page 137
- ip server-ip-address, on page 138
- ip src-address, on page 140
- ip subscriber-ip-address, on page 141
- ip total-length, on page 143
- ip uplink, on page 144
- ip version, on page 145
- mms any-match, on page 146
- mms bcc, on page 147
- mms cc, on page 148
- mms content location, on page 149
- mms content type, on page 150
- mms downlink, on page 151
- mms from, on page 152
- mms message-id, on page 153
- mms pdu-type, on page 155
- mms previous-state, on page 156
- mms response status, on page 157
- mms state, on page 158
- mms status, on page 159
- mms subject, on page 160
- mms tid, on page 161
- mms to, on page 163
- mms uplink, on page 164
- mms version, on page 165
- multi-line-or all-lines, on page 166
- p2p any-match, on page 166
- p2p app-identifier, on page 167
- p2p behavioral, on page 169
- p2p protocol, on page 170
- p2p protocol-group, on page 182
- p2p set-app-proto, on page 184
- p2p traffic-type, on page 185
- pop3 any-match, on page 186
- pop3 command args, on page 187
- pop3 command id, on page 188
- pop3 command name, on page 189
- pop3 mail-size, on page 190
- pop3 pdu-length, on page 191

- pop3 pdu-type, on page 192
- pop3 previous-state, on page 193
- pop3 reply args, on page 195
- pop3 reply id, on page 196
- pop3 reply status, on page 197
- pop3 session-length, on page 198
- pop3 state, on page 199
- pop3 user-name, on page 200
- pptp any-match, on page 201
- pptp ctrl-msg-type, on page 202
- pptp gre any-match, on page 203
- radius any-match, on page 204
- radius error, on page 205
- radius state, on page 206
- rtcp any-match, on page 207
- rtcp jitter, on page 208
- rtcp parent-proto, on page 209
- rtcp pdu-length, on page 210
- rtcp rtsp-id, on page 211
- rtcp session-length, on page 212
- rtcp uri, on page 213
- rtp any-match, on page 214
- rtp parent-proto, on page 215
- rtp pdu-length, on page 216
- rtp rtsp-id, on page 217
- rtp session-length, on page 218
- rtp uri, on page 219
- rtsp any-match, on page 220
- rtsp content length, on page 221
- rtsp content type, on page 222
- rtsp date, on page 223
- rtsp previous-state, on page 225
- rtsp reply code, on page 226
- rtsp request method, on page 227
- rtsp request packet, on page 228
- rtsp rtp-seq, on page 229
- rtsp rtp-time, on page 230
- rtsp rtp-uri, on page 231
- rtsp session-id, on page 232
- rtsp session-length, on page 233
- rtsp state, on page 234
- rtsp uri, on page 235
- rtsp uri sub-part, on page 238
- rtsp user-agent, on page 240
- rtsp-stream any-match, on page 241
- rtsp-stream first-setup-url, on page 242

- rule-application, on page 244
- sdp any-match, on page 246
- sdp connection-ip-address, on page 247
- sdp media-audio-port, on page 247
- sdp media-video-port, on page 248
- sdp uplink, on page 249
- secure-http any-match, on page 250
- secure-http uplink, on page 251
- sip any-match, on page 252
- sip call-id, on page 253
- sip content length, on page 254
- sip content type, on page 255
- sip from, on page 256
- sip previous-state, on page 257
- sip reply code, on page 259
- sip request method, on page 260
- sip request packet, on page 261
- sip state, on page 262
- sip to, on page 263
- sip uri, on page 264
- smtp any-match, on page 266
- smtp command arguments, on page 267
- smtp command id, on page 268
- smtp command name, on page 269
- smtp mail-size, on page 270
- smtp pdu-length, on page 271
- smtp previous-state, on page 272
- smtp recipient, on page 273
- smtp reply arguments, on page 274
- smtp reply id, on page 276
- smtp reply status, on page 277
- smtp sender, on page 278
- smtp session-length, on page 279
- smtp state, on page 280
- tcp analyzed out-of-order, on page 281
- tcp any-match, on page 282
- tcp client-port, on page 283
- tcp connection-initiator, on page 284
- tcp downlink, on page 285
- tcp dst-port, on page 286
- tcp duplicate, on page 287
- tcp either-port, on page 288
- tcp error, on page 290
- tcp flag, on page 291
- tcp initial-handshake-lost, on page 292
- tcp payload, on page 293

- tcp payload-length, on page 294
- tcp previous-state, on page 295
- tcp proxy-prev-state, on page 296
- tcp proxy-state, on page 297
- tcp server-port, on page 299
- tcp session-length, on page 300
- tcp src-port, on page 301
- tcp state, on page 303
- tcp uplink, on page 304
- tethering-detection, on page 305
- tftp any-match, on page 306
- tftp data-any-match, on page 307
- tls, on page 308
- udp any-match, on page 309
- udp client-port, on page 310
- udp downlink, on page 311
- udp dst-port, on page 312
- udp either-port, on page 313
- udp payload starts-with, on page 315
- udp server-port, on page 316
- udp src-port, on page 317
- udp uplink, on page 318
- wsp any-match, on page 319
- wsp content type, on page 320
- wsp domain, on page 321
- wsp downlink, on page 323
- wsp first-request-packet, on page 324
- wsp host, on page 325
- wsp pdu-length, on page 326
- wsp pdu-type, on page 327
- wsp previous-state, on page 328
- wsp reply code, on page 329
- wsp session-length, on page 330
- wsp session-management, on page 331
- wsp state, on page 332
- wsp status, on page 333
- wsp tid, on page 334
- wsp total-length, on page 334
- wsp transfer-encoding, on page 335
- wsp uplink, on page 336
- wsp url, on page 337
- wsp user-agent, on page 339
- wsp x-header, on page 340
- wtp any-match, on page 342
- wtp downlink, on page 343
- wtp gtr, on page 344

- wtp pdu-length, on page 345
- wtp pdu-type, on page 345
- wtp previous-state, on page 347
- wtp rid, on page 348
- wtp state, on page 349
- wtp tid, on page 350
- wtp transaction class, on page 351
- wtp ttr, on page 352
- wtp uplink, on page 353
- www any-match, on page 354
- www content type, on page 355
- www domain, on page 356
- www downlink, on page 357
- www first-request-packet, on page 358
- www header-length, on page 359
- www host, on page 360
- www payload-length, on page 361
- www pdu-length, on page 362
- www previous-state, on page 363
- www reply code, on page 364
- www state, on page 365
- www transfer-encoding, on page 366
- www url, on page 367


# bearer 3gpp apn

This command allows you to define rule expressions to match Access Point Name (APN) of the bearer flow.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer 3gpp apn [ case-sensitive ]** *operator apn_name*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***apn_name***

Specifies name of the APN to match.

*apn_name* must be an alphanumeric string of 1 through 62 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match an APN in the bearer flow.

**Example**

The following command defines a rule expression to match user traffic based on APN named *apn12*:

```
bearer 3gpp = apn12
```

# bearer 3gpp imsi

This command allows you to define rule expressions to match International Mobile Station Identification (IMSI) number in the bearer flow.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer 3gpp imsi {** *operator imsi* **| { !range | range } imsi-pool** *imsi_pool_name* **}**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***imsi***

Specifies the IMSI number to match.

**!range | range**

**!range | range**: Specifies the range criteria:

- **!range**: Not in the range of
- **range**: In the range of

**imsi-pool *imsi_pool_name***

Specifies the IMSI pool.

*imsi_pool_name* must be the name of an IMSI pool, and must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match an IMSI.

**Example**

The following command defines a rule expression to analyze user traffic for the IMSI number *9198838330912*:

```
bearer 3gpp imsi = 9198838330912
```

# bearer 3gpp rat-type

This command allows you to define rule expressions to match Radio Access Technology (RAT) in the bearer flow.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] bearer 3gpp rat-type`** *`operator rat_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***rat_type***

Specifies the RAT type to match.

*rat_type* must be one of the following:

- **geran**: GSM EDGE Radio Access Network type
- **utran**: UMTS Terrestrial Radio Access Network type
- **wlan**: Wireless LAN type

**Usage Guidelines**

Use this command to define rule expressions to match a RAT type.

**Example**

The following command defines a rule expression to match user traffic based on RAT type **wlan**:

```
bearer 3gpp rat-type = wlan
```

# bearer 3gpp sgsn-address

This command allows you to define rule expressions to match SGSN address associated in the bearer flow.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer 3gpp sgsn-address** *operator ipv4/ipv6_address*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***ipv4/ipv6_address***

Specifies the SGSN IP address to match.

*ipv4/ipv6_address* must be in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

**Usage Guidelines**

Use this command to define rule expressions to match IP address of an SGSN node. This command replaces the **bearer sgsn-address** command.

**Example**

The following command defines a rule expression to analyze user traffic for an SGSN node with IP address *10.1.1.1*:

```
bearer 3gpp sgsn-address = 10.1.1.1
```

# bearer 3gpp2 bsid

This command allows you to define rule expressions to match Base Station Identifier (BSID) associated with the bearer.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer 3gpp2 bsid [ case-sensitive ] [ use-group-of-objects ]** *operator string*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### use-group-of-objects

Specifies using a group-of-objects as a qualifier to match this rule.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

#### *string*

Specifies the name of a group-of-objects to match.

If the **use-group-of-objects** keyword is not included in the command, *string* specifies name of the matching 3GPP2 service Base Station ID (BSID) in bearer flow.

If the **use-group-of-objects** keyword is included in the command, *string* must be the name of the group-of-objects to use. In this case, it is checked if the rule is satisfied for either one or none of the objects in the group-of-objects depending upon the operator used. For example, if the *operator* is **contains**, the expression would be true if any of the objects in the specified object group is contained in the BSID. If the *operator* is **!contains**, then the expression would be true if none of the objects in the object group is contained in the BSID.

*string* must be an alphanumeric string of 1 through 16 characters, and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match a 3GPP2 Base Station Identifier (BSID).

### Example

The following command defines a rule expression to analyze user traffic for 3GPP2 BSID named *bs001_xyz*:

```
bearer 3gpp2 bsid = bs001_xyz
```

# bearer 3gpp2 service-option

This command allows you to define rule expressions to match 3GPP2 service with service options associated with the bearer.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] bearer 3gpp2 service-option`** *`operator service_option_code`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***service_option_code***

Specifies the 3GPP2 service option code to match.

*service_option_code* must be an integer from 0 through 1000.

**Usage Guidelines**

Use this command to define rule expressions to match a 3GPP2 service's service option code.

### Example

The following command defines a rule expression to analyze user traffic for a 3GPP2 service's service option matching *1034*:

```
bearer 3gpp2 service-option = 1034
```

# bearer apn

This command allows you to define rule expressions to match the APN used for the subscriber session.

**Important** In 8.1 and later releases, this command is deprecated and is replaced by the bearer 3gpp apn command.

## Product

GGSN

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**`[ no ] bearer apn [ case-sensitive ]`** *`operator apn_name`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *apn_name*

Specifies the APN to match.

*apn_name* must be the name of an APN, and must be an alphanumeric string of 1 through 62 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match APN used for subscriber session.

**Example**

The following command defines a rule expression to match user traffic based on APN name *apn12*:

```
bearer apn = apn12
```

# bearer imsi

This command allows you to define rule expressions to match IMSI number of the subscriber.

**Important**

In 8.1 and later releases, this command is deprecated and is replaced by the bearer 3gpp imsi command.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer imsi {** *operator imsi* **| { !range | range } imsi-pool** *imsi_pool_name* **}**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***imsi***

Specifies the IMSI number to match.

**!range | range**

Specifies the range criteria:

- **!range**: Not in the range of
- **range**: In the range of

**imsi-pool *imsi_pool_name***

Specifies an IMSI pool.

*imsi_pool_name* must be the name of an IMSI pool, and must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match IMSI number of subscriber.

**Example**

The following command defines a rule expression to match user traffic based on IMSI number *9198838330912*:

```
bearer imsi = 9198838330912
```

# bearer rat-type

This command allows you to define rule expressions to match Radio Access Technology (RAT) in the bearer flow.

**Important**

In 8.1 and later releases, this command is deprecated and is replaced by the command.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer rat-type** *operator rat_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***rat_type***

Specifies the RAT type to match.

*rat_type* must be one of the following:

- **geran**: GSM EDGE Radio Access Network type
- **utran**: UMTS Terrestrial Radio Access Network type
- **wlan**: Wireless LAN type

**Usage Guidelines** Use this command to define rule expressions to match a RAT type.

**Example**

The following command defines a rule expression to match user traffic based on RAT type **wlan**:

```
bearer rat-type = wlan
```

# bearer sgsn-address

This command allows you to define rule expressions to match IP address of the SGSN (in acting as GGSN) / P-GW (if acting as S-GW) in the bearer flow.

**Important** In 8.1 and later releases, this command is deprecated and is replaced by the command.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] bearer sgsn-address** *operator ipv4/ipv6_address*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***ipv4/ipv6_address***

Specifies the SGSN IP address to match.

*ipv4/ipv6_address* must be in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

**Usage Guidelines**

Use this command to define rule expressions to match IP address of the SGSN (in acting as GGSN) / P-GW (if acting as S-GW).

**Example**

The following command defines a rule expression to match user traffic based on SGSN node IP address *10.1.1.1*:

```
bearer sgsn-address = 10.1.1.1
```

# bearer traffic-group

This command allows you to define rule expressions to match traffic group number associated with the subscriber session.

**Important** This functionality is available only if the Content Access Control license has been installed on the chassis.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] bearer traffic-group** *operator group_number*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***group_number***

Specifies the traffic group number to match.

*group_number* must be an integer from 1 through 255.

**Usage Guidelines**

Use this command to define rule expressions to match traffic group of the subscriber session. See the **fa-ha-spi** command in the *HA Service Configuration Mode Commands* chapter for more information.

**Example**

The following command defines a rule expression to analyze all traffic groups assigned a value greater or equal to *23*:

```
bearer traffic-group >= 23
```

# cca quota-state

Specifies the quota state of a subscriber for prepaid credit control service. In release 12.0 and later, this command should be used as a post-processing rule. For more information on post-processing policy command, refer to the *ACS Rulebase Configuration Mode Commands* chapter in this guide.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
[ no ] cca quota-state operator { limit-reached | lower-bandwidth }
```

**no**

Disables the configured credit control quota state.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

**limit-reached**

This state matches an affirmative end-of-quota indication for the current ruledef from the prepay server.

**lower-bandwidth**

This state matches the lower-bandwidth quota state of a rating group.

**Usage Guidelines**

This command supports URL redirection and creates a rule for subscriber prepaid quota state as exhausted or not exhausted.

If a subscriber has exhausted the quota but has not exhausted the qualified period, a different charging-action can be applied via the **cca quota-state** command.

**Example**

The following command defines a rule expression to match user traffic based on the Credit-Control Application (CCA) quota state **limit-reached**:

```
cca quota-state = limit-reached
```

# cca redirect-indicator

This command allows you to define rule expressions to match redirect-indicator state of the Credit Control Application. In release 12.0 and later, this command should be used as a post-processing rule. For more information on post-processing policy command, refer to *ACS Rulebase Configuration Mode Commands* chapter in this reference.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**`[ no ] cca redirect-indicator`** *`operator redirect_indicator`*

### no

Disables the configured CCA redirect-indicator in the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

### *redirect_indicator*

Specifies the redirect indicator for the AVP used for redirection of the URL in the RADIUS dictionary for prepaid service. It must be an integer from 0 through 4294967295.

**Important** For the RADIUS server configured with different values to return for this AVP, the ACS requires ruledefs to match the different values for system to associate with charging actions that have different redirect URLs configured.

## Usage Guidelines

This command is used to configure an AVP to be used from a dictionary that defines the AVP for the redirect-indicator.

For example, a RADIUS dictionary specifies the 3gpp2-release-indicator to be used for the redirect indicator when RADIUS is used as the Credit-Control Application. In this case, the value for 3gpp2-release-indicator that is returned by the RADIUS prepaid server for a quota request for a given content ID is retained by system and associated with the flow.

### Example

The following command defines a rule expression to match redirect indicator *1234* for the URL Redirect AVP:

```
cca redirect-indicator = 1234
```

# copy-packet-to-log

This command allows you to print every packet that hits the current ruledef to a log statement.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] copy-packet-to-log**

**no**

Disables the copy-packet-to-log feature.

**copy-packet-to-log**

Specifies to print packets hitting the current ruledef to a log.

**Usage Guidelines** Use this command to print every packet that hits a ruledef to a log statement. This facilitates debugging.

# description

Allows you to enter descriptive text for this configuration.

**Product** All

**Privilege** Security Administrator, Administrator

**Syntax Description**

```
description text
no description
```

**no**

Clears the description for this configuration.

***text***

Enter descriptive text as an alphanumeric string of 1 to 100 characters.

If you include spaces between words in the description, you must enclose the text within double quotation marks (" "), for example, "AAA BBBB".

**Usage Guidelines** The description should provide useful information about this configuration.

# dns answer-name

This command allows you to define rule expressions to match answer name in the answer section of DNS response messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] dns answer-name [ case-sensitive ]`** *`operator value`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***value***

Specifies the value to match.

*value* must be an alphanumeric string of 1 through 255 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match an answer name from the answer section of DNS response messages.

The answer section of a DNS response may contain more than one answer. A maximum of seven answers from the response packet are parsed. For the equality expressions (=, contains, starts-with, ends-with) a match is sought from any of the answers in the packet (up to the first seven answers). For the inequality expressions (!=, !contains, !starts-with, !ends-with), a non-match is sought from all answers (up to the first seven answers).

### Example

The following command defines a rule expression to match user traffic for answer name *test*:

```
dns answer-name = test
```

# dns any-match

This command allows you to define rule expressions to match all DNS packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] dns any-match`** *`operator condition`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define an any-match rule expression to match all DNS packets.

### Example

The following command defines an any-match rule expression to match all DNS packets:

```
dns any-match = TRUE
```

# dns previous-state

This command allows you to define rule expressions to match previous state of the DNS FSM.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] dns previous-state`** *`operator dns_previous_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***dns_previous_state***

Specifies the previous state to match.

*dns_previous_state* must be one of the following:

- **dns-timeout**
- **init**
- **req-sent**
- **resp-error**
- **resp-success**

**Usage Guidelines**

Use this command to define rule expressions to match previous state of DNS FSM.

### Example

The following command defines a rule expression to match the DNS FSM previous state **req-sent**:

```
dns previous-state = req-sent
```

# dns query-name

This command allows you to define rule expressions to match query name in DNS request messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] dns query-name [ case-sensitive ]`** *`operator query_name`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***query_name***

Specifies the query name to match.

*query_name* must be an alphanumeric string of 1 through 255 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match query name in DNS request messages.

**Example**

The following command defines a rule expression to match DNS query name *test*:

```
dns query-name = test
```

# dns query-type

This command allows you to define rule expressions to match the query type in the DNS request messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] dns query-type** *operator query_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **=**: Specifies that the query-name must be equal to the one specified.
- **!=**: Specifies that the query-name must not be equal to the one specified.

***query_type***

Specifies the query type to match.

The following *query_type* are supported:

- A
- CNAME

- NS
- PTR
- SRV
- AAA
- TXT
- ANY
- NULL

**Usage Guidelines**

Use this command to define rule expressions to match the query type in the DNS request messages.

When enabled, the **dns query-type** CLI supports the following behavior:

- DNS request with only one query is supported.
- DNS response with multiple answers is supported. Query-type corresponding to all the answers is stored and matched to the highest priority ruledef.
- For DNS response with multiple answers, unsupported query-type (mentioned previously) is skipped and parsing continues for remaining answers.
- For 'TXT' and 'NULL' query types, minimal parsing occurs like only a DNS record is created and query-type is stored. 'Answer-name' is not extracted and hence the corresponding EDR field is not populated.
- For NULL query types, response is not parsed and matching is based on the same ruledef as a Request.

This CLI is disabled by default.

### Example

The following command defines a rule expression to match the DNS query type *txt*:

```
dns query-type = txt
```

# dns return-code

This command allows you to define rule expressions to match response code in DNS response messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] dns return-code** *operator return_code*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***return_code***

Specifies the response code to match.

*return_code* must be one of the following:

- **format-error**
- **name-error**
- **no-error**
- **not-implemented**
- **refused**
- **server-failure**

**Usage Guidelines**

Use this command to define rule expressions to match response code in DNS response messages.

**Example**

The following command defines a rule expression to match a DNS response code **refused**:

```
dns return-code = refused
```

# dns state

This command allows you to define rule expressions to match current state of DNS FSM.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] dns state** *operator dns_current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***dns_current_state***

Specifies the current state to match.

*dns_current_state* must be one of the following:

- **dns-timeout**
- **init**
- **req-sent**
- **resp-error**
- **resp-success**

**Usage Guidelines**

Use this command to define rule expressions to match DNS FSM current state.

**Example**

The following command defines a rule expression to match DNS FSM current state of **req-sent**:

```
dns state = req-sent
```

# dns tid

This command allows you to define rule expressions to match Transaction Identifier (TID) field in DNS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] dns tid** *operator tid_value*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***tid_value***

Specifies the DNS transaction identifier to match.

*tid_value* must be an integer from 1 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match a TID field of DNS messages.

**Example**

The following command defines a rule expression to match DNS TID field value of *test*:

```
dns tid = test
```

# email

This command allows you to define rule expressions to match generic e-mail message parameters. These expressions will be applicable for IMAP, MMS, POP3, and SMTP protocols.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] email { cc | content { class | type } | from | size | subject | to } [ case-sensitive ]** *operator value*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### cc

Specifies to match the "cc" field of standard e-mail message.

### content { class | type }

Specifies to match the "content-type" or "content-class" field of standard e-mail message.

### from

Specifies to match the "from" field of standard e-mail message.

### subject

Specifies to match the "subject" field of standard e-mail message.

### to

Specifies to match the "to" field of standard e-mail message.

### size

Specifies to match with the total size of e-mail message specified in bytes.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following except for **size**:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with

- **starts-with**: Starts with

*operator* must be one of the following for **size**:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***value***

Specifies the value to match.

*value* must be an alphanumeric string and can contain punctuation characters.

- **cc**: A string of 1 through 512 characters
- **content**: A string of 1 through 128 characters
- **from**: A string of 1 through 64 characters
- **size**: A range of bytes from 1 through 4000000000 bytes
- **subject**: A string of 1 through 128 characters
- **to**: A string of 1 through 512 characters

**Usage Guidelines** Use this command to define rule expressions to match different fields/parameters within standard e-mail messages.

**Example**

The following command defines a rule expression to analyze user traffic for the occurrence of *triangle* in the "cc" field of e-mail messages:

```
email cc contains triangle@xyz.com
```

# end

Exits the current configuration mode and returns to the Exec mode.

**Product** All

**Privilege** Security Administrator, Administrator

**Syntax Description**

```
end
```

**Usage Guidelines** Use this command to return to the Exec mode.

# exit

Exits the current mode and returns to the parent configuration mode.

**Product**

All

**Privilege**

Security Administrator, Administrator

**Syntax Description**

**exit**

**Usage Guidelines**

Use this command to return to the parent configuration mode.

# file-transfer any-match

This command allows you to define rule expressions to match all file-transfer packets. This expression applies to file transfers that use the FTP or HTTP protocols.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] file-transfer any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**

- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match all file-transfer packets. This expression applies to file transfers that use the FTP or HTTP protocols.

### Example

The following command defines a rule expression to match all file-transfer packets:

```
file-transfer any-match = TRUE
```

# file-transfer chunk-number

This command allows you to define rule expressions to match the total number of chunks in an HTTP file as determined by the File Transfer analyzer.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] file-transfer chunk-number** *operator chunks_number*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

### *chunks_number*

Specifies the number of chunks to match.

*chunks_number* must be an integer from 1 through 65535.

**Usage Guidelines** Use this command to define rule expressions to match the total number of chunks in an HTTP file as determined by the File Transfer analyzer.

### Example

The following command defines a rule expression to match *150* number of chunks:

```
file-transfer chunk-number = 150
```

# file-transfer current-chunk-length

This command allows you to define rule expressions to match the length of an HTTP chunk currently in the File Transfer analyzer.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] file-transfer current-chunk-length`** *`operator current_chunk_length`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *current_chunk_length*

Specifies the current chunk length value (in bytes) to match.

*current_chunk_length* must be an integer from 1 through 40000000.

**Usage Guidelines** Use this command to define rule expressions to match the length of an HTTP chunk currently in the File Transfer analyzer.

### Example

The following command defines a rule expression to match length of current HTTP chunk as *1500000* bytes:

```
file-transfer current-chunk-length = 1500000
```

# file-transfer declared-chunk-length

This command allows you to define rule expressions to match the declared length of an HTTP chunk currently in the File Transfer analyzer.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] file-transfer declared-chunk-length** *operator declared_chunk_length*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *declared_chunk_length*

Specifies the declared chunk length value (in bytes) to match.

*declared_chunk_length* must be an integer from 1 through 40000000.

**Usage Guidelines**

Use this command to define rule expressions to match the declared length of an HTTP chunk currently in the File Transfer analyzer.

### Example

The following command defines a rule expression to match declared length of the current HTTP chunk as *2500000* bytes:

```
file-transfer declared-chunk-length = 2500000
```

# file-transfer declared-file-size

This command allows you to define rule expressions to match the declared file size by the File Transfer analyzer decoding the FTP handshake.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] file-transfer declared-file-size`** *`operator declared_file_size`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *declared_file_size*

Specifies the declared file size (in bytes) to match.

*declared_file_size* must be an integer from 1 through 40000000.

**Usage Guidelines**

Use this command to define rule expressions to match the declared file size by the File Transfer analyzer decoding the FTP handshake.

### Example

The following command defines a rule expression to match declared file size as *2500000* bytes:

```
file-transfer declared-file-size = 2500000
```

# file-transfer filename

This command allows you to define rule expressions to match file name.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] file-transfer filename [ case-sensitive ]`** *`operator file_name`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***file_name***

Specifies the file name to match.

*file_name* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match file name in file-transfer.

**Example**

The following command defines a rule expression to match file name containing *star1*:

```
file-transfer filename contains star1
```

# file-transfer previous-state

This command allows you to define rule expressions to match previous state of File Transfer FSM.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] file-transfer previous-state** *operator file_transfer_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***file_transfer_previous_state***

Specifies the previous state to match.

*file_transfer_previous_state* must be one of the following:

- **init**: Specifies previous state as initialization.
- **request-sent**: Specifies previous state as request sent.

- **transfer-error**: Specifies previous state as transfer error.
- **transfer-ok**: Specifies previous state as transfer ok.

**Usage Guidelines** Use this command to define rule expressions to match previous state of File Transfer FSM.

**Example**

The following command defines a rule expression to match previous state of **init**:

```
file-transfer previous-state = init
```

# file-transfer state

This command allows you to define rule expressions to match the current state of File Transfer FSM.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] file-transfer state** *operator file_transfer_current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***file_transfer_current_state***

Specifies the current state to match.

*file_transfer_current_state* must be one of the following

- **init**: Specifies current state as initialization.
- **request-sent**: Specifies current state as request sent.

- **transfer-error**: Specifies current state as transfer error.
- **transfer-ok**: Specifies current state as transfer ok.

**Usage Guidelines**

Use this command to define rule expressions to match current state of File Transfer FSM.

The following table describes details of File Transfer FSM states with event:

| Event | init | request-sent | transfer-ok | transfer-err |
|---|---|---|---|---|
| FTP "RETR" command or HTTP "GET" request received with chunk encoding | request-sent | Discarded | Discarded | Discarded |
| HTTP 2xx response received | transfer-ok | Discarded | Discarded | Discarded |
| HTTP 4xx or HTTP 5xx response received | transfer-error | Discarded | Discarded | Discarded |
| FTP reply received with reply status as file-transfer complete/successful | Discarded | transfer-ok | Discarded | Discarded |
| FTP reply received with reply status as file-transfer unsuccessful | Discarded | transfer-error | Discarded | Discarded |

### Example

The following command defines a rule expression to match file-transfer current state of **init**:

```
file-transfer state = init
```

# file-transfer transferred-file-size

This command allows you to define rule expressions to match the size of a file that has been transferred so far, as detected by the File Transfer analyzer.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] file-transfer transferred-file-size** *operator transferred_file_size*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***transferred_file_size***

Specifies the transferred file size (in bytes) to match.

*transferred_file_size* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match the size of the file that has been transferred so far, as detected by the File Transfer analyzer.

**Example**

The following command defines a rule expression to match file transferred size of *2500* bytes:

```
file-transfer transferred-file-size = 2500
```

# ftp any-match

This command allows you to define rule expressions to match all FTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define a rule expression to match all FTP packets.

**Example**

The following command defines a rule expression to match all FTP packets:

```
ftp any-match = TRUE
```

# ftp client-ip-address

This command allows you to define rule expressions to match IP address of the FTP client.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] ftp client-ip-address`** *`operator ipv4/ipv6_address`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***ipipv4/ipv6_address***

Specifies the FTP client IP address to match.

*ipv4/ipv6_address* must be in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

**Usage Guidelines**

Use this command to define rule expressions to match an FTP client IP address, which will be either the IP source address or the IP destination address, depending on the direction.

**Example**

The following command defines a rule expression to match client IP address *10.1.1.1*:

```
ftp client-ip-address = 10.1.1.1
```

# ftp client-port

This command allows you to define rule expressions to match port number of the FTP client.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ftp client-port** *operator port_number*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***port_number***

Specifies the client port number to match.

*port_number* must be an integer from 1 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match port number of the FTP client, which will be either the TCP source port or the TCP destination port, depending on the direction.

**Example**

The following command defines a rule expression to match FTP client port number *10*:

```
ftp client-port = 10
```

# ftp command args

This command allows you to define rule expressions to match arguments within an FTP command.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ftp command args [ case-sensitive ]** *operator argument*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***argument***

Specifies the argument to match.

*argument* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines** Use this command to define rule expressions to match arguments within an FTP command.

**Example**

The following command defines a rule expression to match argument *ascii* within an FTP command:

```
ftp command args = ascii
```

# ftp command id

This command allows you to define rule expressions to match FTP command ID.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] ftp command id** *operator command_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***command_id***

Specifies the command identifier to match.

In 8.3 and earlier releases, *command_id* must be an integer from 0 through 15.

In 9.0 and later releases, *command_id* must be an integer from 0 through 18.

**Usage Guidelines**

Use this command to define rule expressions to match FTP command ID.

**Example**

The following command defines a rule expression to match the FTP command ID *10*:

```
ftp command id = 10
```

# ftp command name

This command allows you to define rule expressions to match FTP command name.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ftp command name** *operator command_name*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***command_name***

Specifies the command name to match.

*command_name* must be one of the following:

- **abor**: Abort command
- **cwd**: Current working directory command
- **eprt**: eprt command
- **epsv**: epsv command
- **list**: List command
- **mode**: Transfer mode command
- **pass**: Password command
- **pasv**: Passive command
- **port**: Port command
- **quit**: Quit command
- **rest**: Restore command
- **retr**: Retry command
- **stor**: Store command
- **stru**: File structure command
- **syst**: System command
- **type**: Type command
- **user**: User command

## Usage Guidelines

Use this command to define rule expressions to match FTP command name.

### Example

The following command defines a rule expression to match FTP command name **list**:

```
ftp command name = list
```

# ftp connection-type

This command allows you to define rule expressions to match FTP connection type.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] ftp connection-type`** *`operator connection_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***connection_type***

Specifies the connection type to match.

*connection_type* must be one of the following:

- **0**: Unknown
- **1**: Control connection
- **2**: Data connection

**Usage Guidelines**

Use this command to define rule expressions to match an FTP connection type.

**Example**

The following command defines a rule expression to match FTP connection type **1**:

```
ftp connection-type = 1
```

# ftp data-any-match

This command allows you to define rule expressions to match all FTP data packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp data-any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all FTP data packets.

### Example

The following command defines a rule expression to match all FTP data packets:

```
ftp data-any-match = TRUE
```

# ftp filename

This command allows you to define rule expressions to match FTP file name.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp filename [ case-sensitive ]`** *`operator file_name`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***file_name***

Specifies the file name to match.

*file_name* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match an FTP file name.

### Example

The following command defines a rule expression to match a file named *testtransfer*:

```
ftp filename = testtransfer
```

# ftp pdu-length

This command allows you to define rule expressions to match the length of a current FTP packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ftp pdu-length** *operator pdu_length*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *pdu_length*

Specifies the FTP PDU length (in bytes) to match.

*pdu_length* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match the length of a current FTP packet, that is, FTP PDU length (FTP header + FTP payload).

### Example

The following command defines a rule expression to match an FTP PDU length of *9647* bytes:

```
ftp pdu-length = 9647
```

# ftp pdu-type

This command allows you to define rule expressions to match FTP Protocol Data Unit (PDU) type.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp pdu-type`** *operator pdu_type*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *pdu_type*

Specifies the PDU type to match.

*pdu_type* must be one of the following:

- **0**: Unknown
- **1**: Command
- **2**: Reply

**Usage Guidelines** Use this command to define rule expressions to match a PDU type of FTP packet.

**Example**

The following command defines a rule expression to match FTP PDU type **1**:

```
ftp pdu-type = 1
```

# ftp previous-state

This command allows you to define rule expressions to match previous state of FTP session.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] ftp previous-state** *operator ftp_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***ftp_previous_state***

Specifies the previous state to match.

*ftp_previous_state* must be one of the following:

- **command-sent**
- **init**
- **response-error**
- **response-ok**

**Usage Guidelines** Use this command to define rule expressions to match a previous state of FTP session.

**Example**

The following command defines a rule expression to match previous FTP state **init**:

```
ftp previous-state = init
```

# ftp reply code

This command allows you to define rule expressions to match FTP reply code.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] ftp reply code** *operator reply_code*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***reply_code***

Specifies the FTP reply code to match.

*reply_code* must be an integer from 100 through 599.

**Usage Guidelines** Use this command to define rule expressions to match an FTP reply code.

### Example

The following command defines a rule expression to match FTP reply code *150*:

```
ftp reply code = 150
```

# ftp server-ip-address

This command allows you to define rule expressions to match FTP server IP address.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp server-ip-address`** *operator ipv4/ipv6_address*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *ipv4/ipv6_address*

Specifies IP address of the server to match

*ipv4/ipv6_address* must be in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

**Usage Guidelines**

Use this command to define rule expressions to match an FTP server IP address, which will be either the IP source address or the IP destination address, depending on the direction.

### Example

The following command defines a rule expression to match the FTP server IP address *10.1.1.1*:

```
ftp server-ip-address = 10.1.1.1
```

# ftp server-port

This command allows you to define rule expressions to match FTP server port number.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp server-port`** *operator port*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *port*

Specifies the FTP server port number to match.

*port* must be an integer from 1 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match an FTP server port number, which will be either the TCP source port or the TCP destination port, depending on the direction.

### Example

The following command defines a rule expression to analyze user traffic for FTP server port *21*:

```
ftp server-port = 21
```

# ftp session-length

This command allows you to define rule expressions to match the total number of bytes sent on an FTP control connection.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp session-length`** *`operator session_length`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *session_length*

Specifies the FTP session length (in bytes) to match.

*session_length* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match the total number of bytes sent on an FTP control connection.

### Example

The following command defines a rule expression to match FTP session length of *40000* bytes:

```
ftp session-length = 40000
```

# ftp state

This command allows you to define rule expressions to match the current state of an FTP session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp state`** *operator ftp_state*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *ftp_state*

Specifies the FTP state to match.

*ftp_state* must be one of the following:

- **close**: FTP transmissions that are in closed state.
- **command-sent**: FTP transmissions that are in command-sent state.
- **response-error**: FTP transmissions that are in response-error state.
- **response-ok**: FTP transmissions that are in response-ok state.

**Usage Guidelines**

Use this command to define rule expressions to match the current state of an FTP session.

### Example

The following command defines a rule expression to match FTP current state **close**:

```
ftp state = close
```

# ftp url

This command allows you to define rule expressions to match the FTP URL/path of a file being transferred.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp url [ case-sensitive ]`** *`operator url`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***url***

Specifies the URL to match.

*url* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the FTP URL/path of a file being transferred.

**Example**

The following command defines a rule expression to match the URL *ftp://rfc.ietf.org/rfc/rfc1738.txt*:

```
ftp url = ftp://rfc.ietf.org/rfc/rfc1738.txt
```

# ftp user

This command allows you to define rule expressions to match the user name FTP command packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ftp user [ case-sensitive ]`** *`operator ftp_user`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals

- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***ftp_user***

Specifies the FTP user name to match.

*ftp_user* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match a user name FTP command.

**Example**

The following command defines a rule expression to match FTP user name *user1*:

```
ftp user = user1
```

# http accept

This command allows you to define rule expressions to match content types that are acceptable for the response.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http accept [ case-sensitive ]** *operator accept_field*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal

- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!present**: Not present
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **present**: Present
- **starts-with**: Starts with

***accept_field***

Specifies the ACCEPT field present in the HTTP header to be matched.

*accept_field* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match content types in the HTTP header that are acceptable for the response.

**Example**

The following command defines a rule expression to match content that contains *cisco* in HTTP ACCEPT field:

```
http accept contains cisco
```

# http any-match

This command allows you to define rule expressions to match all HTTP and HTTPS Connect Method packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match all HTTP packets.

**Example**

The following command defines a rule expression to match all HTTP packets:

```
http any-match = TRUE
```

# http attribute-in-data

This command allows you to define rule expressions to match any arbitrary attribute in the payload following the HTTP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] http attribute-in-data** *attribute* **[ case-sensitive ]** *operator value*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***attribute***

*attribute* must be an alphanumeric string of 1 through 31 characters.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***value***

Specifies the value as an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match arbitrary attribute in the payload following the HTTP headers.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

# http attribute-in-url

This command allows you to define rule expressions to match arbitrary attribute in the combined Host+URI HTTP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http attribute-in-url** *attribute* **[ case-sensitive ]** *operator value*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***attribute***

*attribute* must be an alphanumeric string of 1 through 31 characters.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***value***

Specifies the value as an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to configure rule expression to match an arbitrary attribute in the combined Host+URI HTTP headers.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

# http content disposition

This command allows you to define rule expressions to match optional content-disposition field of HTTP entity header.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] http content disposition [ case-sensitive ]** *operator content_disposition*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *content_disposition*

This field offers a mechanism for the sender to transmit presentational information to the recipient, allowing each component of a message to be tagged with an indication of its desired presentation semantics.

*content_disposition* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match optional content-disposition field of HTTP entity header. This feature supports RFC 2616 for HTTP and RFC 1806 for Content Disposition.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

### Example

The following command defines a rule expression to match content disposition *successful*:

```
http content disposition = successful
```

# http content length

This command allows you to define rule expressions to match the value in HTTP Content-Length entity-header field.

## Product

ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

## Syntax Description

**`[ no ] http content length`** *`operator content_length`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

### *content_length*

Specifies the HTTP body length (in bytes) to match.

*content_length* must be an integer from 1 through 4000000000.

## Usage Guidelines

Use this command to define rule expressions to match value in HTTP Content-Length entity-header field.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

### Example

The following command defines a rule expression to match value of *10000* bytes in HTTP Content-Length entity-header field:

```
http content length = 10000
```

# http content range

This command allows you to define rule expressions for CAE re-addressing to verify if the HTTP Response has content-range header or not.

**Important** In release 20.0, MVG is not supported. This command must not be used in release 20.0. For more information, contact your Cisco account representative.

**Product**

ACS

MVG

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http content range = TRUE`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**Usage Guidelines**

Use this command to define rule expressions for CAE re-addressing to verify if the HTTP Response has content-range header or not. This header is useful in detecting HTTP video requests when using ECS DPI ruledefs based on HTTP headers/URI.

# http content type

This command allows you to define rule expressions to match value in HTTP Content-Type entity-header field.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http content type [ case-sensitive ]`** *`operator content_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies the content type to match.

*content_type* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match value in HTTP Content-Type entity-header field.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

**Example**

The following command defines a rule expression to match *abc100* in HTTP Content-Type entity-header field:

```
http content type = abc100
```

# http cookie

This command allows you to define rule expressions to match strings in the HTTP cookie header.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] http cookie [ case-sensitive ]`** *`operator cookie_string`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!present**: Not present
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **present**: Present
- **starts-with**: Starts with

***cookie_string***

Specifies the string to match in the HTTP cookie header.

*cookie_string* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match strings in an HTTP cookie header.

The cookie match ruleline can be combined with other rulelines having different match criteria. Multiple line cookie header strings can be combined together using a comma (,) separator.

**Important** The HTTP parser can parse up to a maximum of 4096 bytes in the cookie header. In the case of multiple line cookie headers, the maximum of 4096 bytes includes the total size of all cookie header values, and the separators added to combine them.

### Example

The following command defines a rule expression to match the HTTP cookie header with the string *tollfree*:

```
http cookie = tollfree
```

# http domain

This command allows you to define rule expressions to match the domain portion of URIs in HTTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http domain [ case-sensitive ]`** *`operator domain`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals

- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***domain***

Specifies the domain to match.

*domain* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the domain portion of URIs in HTTP packets.

From the URL, after http:// (if present) is removed, everything until the first "/" is the domain.

**Example**

The following command defines a rule expression to match user traffic based on domain name *testdomain*:

```
http domain = testdomain
```

# http error

This command allows you to define rule expressions to match for errors in HTTP packets (for example, invalid HTTP header) and errors in the HTTP analyzer FSM (Finite State Machine) while parsing HTTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http error`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match for errors in HTTP packets and other errors in HTTP analyzer FSM while parsing HTTP packets. For example, FSM error, invalid header field values, ACS memory and buffer limit, packet related errors, and so on.

ACS supports pipelining of up to 32 HTTP requests on the same TCP connection. Pipeline overflow requests are not analyzed. Such overflow requests are treated as HTTP error. The billing system, based on this information, decides to charge or not charge, or refund the subscriber accordingly.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

**Example**

The following command defines a rule expression to match user traffic based on HTTP error status of **TRUE**:

```
http error = TRUE
```

# http first-request-packet

This command allows you to define rule expressions to match the GET or POST request, if it is the first HTTP request for the subscriber's session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http first-request-packet** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match the GET or POST request, if it is the first HTTP request for the subscriber's session.

This expression can be connected with a charging action, so the subscriber is redirected to a splash page for the first Web access attempted.

**Example**

The following command defines a rule expression to match first-request-packet:

```
http first-request-packet = TRUE
```

# http header-length

This command allows you to define rule expressions to match HTTP header length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http header-length** *operator header_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***header_length***

Specifies the HTTP header length (in bytes) to match.

*header_length* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match the length of an HTTP header.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

**Example**

The following command defines a rule expression to match an HTTP header length of *8000*:

```
http header-length = 8000
```

# http host

This command allows you to define rule expressions to match value in HTTP Host request-header field.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http host [ case-sensitive ]** *operator host_name*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***host_name***

Specifies the host name to match.

*host_name* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match value in HTTP Host request-header field.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 1: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |

| Regex Character | Description |
|---|---|
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |

| Regex Character | Description |
|---|---|
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match *host1* in HTTP Host request-header field:

```
http host = host1
```

The following command defines a regex rule expression to match either of the following values in the HTTP Host request-header field: host1, host23w01.

```
http host regex "host1|host23w01"
```

# http payload-length

This command allows you to define rule expressions to match HTTP payload length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http payload-length** *operator payload_length*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals

- =: Equals
- >=: Greater than or equals

***payload_length***

Specifies the HTTP payload (data) length (in bytes) to match.

*payload_length* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match HTTP payload (data) length (pdu-length - header-length).

**Example**

The following command defines a rule expression to match HTTP payload length of *100000* bytes:

```
http payload-length = 100000
```

# http pdu-length

This command allows you to define rule expressions to match the total length of a single HTTP packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http pdu-length** *operator pdu_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***pdu_length***

Specifies the HTTP PDU length (in bytes) to match.

*pdu_length* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match the total length of a single HTTP packet. This will also match packets with partial HTTP message (due to fragmentation).

**Example**

The following command defines a rule expression to match an HTTP PDU length of *10000* bytes:

```
http pdu-length = 10000
```

# http previous-state

This command allows you to define rule expressions to match previous state of HTTP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http previous-state`** *`operator http_previous_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***http_previous_state***

Specifies the previous state to match.

*http_previous_state* must be one of the following:

- **init**: Initialized state

- **response-error**: Response error state
- **response-ok**: Response ok state
- **waiting-for-response**: Waiting for response state

**Usage Guidelines** Use this command to define rule expressions to match a previous state of HTTP sessions.

#### Example

The following command defines a rule expression to match HTTP previous state **response-ok**:

```
http previous-state = response-ok
```

# http referer

This command allows you to define rule expressions to match the value in the HTTP Referer request-header field.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] http referer [ case-sensitive ]`** *`operator referer_name`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!present**: Not present

- **!starts-with**: Does not start with
- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **present**: Present
- **regex**: Regular expression
- **starts-with**: Starts with

***referer_name***

Specifies the HTTP referer name to match.

*referer_name* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match value in HTTP Referer request-header field.

This feature allows an operator to collect or track all URLs visited during a particular subscriber session. These URLs include the entire string of visited URLs, including all referral links. This information is output in an Event Data Record (EDR) format to support reporting or billing functions.

For example, if a subscriber begins a mobile web session and clicks on the "Sports" link from the home deck, and then selects ESPN and moves to an advertiser link, the operator can capture all URLs for that entire session. During this period ACS collects the URLs for a particular subscriber session; collection can be limited by time duration or number of URLs visited.

ACS generates EDRs that contain HTTP URL and the HTTP referer fields along with other fields.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 2: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |

| Regex Character | Description |
|---|---|
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |

| Regex Character | Description |
|---|---|
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match the HTTP referer *cricket.espn.com*:

```
http referer = cricket.espn.com
```

# http reply code

This command allows you to define rule expressions to match status code associated with HTTP response packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http reply code** *operator reply_code*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals

- >=: Greater than or equals

***reply_code***

Specifies the HTTP reply code to match.

*reply_code* must be an integer from 100 through 599.

**Usage Guidelines**

Use this command to define rule expressions to match status code associated with HTTP response codes.

**Example**

The following command defines a rule expression to match HTTP response code *204*:

```
http reply code = 204
```

# http reply payload

This command allows you to define rule expressions to enable video detection using HTTP payload content.

**Important**

In release 20.0, MVG is not supported. This command must not be used in release 20.0. For more information, contact your Cisco account representative.

**Product**

ACS

MVG

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
[ no ] http reply payload type = video
```

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**Usage Guidelines**

Use this command to enable inspection for video in HTTP Response payload. Request payloads will not be inspected.

# http request method

This command allows you to define rule expressions to match HTTP request method.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http request method`** *operator request_method*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***request_method***

Specifies the HTTP request method to match.

*request_method* must be one of the following:

- **connect**
- **delete**
- **get**
- **head**
- **options**
- **post**
- **put**
- **trace**

**Usage Guidelines**

Use this command to define rule expressions to match an HTTP request method.

**Example**

The following command defines a rule expression to match user traffic based on HTTP request method **connect**:

```
http request method = connect
```

# http session-length

This command allows you to define rule expressions to match HTTP session length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http session-length`** *`operator session_length`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***session_length***

Specifies the HTTP total session length (in bytes) to match.

*session_length* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match a total HTTP session length.

**Example**

The following command defines a rule expression to match an HTTP session length of *200000*:

```
http session-length = 200000
```

# http state

This command allows you to define rule expressions to match current state of an HTTP session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] http state`** *`operator current_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state of HTTP session to match.

*current_state* must be one of the following:

- **close**: Closed state
- **response-error**: Response error state
- **response-ok**: Response ok state
- **waiting-for-response**: Waiting for response state

**Usage Guidelines**

Use this command to define rule expressions to match a current state of an HTTP session.

**Example**

The following command defines a rule expression to match current state **close**:

```
http state = close
```

# http transaction-length

This command allows you to define rule expressions to match HTTP transaction length (combined length of one HTTP GET Request message and its associated response messages).

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http transaction-length {`** *`operator transaction_length`* **`| { { range | !range }`** *`range_from`* **`to`** *`range_to`* **`} }`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***transaction_length***

Specifies the HTTP transaction length (in bytes) to match.

*transaction_length* must be an integer from 1 through 4000000000.

**{ range | !range }** ***range_from*** **to** ***range_to***

Enables or disables the range criteria for length of transaction.

- **range**: Enables the range criteria for HTTP transaction length.
- **!range**: Disables the range criteria for HTTP transaction length.
- *range_from*: Specifies the start of range (in bytes) for HTTP transaction length.
- *range_to*: Specifies the end of range (in bytes) for HTTP transaction length.

**Usage Guidelines** Use this command to define rule expressions to match an HTTP transaction length [one HTTP GET Request message + associated response message(s)] in bytes.

### Example

The following command defines a rule expression to match an HTTP transaction length of *10200* bytes:

```
http transaction-length = 10200
```

# http transfer-encoding

This command allows you to define rule expressions to match the value in HTTP Transfer-Encoding general-header field.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] http transfer-encoding [ case-sensitive ]`** *`operator transfer_encoding`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains

- **ends-with**: Ends with
- **starts-with**: Starts with

***transfer_encoding***

Specifies the HTTP transfer encoding to match.

*transfer_encoding* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the value in HTTP Transfer-Encoding general-header field.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

**Example**

The following command defines a rule expression to match the value *chunked* in HTTP Transfer-Encoding general-header field:

```
http transfer-encoding = chunked
```

# http uri

This command allows you to define rule expressions to match HTTP URI.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http uri [ case-sensitive ]** *operator uri*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***uri***

Specifies the HTTP URI to match.

*uri* must be an alphanumeric string of 1 through 127 characters, and can contain punctuation characters, and excludes the "host" portion.

## Usage Guidelines

Use this command to define rule expressions to match an HTTP URI, excluding the host portion.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 3: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |

| Regex Character | Description |
|---|---|
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match the HTTP URI string *http://www.somehost.com*:

```
http uri = http://www.somehost.com
```

The following command defines a regex rule expression to match either of the following or similar values in the HTTP URI string: http://server19.com/search?form=zip, http://server20.com/search?form=pdf

```
http uri regex
"(http://|http://www).server[0-2][0-9].com/search?form=(pdf|zip)"
```

# http url

This command allows you to define rule expressions to match HTTP URL.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http url [ case-sensitive ]** *operator url*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***url***

Specifies the HTTP URL to match.

*url* must be an alphanumeric string of 1 through 127 characters. that allows punctuation characters and includes "host + URI" for HTTP PDUs.

For example, in case of the URL "http://www.google.fr/", the host is "http://www.google.fr", and the URI is "/":

```
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Accept: */*\r\n
  Accept-Language: fr\r\n
  Accept-Encoding: gzip, deflate\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)\r\n
  Host: www.google.fr\r\n
  Connection: Keep-Alive\r\n
  \r\n
```

## Usage Guidelines

Use this command to define rule expressions to match HTTP URL.

**Important** When rule lines are added or modified, the entire trie is recreated and it mallocs memory for every URL present in the configuration. This leads to huge memory allocation that gets freed once the trie is created.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *ECS Administration Guide*.

***Table 4: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |

| Regex Character | Description |
|---|---|
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match the HTTP URL *http://rfc.ietf.org/rfc/rfc1738.txt*:

```
http url = http://rfc.ietf.org/rfc/rfc1738.txt
```

The following command defines a regex rule expression to match either of the following or similar values in the HTTP URL string: http://yahoo.com, http://www.yahoo.co.in, http://yahoo.com/news.

```
http url regex "(http://|http://www).yahoo.(co.in|com)*"
```

# http user-agent

This command allows you to define rule expressions to match the User-Agent request-header field of HTTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http user-agent [ case-sensitive ]`** *`operator user_agent`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!present**: Not present
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **present**: Present
- **starts-with**: Starts with

***user_agent***

Specifies the HTTP user agent value to match.

*user_agent* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match value in HTTP user-agent header field.

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

**Example**

The following command defines a rule expression to match *xyz.123* in HTTP user-agent header field:

```
http user-agent = xyz.123
```

# http version

This command allows you to define rule expressions to match version information in HTTP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] http version [ case-sensitive ]`** *`operator http_version`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!present**: Not present

- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **present**: Present
- **starts-with**: Starts with

***http_version***

Specifies this HTTP version value to match.

*http_version* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match HTTP version.

**Example**

The following command defines a rule expression to match HTTP version *http4.2*:

```
http version = http4.2
```

# http x-header

This command allows you to define rule expressions to match specified field within extension-headers (x-headers).

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] http x-header** *field_name* **[ case-sensitive ]** *operator string*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***field_name***

*field_name* must be an alphanumeric string of 1 through 31 characters.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!present**: Not present
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **present**: Present
- **starts-with**: Starts with

#### *string*

Specifies the HTTP x-header value to match.

*string* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match specified fields within x-headers. The extension-header can be any header field not specified in RFCs.

All x-header fields must begin with "x-".

In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

#### Example

The following command defines a rule expression to match the extension-header *test_field* for the value *test_string*:

```
http x-header test_field = test_string
```

# icmp any-match

This command allows you to define rule expressions to match all ICMP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] icmp any-match`** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all ICMP packets.

**Example**

The following command defines a rule expression to match all ICMP packets:

```
icmp any-match = TRUE
```

# icmp code

This command allows you to define rule expressions to match value in the Code field of ICMP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] icmp code`** *`operator code`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***code***

Specifies the ICMP code to match.

*code* must be an integer from 0 through 255.

**Usage Guidelines**

Use this command to define rule expressions to match a code field of ICMP packets.

**Example**

The following command defines a rule expression to match ICMP code *11*:

```
icmp code = 11
```

# icmp type

This command allows you to define rule expressions to match value in Type field of ICMP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] icmp type`** *`operator type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***type***

Specifies the ICMP type to match.

*type* must be an integer from 0 through 255. For example, 0 for Echo Reply, 3 for Destination Unreachable, and 5 for Redirect.

**Usage Guidelines** Use this command to define rule expressions to match a type field of ICMP packets.

**Example**

The following command defines a rule expression to match user traffic based on ICMP type *3*:

```
icmp type = 3
```

# icmpv6 any-match

This command allows you to define rule expressions to match all ICMPv6 packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] icmpv6 any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all ICMPv6 packets.

**Example**

The following command defines a rule expression to match all ICMPv6 packets:

```
icmpv6 any-match = TRUE
```

# icmpv6 code

This command allows you to define rule expressions to match value in Code field of ICMPv6 packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] icmpv6 code** *operator code*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***code***

Specifies the ICMPv6 code to match.

*code* must be an integer from 0 through 255.

**Usage Guidelines**

Use this command to define rule expressions to match a code field of ICMPv6 packets.

**Example**

The following command defines a rule expression to match ICMPv6 code *134*:

```
icmpv6 code = 134
```

# icmpv6 type

This command allows you to define rule expressions to match type field of ICMPv6 packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] icmpv6 type** *operator type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***type***

Specifies the ICMPv6 type to match.

*type* must be an integer from 0 through 255. For example, 129 for Echo Reply, 3 for Time Exceeded, and 137 for Redirect Message.

**Usage Guidelines**

Use this command to define rule expressions to match type field of ICMPv6 packets.

**Example**

The following command defines a rule expression to match ICMPv6 type *133*:

```
icmpv6 type = 133
```

# if-protocol

This command allows you to associate different content IDs with the same ruledef, depending on the protocol being used.

**Product**

**Important** In StarOS 18.0 and later releases, this command has been deprecated.

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
if-protocol { http | wsp-connection-less | wsp-connection-oriented }
content-id content_id
no if-protocol { http | wsp-connection-less | wsp-connection-oriented }
```

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

#### http

Specifies HTTP protocol.

This is the same as the rule expression **http any-match = true**.

#### wsp-connection-less

Specifies WSP connection-less protocol.

This is the same as requiring "**wsp any-match = true**" but "**wtp any-match = false**" (that is, connection-less WAP1.x).

#### wsp-connection-oriented

Specifies WSP connection-oriented protocol.

This is the same as the combined rule expression "**wsp any-match = true**" and "**wtp any-match = true**" (that is, connection-oriented WAP1.x).

#### content-id *content_id*

Specifies the content ID for the specified protocol.

In 12.1 and earlier releases, *content_id* must be an integer from 1 through 65535.

In 12.2 and later releases, *content_id* must be an integer from 1 through 2147483647.

### Usage Guidelines

Use this command to associate different content IDs with the same ruledef, depending on the protocol being used.

This command is only effective for charging ruledefs. See the command for information on how to configure charging ruledefs.

If a particular ruledef should have three different values for content-id, depending on whether the traffic is connection-oriented WAP1.x, connection-less WAP1.x, or WAP2.0, within the ruledef we should have configuration similar to the following:

**if-protocol wsp-connection-oriented content-id** *1*

**if-protocol wsp-connection-less content-id** *2*

**if-protocol http content-id** *3*

Presumably, the ruledef would have another configurable like "**www url contains** *foo*", which would cause it to use different content IDs when "foo" was accessed, depending upon the protocol being used.

#### Example

The following command associates HTTP protocol and a content ID of *23*:

```
if-protocol http content-id 23
```

# imap any-match

This command allows you to define rule expressions to match all IMAP packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] imap any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match all IMAP packets.

**Example**

The following command defines a rule expression to match all IMAP packets:

```
imap any-match = TRUE
```

# imap cc

This command allows you to define rule expressions to match recipient address in the Carbon Copy (cc) field of e-mails in IMAP messages.

**Product** ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] imap cc [ case-sensitive ]** *operator cc_address*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *cc_address*

Specifies the e-mail "cc" address/name to match.

*cc_address* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match recipient address in the "cc" field of e-mails in IMAP messages.

### Example

The following command defines a rule expression to match recipient address *triangle@xyz.com* in the "cc" field of e-mails in IMAP messages:

```
imap cc contains triangle@xyz.com
```

# imap command

This command allows you to define rule expressions to match embedded IMAP commands in IMAP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] imap command** *operator command*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***command***

Specifies the command to match.

*command* must be one of the following:

- **append**
- **authenticate**
- **capability**
- **check**
- **close**
- **copy**
- **create**
- **delete**
- **examine**
- **expunge**

- **fetch**
- **list**
- **login**
- **logout**
- **lsub**
- **noop**
- **rename**
- **search**
- **select**
- **starttls**
- **status**
- **store**
- **subscribe**
- **uid-copy**
- **uid-fetch**
- **uid-search**
- **uid-store**
- **unsubscribe**

**Usage Guidelines** Use this command to define rule expressions to match an embedded command in the IMAP message.

### Example

The following command defines a rule expression to match **close** command in IMAP messages:

```
imap command = close
```

# imap content class

This command allows you to define rule expressions to match the content-class field of e-mails in IMAP messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**`[ no ] imap content class [ case-sensitive ]`** *`operator content_class`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *content_class*

Specifies the content class to match.

*content_class* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match the content-class field of e-mails in IMAP messages.

### Example

The following command defines a rule expression to analyze user traffic matching content class *javax.mail.internet.MimeMultipart* in the content-class field of e-mails in IMAP messages:

```
imap content class contains javax.mail.internet.MimeMultipart
```

# imap content type

This command allows you to define rule expressions to match the content-type field of e-mails in IMAP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] imap content type [ case-sensitive ]`** *`operator content_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies the content type field to match.

*content_type* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines** Use this command to define rule expressions to match the content-type field of e-mails in IMAP messages.

### Example

The following command defines a rule expression to analyze user traffic matching content type *TEXT/plain; charset=iso-8859-1* in the content-type field of e-mails in IMAP messages:

```
imap content type contains TEXT/plain; charset=iso-8859-1
```

# imap date

This command allows you to define rule expressions to match the Date field of e-mails in IMAP messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] imap date [ case-sensitive ]** *operator date*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***date***

Specifies the date to match.

*date* must be an alphanumeric string of 1 through 127 characters that may include punctuation marks and spaces as shown in the example below.

**Usage Guidelines**

Use this command to define rule expressions to match the date field of e-mails in IMAP messages.

**Example**

The following command defines a rule expression to analyze user traffic matching date *Fri, 20 Jan 2012 11:00:00 -0600* in the "date" field of e-mails in IMAP messages:

```
imap date contains Fri, 21 Jan 2012 11:00:00 -0600
```

# imap final-reply

This command allows you to define rule expressions to match final-reply value for the last IMAP final-reply message.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] imap final-reply`** *`operator final_reply`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***final_reply***

Specifies the "final-reply" condition to match.

*final_reply* must be one of the following:

- **bad**: Final reply is invalid or bad.
- **no**: There is no final reply.
- **ok**: Final reply is valid.

**Usage Guidelines** Use this command to define rule expressions to match a final-reply value for the last IMAP final-reply message.

### Example

The following command defines a rule expression to analyze user traffic matching the final-reply condition **bad** in the last IMAP final-reply message:

```
imap final-reply = bad
```

# imap from

This command allows you to define rule expressions to match the from field of e-mails in IMAP messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] imap from [ case-sensitive ]** *operator from_address*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with

- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***from_address***

Specifies the "from" address/value to match.

*from_address* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the from field of e-mails in IMAP messages.

**Example**

The following command defines a rule expression to analyze user traffic matching *triangle* in the "from" field of e-mails in the IMAP messages:

```
imap from contains triangle
```

# imap mail-size

This command allows you to define rule expressions to match IMAP e-mail users that have e-mails of a specified size in their mailboxes.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] imap mail-size** *operator mail_size*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal

- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***mail_size***

Specifies the total size of mail, in bytes, to match.

*mail_size* must be an integer from 0 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to discover the number of IMAP e-mail users that have e-mails of a specified size in their mailboxes.

**Example**

The following command defines a rule expression to match users with e-mail size less than or equal to *23400* bytes:

```
imap mail-size <= 23400
```

# imap mailbox-size

This command allows you to define rule expressions to match IMAP e-mail user having a specified number of messages in their mailboxes.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] imap mailbox-size`** *operator number_of_email*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals

- =: Equals
- >=: Greater than or equals

***number_of_email***

Specifies the total number of e-mail messages in mailbox of an IMAP user to match.

*number_of_email* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match the number of IMAP e-mail users having a specified number of messages in their mailboxes.

**Example**

The following command defines a rule expression to match e-mail users having less than or equal to *1024* e-mail messages in their mailboxes:

```
imap mailbox-size <= 1024
```

# imap message-type

This command allows you to define rule expressions to match the type of IMAP packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] imap message-type** *operator message_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!**=: Does not equal
- =: Equals

***message_type***

Specifies the IMAP packet message-type to match.

*message_type* must be one of the following:

- **command-continuation-reply**: Message with command-continuation-reply type.
- **final-reply**: Message is of final reply type.
- **request**: There is of request type.
- **untagged-reply**: Message of reply type, but without any tag.

**Usage Guidelines** Use this command to define rule expressions to match the IMAP message type.

**Example**

The following command defines a rule expression to match IMAP sessions with message type **request**:

```
imap message-type = request
```

# imap previous-state

This command allows you to define rule expressions to match the previous state of IMAP request sessions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] imap previous-state** *operator imap_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***imap_previous_state***

Specifies the previous state to match.

*imap_previous_state* must be one of the following:

- **init**: Message in initialization state.
- **request-sent**: Message in request-sent state.

**Usage Guidelines** Use this command to define rule expressions to match previous state of IMAP request session.

**Example**

The following command defines a rule expression to match IMAP sessions with previous state **init**:

```
imap previous-state = init
```

# imap session-length

This command allows you to define rule expressions to match the total length of an IMAP session.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] imap session-length`** *`operator session_length`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***session_length***

Specifies the total length of IMAP session (in bytes) to match.

*session_length* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match the total length of IMAP sessions.

The session length is calculated by adding together the IP payloads (that is, starting after the IP header) of all relevant IMAP session packets.

**Example**

The following command defines a rule expression to match IMAP sessions with length less than or equal to *4000* bytes:

```
imap session-length <= 4000
```

# imap session-previous-state

This command allows you to define rule expressions to match the previous state of an IMAP session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] imap session-previous-state** *operator imap_session_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***imap_session_previous_state***

Specifies the previous state of IMAP session to match.

*imap_session_previous_state* must be one of the following:

- **authenticated**: Session authenticated
- **connected**: Session connected
- **init**: Session initialized
- **mailbox-selected**: Mailbox selected

**Usage Guidelines** Use this command to define rule expressions to match the previous state of IMAP sessions.

**Example**

The following command defines a rule expression to match IMAP sessions with previous state **init**:

```
imap session-previous-state = init
```

# imap session-state

This command allows you to define rule expressions to match the current state of IMAP sessions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
[ no ] imap session-state operator session_current_state
```

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***session_current_state***

Specifies the current state to match.

*session_current_state* must be one of the following:

- **authenticated**: Session authenticating.
- **connected**: Session connecting.
- **logout**: Session logged out.
- **mailbox-selected**: Mailbox selecting.

**Usage Guidelines** Use this command to define rule expressions to match the current state of IMAP sessions.

**Example**

The following command defines a rule expression to match IMAP sessions with current state **connected**:

```
imap session-state = connected
```

# imap state

This command allows you to define rule expressions to match the current state of IMAP sessions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] imap state** *operator current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies current state of IMAP session to match.

*current_state* must be one of the following:

- **request-sent**: Request message sent
- **response-fail**: Request response failed
- **response-ok**: Request response is good

**Usage Guidelines** Use this command to define rule expressions to match the current state of IMAP session.

### Example

The following command defines a rule expression to match IMAP sessions with current state **response-fail**:

```
imap state = response-fail
```

# imap subject

This command allows you to define rule expressions to match the subject field of e-mails in IMAP messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] imap subject [ case-sensitive ]`** *`operator subject`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with

- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***subject***

Specifies the "subject" to match.

*subject* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters and space as shown in the example below.

**Usage Guidelines**

Use this command to define rule expressions to match "subject" field of e-mail in IMAP message.

**Example**

The following command defines rule expression to match occurrence of the string *My test* in the "subject" field of e-mails in IMAP message:

```
imap subject contains My test
```

# imap to

This command allows you to define rule expressions to match the "to" field of e-mails in IMAP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] imap to [ case-sensitive ]** *operator to*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***to***

Specifies the "to" field value to match.

*to* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match "to" field of e-mails in IMAP messages.

**Example**

The following command defines a rule expression to analyze user traffic matching the occurrence *xyz.com* in the "to" field of e-mails in the IMAP message:

```
imap to contains xyz.com
```

# ip any-match

This command allows you to define rule expressions to match all IPv4/IPv6 packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ip any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match IPv4/IPv6 packets.

**Example**

The following command defines a rule expression to match IPv4/IPv6 packets:

```
ip any-match = TRUE
```

# ip dscp

This command enables you to configure a ruledef with the DSCP value and match it with the DSCP value in the incoming IP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
[ no ] ip dscp { operator } ipv4_tos_value | ipv6_tc_value [ mask mask_value ]
```

**no**

If previously configured, removes the specified DSCP value and the mask from the configuration.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***ipv4_tos_value | ipv6_tc_value***

Specifies the DSCP value to match with the incoming IP packets.

The *ipv4_tos_value* or *ipv6_tc_value* must be an integer from 0 through 63.

**mask *mask_value***

Specifies the mask for the number of bits in the DSCP value to be considered for matching.

*mask_value* must be an integer from 0 through 63. The default mask value is 63.

**Usage Guidelines**

Use this command to check if the DSCP value in the IPv4 ToS or IPv6 TC field of incoming IP packet matches with configured ToS/TC value.

**Example**

The following command will match all incoming packets which has DSCP value 20:

```
ip dscp = 20 mask 31
```

# ip downlink

This command allows you to define rule expressions to match downlink (network to subscriber) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ip downlink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match downlink (to subscriber) IP packets.

**Example**

The following command defines a rule expression to match IP packet in downlink direction:

```
ip downlink = TRUE
```

# ip dst-address

This command allows you to define rule expressions to match IP destination address field within IP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ip dst-address { `***`operator`*** **` { `***`ipv4/ipv6_address`*** **` | `***`ipv4/ipv6_address/mask`*** **` | address-group `***`ipv6_address`*** **` } | { !range | range } host-pool `***`host_pool_name`*** **` }`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

*operator*: Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals

- =: Equals
- >=: Greater than or equals

***ipv4/ipv6_address***

Specifies the IP address of the destination node for outgoing traffic. *ipv4/ipv6_address* must be an IP address in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

***ipv4/ipv6_address/mask***

Specifies the IP address of the destination node for outgoing traffic. *ipv4/ipv6_address/mask* must be an IP address in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation with subnet mask bit. The mask bit is a numeric value which corresponding to the number of bits in the subnet mask.

**address-group *ipv6_address***

**Important** The **address-group** keyword can be configured only after the = operator. The wildcard support has not been provided for IPv4 addresses.

Specifies a group of IPv6 addresses configured with wildcard input and/or specialized range input. Multiple wildcard characters can be accepted as input and only one 2 byte range input will be accepted. Both wildcard character input and 2 byte range input can be configured together within a given IPv6 address.

In the example — *2607:7700:*:[2020-3040]::ce1d:b083/128*, * is a wildcard input and [2020-3040] is a 2 byte specialized range input.

**{ !range | range } host-pool *host_pool_name***

**!range | range**: Specifies the range criteria:

- **!range**: Not in the range of
- **range**: In the range of

**host-pool** *host_pool_name*: Specifies the name of the host pool. *host_pool_name* must be an alphanumeric string of 1 through 63 characters.

## Usage Guidelines

Use this command to define rule expressions to match the IP destination address field within IP headers.

### Example

The following command defines a rule expression to match user traffic based on the IPv4 destination address *10.1.1.1*:

```
ip dst-address = 10.1.1.1
```

The following command defines a rule expression to match user traffic based on the given destination IPv6 address where * is the wildcard input and *[2020-3040]* is the 2 byte specialized range input:

```
ip dst-address = 2607:7700:*:[2020-3040]::ce1d:b083/128
```

# ip error

This command allows you to define rule expressions to match user traffic for invalid IP packets and other errors, for example IP header error, while parsing IP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] ip error`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match invalid IP packets and any other errors while parsing IP packets.

### Example

The following command defines a rule expression to match user traffic for invalid IP packets and other errors:

```
ip error = TRUE
```

# ip protocol

This command allows you to define rule expressions to match the protocol field in IP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ip protocol`** *`operator`* **`{`** *`protocol_assignment_no`* **`|`** *`protocol`* **`}`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals—available only in 8.1 and later releases
- **=**: Equals
- **>=**: Greater than or equals—available only in 8.1 and later releases

***protocol_assignment_no***

Specifies the protocol by assignment number.

*protocol_assignment_no* must be an integer from 0 through 255.

For example, 1 for ICMP, 6 for TCP, and 17 for UDP.

***protocol***

Specifies the protocol by name.

*protocol* must be one of the following:

- **ah**
- **esp**
- **gre**
- **icmp**

- **icmpv6**
- **tcp**
- **udp**

**Usage Guidelines** Use this command to define rule expressions to match protocol field in IP packet headers.

**Example**

The following command defines a rule expression to match protocol assignment number *1*:

```
ip protocol = 1
```

# ip server-domain-name

This command allows you to define rule expressions to match host names (domain names).

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] ip server-domain-name** *operator domain_name*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***domain_name***

Specifies the domain name to match.

*domain_name* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match full or partial host names (domain names).

The rule will be matched for the learnt IP addresses resolved from DNS queries to the specified domain names. DNS responses for the specified domain names will be snooped and the learnt IP addresses stored.

Besides being used for standard rule matching, this command also enables the DNS Snooping feature if the rulebase references any ruledefs with this configuration. The DNS protocol analyzer must also be enabled in the rulebase.

### Example

The following command defines a rule expression to match domain name values containing *star*:

```
ip server-domain-name contains star
```

# ip server-ip-address

This command allows you to define rule expressions to match the IP address of the destination end of the connection.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ip server-ip-address { *operator* { *ipv4/ipv6_address* | *ipv4/ipv6_address/mask* | address-group *ipv6_address* } | { !range | range } host-pool *host_pool_name* }**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

*operator*: Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***ipv4/ipv6_address***

Specifies the server IP address. For uplink packets (subscriber to network), this field matches the destination IP address in the IP header. For downlink packets (network to subscriber), this field matches the source IP address in the IP header.*ipv4/ipv6_address* must be an IP address in IPv4 dotted-decimal notation or IPv6 colon-separated-hexadecimal notation.

***ipv4/ipv6_address/mask***

Specifies the server IP address with subnet mask bit. For uplink packets (subscriber to network), this field matches the destination IP address in the IP header. For downlink packets (network to subscriber), this field matches the source IP address in the IP header. *ipv4/ipv6_address/mask* must be an IP address in IPv4 dotted-decimal notation or IPv6 colon-separated-hexadecimal notation with subnet mask bit. The mask bit is a numeric value which is the number of bits in the subnet mask.

**address-group *ipv6_address***

**Important** The **address-group** keyword can be configured only after the = operator. The wildcard support has not been provided for IPv4 addresses.

Specifies a group of IPv6 addresses configured with wildcard input and/or specialized range input. Multiple wildcard characters can be accepted as input and only one 2 byte range input will be accepted. Both wildcard character input and 2 byte range input can be configured together within a given IPv6 address.

In the example — *2607:7700:*:[2020-3040]::ce1d:b083/128*, * is a wildcard input and [2020-3040] is a 2 byte specialized range input.

**{ !range | range } host-pool *host_pool_name***

**!range | range**: Specifies the range criteria:

- **!range**: Not in the range of
- **range**: In the range of

**host-pool** *host_pool_name*: Specifies name of the host pool. *host_pool_name* must be an alphanumeric string of 1 through 63 characters.

## Usage Guidelines

Use this command to define rule expressions to match the IP address of the destination end of the connection.

For uplink packets, this field matches the destination IP address in the IP header. For downlink packets, this field matches the source IP address in the IP header.

### Example

The following command defines a rule expression to match user traffic based on IPv4 server address *10.1.1.1*:

```
ip server-ip-address = 10.1.1.1
```

The following command defines a rule expression to match user traffic based on the given destination IPv6 address where * is the wildcard input and *[2020-3040]* is the 2 byte specialized range input:

```
ip server-ip-address = 2607:7700:*:[2020-3040]::ce1d:b083/128
```

# ip src-address

This command allows you to define rule expressions to match the source IP address field within IP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] ip src-address { *operator* { *ipv4/ipv6_address* | *ipv4/ipv6_address/mask* | address-group *ipv6_address* } | { !range | range } host-pool *host_pool_name* }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

*operator*: Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***ipv4/ipv6_address***

Specifies IP address of the source node for incoming traffic. *ipv4/ipv6_address* must be an IP address in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

***ipv4/ipv6_address/mask***

Specifies the IP address of the source node for incoming traffic with subnet mask bit. *ipv4/ipv6_address/mask* must be an IP address in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation with subnet mask bit. The mask bit is a numeric value which corresponds to the number of bits in the subnet mask.

**address-group *ipv6_address***

**Important** The **address-group** keyword can be configured only after the = operator. The wildcard support has not been provided for IPv4 addresses.

Specifies a group of IPv6 addresses configured with wildcard input and/or specialized range input. Multiple wildcard characters can be accepted as input and only one 2 byte range input will be accepted. Both wildcard character input and 2 byte range input can be configured together within a given IPv6 address.

In the example — *2607:7700:*:[2020-3040]::ce1d:b083/128*, * is a wildcard input and [2020-3040] is a 2 byte specialized range input.

**{ !range | range } host-pool *host_pool_name***

**!range | range**: Specifies the range criteria:

- **!range**: Not in the range of
- **range**: In the range of

**host-pool** *host_pool_name*: Specifies name of the host pool. *host_pool_name* must be a string of 1 through 63 characters.

**Usage Guidelines** Use this command to define rule expressions to match IP source address field within IP header.

**Example**

The following command defines a rule expression to match user traffic based on IPv4 source address *10.1.1.1*:

```
ip src-address = 10.1.1.1
```

The following command defines a rule expression to match user traffic based on the given source IPv6 address where * is the wildcard input and *[2020-3040]* is the 2 byte specialized range input:

```
ip src-address = 2607:7700:*:[2020-3040]::ce1d:b083/128
```

# ip subscriber-ip-address

This command allows you to define rule expressions to match the IP address of the subscriber, which will be either the source or destination address depending on the direction.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] ip subscriber-ip-address { *operator* { *ipv4/ipv6_address* | *ipv4/ipv6_address/mask* | address-group *ipv6_address* } | { !range | range } host-pool *host_pool_name* }**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

*operator*: Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

### *ipv4/ipv6_address*

Specifies the subscriber IP address. Depending on the direction of packet this IP address will be either the IP source address or the IP destination address. *ipv4/ipv6_address* must be an IP address in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

### *ipv4/ipv6_address/mask*

Specifies the subscriber IP address with subnet mask bit. Depending on the direction of packet this IP address will either be the IP source address or the IP destination address. *ipv4/ipv6_address/mask* must be an IP address in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation with subnet mask bit. The mask bit is a numeric value which corresponds to the number of bits in the subnet mask.

### address-group *ipv6_address*

**Important** The **address-group** keyword can be configured only after the = operator. The wildcard support has not been provided for IPv4 addresses.

Specifies a group of IPv6 addresses configured with wildcard input and/or specialized range input. Multiple wildcard characters can be accepted as input and only one 2 byte range input will be accepted. Both wildcard character input and 2 byte range input can be configured together within a given IPv6 address.

In the example — *2607:7700:*:[2020-3040]::ce1d:b083/128*, * is a wildcard input and [2020-3040] is a 2 byte specialized range input.

### { !range | range } host-pool *host_pool_name*

**!range | range**: Specifies the range criteria:

- **!range**: Not in the range of
- **range**: In the range of

**host-pool** *host_pool_name*: Specifies the name of the host pool. *host_pool_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the IP address of the subscriber, which will be either the source or destination address depending on the direction.

#### Example

The following command defines a rule expression to match user traffic based on subscriber IPv4 address *10.1.1.1*:

```
ip subscriber-ip-address = 10.1.1.1
```

The following command defines a rule expression to match user traffic based on the given subscriber IPv6 address where * is the wildcard input and *[2020-3040]* is the 2 byte specialized range input:

```
ip subscriber-ip-address = 2607:7700:*:[2020-3040]::ce1d:b083/128
```

# ip total-length

This command allows you to define rule expressions to match the total length field in IP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ip total-length`** *`operator total_length`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals

- >=: Greater than or equals

***total_length***

Specifies the total length of the IP packet (including payload) to match.

*total_length* must be an integer from 0 through 4096.

**Usage Guidelines**

Use this command to define rule expressions to match the total length field in IP headers.

**Example**

The following command defines a rule expression to match user traffic based on IP total length of *2000* bytes:

```
ip total-length = 2000
```

# ip uplink

This command allows you to define rule expressions to match uplink (subscriber to network) IP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] ip uplink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match uplink (subscriber to network) IP packets.

**Example**

The following command defines a rule expression to match uplink packets:

```
ip uplink = TRUE
```

# ip version

This command allows you to define rule expressions to match the version number in IP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] ip version** *operator ip_version*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be = (equals).

***ip_version***

Specifies the IP version to match.

*ip_version* must be one of the following:

- **ipv4**
- **ipv6**

**Usage Guidelines** Use this command to define rule expressions to match version number in IP header.

### Example

The following command defines a rule expression to match user traffic for the IP version **ipv6**:

```
ip version = ipv6
```

# mms any-match

This command allows you to define rule expressions to match all Multimedia Messenging Service (MMS) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] mms any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all MMS packets.

### Example

The following command defines a rule expression to match all MMS packets:

```
mms any-match = TRUE
```

# mms bcc

This command allows you to define rule expressions to match recipient addresses in the bcc field of MMS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] mms bcc [ case-sensitive ]`** *`operator bcc_address`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

#### *bcc_address*

Specifies the "bcc" address/value to match.

*bcc_address* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters and space.

### Usage Guidelines

Use this command to define rule expressions to match recipient address in the "bcc" field of MMS messages.

#### Example

The following command defines a rule expression to match recipient address containing *test1* in "bcc" field of MMS messages:

```
mms bcc contains test1
```

# mms cc

This command allows you to define rule expressions to match recipient addresses in the cc field of MMS messages.

### Product

ACS

### Privilege

Security Administrator, Administrator

### Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

### Syntax Description

**[ no ] mms cc [ case-sensitive ]** *operator cc_address*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with

- **!starts-with**: Does not start with
- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***cc_address***

Specifies the "cc" address/value to match.

*cc_address* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters and space.

**Usage Guidelines**

Use this command to define rule expressions to match recipient addresses in "cc" field of MMS messages.

**Example**

The following command defines a rule expression to match recipient address containing *test1* in the "cc" field of MMS messages:

```
mms cc contains test1
```

# mms content location

This command allows you to define rule expressions to match the content-location field of MMS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms content location [ case-sensitive ]** *operator string*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***string***

Specifies the value to match.

*string* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters and space.

**Usage Guidelines**

Use this command to define rule expressions to match the content-location field of MMS messages.

**Example**

The following command defines a rule expression to match *test1* in content-location field of MMS messages:

```
mms content location contains test1
```

# mms content type

This command allows you to define rule expressions to match the content-type field of MMS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms content type [ case-sensitive ]** *operator content_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies the MMS content type to match.

*content_type* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters and space.

**Usage Guidelines**

Use this command to define rule expressions to match content-type field of MMS messages.

**Example**

The following command defines a rule expression to match *image* in content-type field of MMS messages:

```
mms content type contains image
```

# mms downlink

This command allows you to define rule expressions to match downlink (network to subscriber) MMS packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms downlink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the downlink (from the Mobile Node direction) status to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match downlink MMS packets.

**Example**

The following command defines a rule expression to match all downlink MMS packets:

```
mms downlink = TRUE
```

# mms from

This command allows you to define rule expressions to match the "from" field in MMS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] mms from [ case-sensitive ]`** *`operator from_address`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***from_address***

Specifies the "from" address/value to match.

*from_address* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters and space.

**Usage Guidelines**

Use this command to define rule expressions to match the "from" field of MMS messages.

**Example**

The following command defines a rule expression to match *test1* in the "from" field of MMS messages:

```
mms from contains test1
```

# mms message-id

This command allows you to define rule expressions to match the message ID field of MMS messages.

**Product**

ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] mms message-id [ case-sensitive ]** *operator message_id*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *message_id*

Specifies the MMS message ID to match.

*message_id* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match the "message ID" field of MMS messages.

### Example

The following command defines a rule expression to match *test1* in the "message ID" field of MMS messages:

```
mms message-id contains test1
```

# mms pdu-type

This command allows you to define rule expressions to match Protocol Data Unit (PDU) type in the current MMS packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms pdu-type** *operator pdu_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***pdu_type***

Specifies the MMS PDU type to match.

*pdu_type* must be one of the following:

- **mms-pdu-type-m-acknowledge-ind**
- **mms-pdu-type-m-delivery-ind**
- **mms-pdu-type-m-http-get**
- **mms-pdu-type-m-notification-ind**
- **mms-pdu-type-m-notify-rsp-ind**
- **mms-pdu-type-m-retrieve-conf**
- **mms-pdu-type-m-send-conf**
- **mms-pdu-type-m-send-request**
- **mms-pdu-type-m-wsp-get**

- **mms-pdu-type-response**: This option is deprecated. Use the **mms_pdu_type_m_retrieve_conf** option instead.

**Usage Guidelines** Use this command to define rule expressions to match the PDU type in the current MMS packet.

### Example

The following command defines a rule expression to match PDU type **mms-pdu-type-m-http-get** in the current MMS packet:

```
mms pdu-type = mms-pdu-type-m-http-get
```

# mms previous-state

This command allows you to define rule expressions to match the previous state of MMS sessions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] mms previous-state** *operator mss_previous_state*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *mms_previous_state*

Specifies the previous state to match.

*mms_previous_state* must be one of the following:

- **delayed-ack-pending**: This option is deprecated, use **retrieve-conf-received**.
- **delayed-m-notify-rsp-sent**: This option is deprecated, use **notify-rsp-sent**.

- **delayed-retrieval-pending**: This option is deprecated, use **retrieval-pending**.
- **immediate-retrieval-pending**: This option is deprecated, use **retrieval-pending**.
- **init**
- **m-send-conf-rcvd**: This option is deprecated, use **send-success**.
- **m-send-req-sent**
- **notification-ind-rcvd**
- **notify-rsp-sent**
- **retrieval-pending**
- **retrieve-conf-received**
- **send-success**

**Usage Guidelines** Use this command to define rule expressions to match the previous state of MMS sessions.

#### Example

The following command defines a rule expression to match user traffic based on MMS previous state of **retrieval-pending**:

```
mms previous-state = retrieval-pending
```

# mms response status

This command allows you to define rule expressions to match the response status code of MMS messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] mms response status** *operator status_code*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***status_code***

Specifies the status code to match.

*status_code* must be an integer from 128 through 136.

**Usage Guidelines**

Use this command to define rule expressions to match response status code of MMS messages.

**Example**

The following command defines a rule expression to match user traffic based on MMS response status code *129*:

```
mms response status = 129
```

# mms state

This command allows you to define rule expressions to match the current state of MMS sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms state** *operator current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies current state of MMS session to match.

*current_state* must be one of the following:

- **delayed-ack-pending**: This option is deprecated, use **retrieve-conf-received**.
- **delayed-m-notify-rsp-sent**: This option is deprecated, use **notify-rsp-sent**.
- **delayed-retrieval-pending**: This option is deprecated, use **retrieval-pending**.
- **delivery-failed**
- **delivery-success**
- **immediate-retrieval-pending**: This option is deprecated, use **retrieval-pending**.
- **m-send-conf-rcvd**: This option is deprecated, use **send-success**.
- **m-send-req-sent**
- **notification-ind-rcvd**
- **notify-rsp-sent**
- **retrieval-failed**
- **retrieval-pending**
- **retrieval-success**
- **retrieve-conf-received**
- **send-success**

**Usage Guidelines**

Use this command to define rule expressions to match the current state of MMS session.

**Example**

The following command defines a rule expression to match user traffic based on the current state of MMS session as **retrieval-failed**:

```
mms state = retrieval-failed
```

# mms status

This command allows you to define rule expressions to match the current status of MMS sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms status** *operator status*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***status***

Specifies the MMS status to match.

*status* must be an integer from 128 through 132.

**Usage Guidelines**

Use this command to define rule expressions to match current status of MMS sessions.

**Example**

The following command defines a rule expression to match user traffic based on MMS current status *130*:

```
mms status = 130
```

# mms subject

This command allows you to define rule expressions to match the "subject" field of MMS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] mms subject [ case-sensitive ]** *operator subject_string*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

#### *subject_string*

Specifies the value to match.

*subject_string* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters and space.

### Usage Guidelines

Use this command to define rule expressions to match "subject" field of MMS messages.

#### Example

The following command defines a rule expression to match *test1* in the "subject" field of MMS messages:

```
mms subject contains test1
```

# mms tid

This command allows you to define rule expressions to match the "Transaction Identifier" (TID) field of MMS messages.

### Product

ACS

### Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] mms tid [ case-sensitive ]** *operator transaction_id*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *transaction_id*

Specifies the MMS TID to match.

*transaction_id* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match TID field of MMS messages.

### Example

The following command defines a rule expression to match *test* in TID field of MMS messages:

```
mms tid = test
```

# mms to

This command allows you to define rule expressions to match the "to" field of MMS messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] mms to [ case-sensitive ]`** *`operator to_address`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***to_address***

Specifies the "to" address/name to match.

*to_address* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters and space.

**Usage Guidelines**

Use this command to define rule expressions to match "to" field of MMS messages.

### Example

The following command defines a rule expression to match user traffic based on *test* in "to" field of MMS messages:

```
mms to = test
```

# mms uplink

This command allows you to define rule expressions to match uplink (subscriber to network) MMS packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] mms uplink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the uplink (from the Mobile Node direction) status to match.

*condition* must one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match uplink MMS packets.

### Example

The following command defines a rule expression to match uplink MMS packets:

```
mms uplink = TRUE
```

# mms version

This command allows you to define rule expressions to match the MMS version in MMS packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] mms version`** *`operator version`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *version*

Specifies the MMS version to match.

*version* must be an integer from 1 through 65535.

**Important**

MMS protocol analyzer supports decoding of only MMS version 1.0.

**Usage Guidelines**

Use this command to define rule expressions to match MMS version in MMS packets.

### Example

The following command defines a rule expression to match MMS version *1.0* in MMS packets:

```
mms version = 1
```

# multi-line-or all-lines

This command applies the OR operator to all lines in the current ruledef.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
[ no ] multi-line-or all-lines
```

#### no

If previously configured, deletes this configuration in the current ruledef.

#### multi-line-or all-lines

Applies the OR operator to all lines in the current ruledef.

**Usage Guidelines**

When a ruledef is evaluated, if the **multi-line-or all-lines** command is configured, the logical OR operator is applied to all the rule expressions in the ruledef to decide if the ruledef matches or not. If the **multi-line-or all-lines** command is not configured, the logical AND operator is applied to all the rule expressions.

The intent of this command is to allow a single ruledef to specify multiple URL expressions. Otherwise, multiple ruledefs need to be created, each with one URL expression. When this CLI command is used, each expression in the ruledef impacts the total number of ruledefs allowed. So from a "maximum number of possible ruledefs" perspective, it makes no difference whether there are N ruledefs with one expression each, or one ruledef with N expressions.

# p2p any-match

This command allows you to define rule expressions to match all Peer-to-Peer (P2P) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] p2p any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- =: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **TRUE**: The rule matches any P2P traffic.
- **FALSE**: The rule does not match any P2P traffic.

**Usage Guidelines**

Use this command to define rule expressions to match all P2P packets.

**Example**

The following command defines a rule expression to match all P2P packets:

```
p2p any-match = TRUE
```

# p2p app-identifier

This command allows you to configure application identifiers populated from the plugin and mark the matching flows to a custom-defined protocol (CDP) name.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] p2p app-identifier { quic-sni** *operator quic_sni_string* **| tls-cname** *operator tls_cname_string* **| tls-sni** *operator tls_sni_string* **}**

### no

If previously configured, deletes the specified configuration from the current ruledef.

### quic-sni *operator quic_sni_string*

Specifies the QUIC Server Name Indication (SNI) field value.

*operator* specifies how to match and must be one of the following:

- **!=**: Does not equal
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

*quic_sni_string* specifies the QUIC server name and must be an alphanumeric string of 1 through 127 characters.

### tls-cname *operator tls_cname_string*

Specifies the common name in the Server Hello message of TLS.

SSL renegotiation is supported for the flows that are marked using "tls-cname" rules. This feature is available only if the plugin is loaded with 20.2 or later builds.

*operator* specifies how to match and must be one of the following:

- **!=**: Does not equal
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

*tls_cname_string* specifies the common name and must be an alphanumeric string of 1 through 127 characters.

### tls-sni *operator tls_sni_string*

Specifies the TLS/SSL Server Name Indication (SNI) field.

*operator* specifies how to match and must be one of the following:

- **!=**: Does not equal
- **=**: Equals
- **contains**: Contains

- **ends-with**: Ends with
- **starts-with**: Starts with

*tls_sni_string* specifies the TLS/SSL server name and must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to configure application identifiers populated from the plugin and mark the matching flows to a custom-defined protocol (CDP) name.

The SNI ruledef supports multi-line-or all-lines or default multi-line-and rule lines. The rule lines configured with "!=" operator will not be optimized.

**Important**

The QUIC SNI Detection feature requires the latest ADC Plugin to be loaded from the adc_v2.x stream along with StarOS changes. The default plugin does not support this feature. Contact your Cisco account representative for more information.

### Example

The following command configures the QUIC SNI app-identifier that is set to *fb.com*:

```
p2p app-identifier quic-sni = fb.com
```

# p2p behavioral

This command allows you to define rule expressions to match behavioral detection type — P2P, Video, VoIP, Behavioral Upload or Behavioral Download.

**Product**

ACS, ADC

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] p2p behavioral** *operator behavioral_list*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- !=: Does not equal
- =: Equals

***behavioral_list***

Specifies the behavior to match. The behavioral list is the list of supported behavioral detection logic populated from the currently loaded ADC plugin.

*behavioral_list* must be one of the following:

- **download**: Detects unknown flows which are data download using behavioral analysis
- **p2p**: Detects P2P/file sharing protocols using behavioral analysis
- **upload**: Detects unknown flows which are data upload using behavioral analysis
- **video**: Detects video flows using behavioral analysis
- **voip**: Detects VoIP (voice and video) protocols using behavioral analysis

**Usage Guidelines**

Use this command to define rule expressions to detect behavioral protocols. Behavioral P2P and behavioral VoIP are meant for zero day detection of P2P/file sharing protocols and VoIP traffic respectively. Behavioral upload/download is similar to client-server upload/download using HTTP, FTP, SFTP, etc. It must also detect flows of non-standard ports which ECS cannot detect and falls under the client-server model. This feature is disabled by default and meant only for statistical purposes (not for charging purposes). For detection purposes use the **p2p-detection behavioral** command in the ACS Configuration Mode.

**Example**

The following command specifies to configure behavioral VoIP:

```
p2p behavioral = voip
```

# p2p protocol

This command allows you to define rule expressions to match P2P protocol. This command must be used for charging purposes. It must not be used for detection purposes.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] p2p protocol** *operator protocol*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be = (equals).

***protocol***

Specifies the protocol to match.

*protocol* must be one of the following:

- **120Sports**
- **8tracks**
- **abcnetworks**
- **abscbn**
- **accuradio**
- **actionvoip**
- **actsync**
- **adobeconnect**
- **aenetworks**
- **aimini**
- **amazoncloud**
- **amazonmusic**
- **amazonvideo**
- **android_messages**
- **antsp2p**
- **anyconnect**
- **apple-push**
- **apple-store**
- **applejuice**
- **applemaps**
- **ares**
- **armagettron**
- **avi**

- **badoo**
- **baeblemusic**
- **baidumovie**
- **battlefld**
- **bbm**
- **beatport**
- **betternet**
- **bitcasa**
- **bittorrent**
- **bittorrent-sync**
- **blackberry-store**
- **blackberry**
- **blackdialer**
- **blackplanet-radio**
- **box**
- **btn**
- **callofduty**
- **cbssports**
- **chikka**
- **cisco-jabber**
- **citrix**
- **clubbox**
- **clubpenguin**
- **comodounite**
- **crackle**
- **crossfire**
- **crunchyroll**
- **curiosity-stream**
- **cyberghost**
- **danzwave**
- **dashradio**
- **ddlink**

- **deezer**
- **didi**
- **directconnect**
- **directv**
- **discord**
- **disneymovies**
- **dish-anywhere**
- **dns-tunneling**
- **dofus**
- **dramafever**
- **dropbox**
- **ebuddy**
- **edonkey**
- **epix**
- **eros**
- **espn**
- **expressvpn**
- **facebook**
- **facetime**

**Important** The **facetime** protocol is available only in 9.0 and in 11.0 and later releases.

- **fandor**
- **fasttrack**
- **feidian**
- **ficall**
- **fiesta**
- **filetopia**
- **filmontv**
- **fitradio**
- **flash**
- **flickr**

- **flixea**
- **florensia**
- **foursquare**
- **fox-business**
- **fox-news**
- **fox-now**
- **fox-sports**
- **foxsportsgo**
- **freenet**
- **friendster**
- **fring**
- **fubotv**
- **funshion**
- **fxnow**
- **gaana**
- **gadugadu**
- **gamekit**

**Important** The **gamekit** protocol is available only in 9.0 and in 11.0 and later releases.

- **gmail**
- **gnutella**
- **go90**
- **goober**
- **google-music**
- **google-push**
- **google**
- **googleplay**
- **googleplus**
- **gotomeeting**
- **gtalk**
- **guildwars**

- **halflife2**
- **hamachivpn**
- **hayu**
- **hbogo**
- **hbonow**
- **hbonordic**
- **heytell**
- **hgtv**
- **hike-messenger**
- **hls**
- **hotspotvpn**
- **http**
- **hulu**
- **hyves**
- **iax**
- **icall**
- **icecast**
- **icloud**
- **idrive**
- **igo**
- **iheartradio**
- **imesh**
- **imessage**
- **imgur**
- **imo**
- **implus**
- **instagram**
- **iplayer**
- **iptv**
- **irc**
- **isakmp**
- **iskoot**

- **itunes**
- **jabber**
- **jap**
- **jumblo**
- **kakaotalk**
- **kidoodle**
- **kik-messenger**
- **kiswe**
- **klowdtv**
- **kontiki**
- **kugoo**
- **kuro**
- **linkedin**
- **livestream**
- **lync**
- **magicjack**
- **manolito**
- **mapfactor**
- **mapi**
- **maplestory**
- **meebo**
- **meetic**
- **mega**
- **mgcp**
- **mig33**
- **mlb**
- **mojo**
- **monkey3**
- **mozy**
- **msn**
- **msrp**
- **mute**

- **mypeople**
- **myspace**
- **nateontalk**
- **natgeotv**
- **naverline**
- **navigon**
- **nbc-sports**
- **nbc-tv**
- **netflix**
- **netmotion**
- **newsy**
- **nick**
- **nimbuzz**
- **nokia-store**
- **nrktv**
- **octoshape**
- **odkmedia**
- **odnoklassniki**
- **off**
- **ogg**
- **oist**
- **oovoo**
- **opendrive**
- **openft**
- **openvpn**
- **operamini**
- **orb**
- **oscar**
- **outlook**
- **paltalk**
- **pando**
- **pandora**

- **path**
- **pbs**
- **pcanywhere**
- **periscope**
- **pinterest**
- **playstation**
- **plingm**
- **poco**
- **pokemon-go**
- **popo**
- **pplive**
- **ppstream**
- **ps3**
- **qello_concerts**
- **qq**
- **qqgame**
- **qqlive**
- **quake**
- **quic**
- **quicktime**
- **radio-paradise**
- **rdp**
- **rdt**
- **redbulltv**
- **regram**
- **rfactor**
- **rhapsody**
- **rmstream**
- **reddit**
- **rodi**
- **rynga**
- **samsung-store**

- **scydo**
- **secondlife**
- **shalomworld**
- **shoutcast**
- **showtime**
- **silverlight**
- **siri**
- **skinny**
- **skydrive**
- **skype**
- **slacker-radio**
- **slingbox**
- **slingtv**
- **smartvoip**
- **smashcast**
- **smule**
- **snapchat**
- **softether**
- **sopcast**
- **soribada**
- **soulseek**
- **soundcloud**
- **subsplash**
- **spark**
- **spdy**
- **speedtest**
- **splashfighter**
- **spotify**
- **ssdp**
- **ssl**
- **starz**
- **stealthnet**

- **steam**
- **stun**
- **sudaphone**
- **svtplay**
- **tagged**
- **talkatone**
- **tango**
- **taxify**
- **teamspeak**
- **teamviewer**
- **telegram**
- **thunder**
- **tidal**
- **tinder**
- **tmo-tv**
- **tor**
- **truecaller**
- **truphone**
- **tumblr**
- **tunein-radio**
- **tunnelvoice**
- **turbovpn**
- **tvants**
- **tvland**
- **tvuplayer**
- **tv2sumo**
- **twitter**
- **twitch**
- **ultrabac**
- **ultrasurf**
- **univision**
- **ufc**

- **upc-phone**
- **usenet**
- **ustream**
- **uusee**
- **vchat**
- **veohtv**
- **vessel**
- **vevo**
- **viber**
- **viki**
- **vimeo**
- **vine**
- **voipdiscount**
- **vopium**
- **voxer**
- **vpnmaster**
- **vpnx**
- **vtok**
- **vtun**
- **vudu**
- **warcft3**
- **waze**
- **webex**
- **wechat**
- **weibo**
- **whatsapp**
- **wii**
- **windows-azure**
- **windows-store**
- **winmx**
- **winny**
- **willow**

- **wmstream**
- **wofkungfu**
- **wofwarcraft**
- **wuala**
- **wwe**
- **xbox**
- **xdcc**
- **xfinity**
- **xing**
- **yahoo**
- **yahoomail**
- **yiptv**
- **yogafree**
- **youku**
- **yourfreetunnel**
- **youtube**
- **zattoo**
- **zello**

**Usage Guidelines** Use this command to define rule expressions to detect P2P protocols for charging purposes. For detection purposes use the **p2p-detection protocol** command in the ACS Configuration Mode.

### Example

The following command specifies to detect orb protocol for charging purposes:

```
p2p protocol = orb
```

# p2p protocol-group

This command allows you to define rule expressions to match ADC application/protocol group.

**Product** ACS, ADC

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] p2p protocol-group** *operator group_list*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- !=: Does not equal
- =: Equals

### *group_list*

Specifies the ADC application/protocol group to match.

*group_list* must be one of the following:

- **anonymous-access**: Anonymous internet access protocols mainly used for illegal purposes.
- **business**: Applications/Protocols used for business purposes.
- **communicator**: Applications/Protocols used mainly for messaging which includes IM, IM based file transfer, VoIP or video chats.
- **cloud**: Applications/Protocols for cloud service.
- **e-mail**: Applications/Protocols used for electronic mail.
- **e-news**: Applications/Protocols used for internet news and magazine reading.
- **e-store**: Applications/Protocols used for electronic stores.
- **internet-privacy**: Applications/Protocols used for file transfers.
- **filesharing**: Applications/Protocols used for gaming.
- **gaming**: Standard protocols used in internet.
- **p2p-filesharing**: Applications/Protocols used for creating a virtual network over internet mainly for business purposes.
- **p2p-anon-filesharing**: Peer to Peer application/protocols used for anonymous filesharing.
- **remote-control**: Peer to Peer application/protocols used for filesharing.
- **social-nw-game**: Application/Protocols used for remote management.
- **social-nw-generic**: Application/Protocols used for social networking games.
- **social-nw-videoconf**: Application/Protocols used for social networking.
- **standard**: Application/Protocols used for social network video conference.

- **streaming**: Application/Protocols used for streaming audio and video.
- **untagged**: Default group for protocols not otherwise classified.

**Usage Guidelines** Use this command to define rule expressions to match ADC protocol group. The list of P2P applications/protocols is populated from the currently loaded P2P plugin.

**Example**

The following command specifies to detect the **gaming** protocol group:

```
p2p protocol-group = gaming
```

# p2p set-app-proto

This command allows you to configure the custom-defined protocol (CDP) name.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] p2p set-app-proto`** *`cdp_name_string`*

**no**

If previously configured, deletes the specified configuration from the current ruledef.

***cdp_name_string***

Specifies the name of the custom defined protocol (CDP) for TLS/SSL flows, QUIC flows or any app-identifier matching the ruledef. *cdp_name_string* must be an alphanumeric string of 1 through 19 characters.

**Usage Guidelines** Use this command to set the CDP name. If the flow/packet matches the rule, the CDP name specified in the ruledef will be taken and the flow will be marked as CDP. If no CDP is configured in the rule, then the flow will be treated as TLS/SSL or QUIC flow.

**Important** The QUIC SNI Detection feature requires the latest ADC Plugin to be loaded from the adc_v2.x stream along with StarOS changes. The default plugin does not support this feature. Contact your Cisco account representative for more information.

### Example

The following command configures the custom-defined application protocol name set to *facebook*:

```
p2p set-app-proto facebook
```

# p2p traffic-type

This command allows you to define rule expressions to match the traffic type.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] p2p traffic-type** *operator traffic_type*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *traffic_type*

Specifies the traffic type to match.

In 11.0 and later releases, *traffic_type* must be one of the following:

- **ads**
- **audio**
- **file-transfer**
- **im**
- **streaming-video**
- **unclassified**

- **video**
- **voipout**

In 10.0 and earlier releases, the supported *traffic_type* was **voice**.

**Usage Guidelines** Use this command to configure the system to detect voice or non-voice P2P traffic. When the detection of a protocol is enabled then the detection of sub-type is enabled by default.

### Example

The following command configures the system to detect video traffic:

```
p2p traffic-type = video
```

# pop3 any-match

This command allows you to define rule expressions to match all Post Office Protocol 3 (POP3) packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] pop3 any-match`** *`operator condition`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**

- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match all POP3 packets.

### Example

The following command defines a rule expression to match all POP3 packets:

```
pop3 any-match = TRUE
```

# pop3 command args

This command allows you to define rule expressions to match POP3 command arguments.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] pop3 command args [ case-sensitive ]** *operator argument*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with

- **starts-with**: Starts with

***argument***

Specifies the command argument to match.

*argument* must be an alphanumeric string of 1 through 40 characters, and may contain punctuation characters.

**Usage Guidelines** Use this command to define rule expressions to match POP3 command argument.

**Example**

The following command defines a rule expression to match POP3 command argument *test*:

```
pop3 command args = test
```

# pop3 command id

This command allows you to define rule expressions to match POP3 command ID.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] pop3 command id** *operator command_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***command_id***

Specifies the command ID to match.

*command_id* must be an integer from 1 through 12.

**Usage Guidelines**

Use this command to define rule expressions to match a POP3 command ID.

**Example**

The following command defines a rule expression to match POP3 command ID *8*:

```
pop3 command id = 8
```

# pop3 command name

This command allows you to define rule expressions to match command sent within a POP3 packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] pop3 command name** *operator command_name*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***command_name***

Specifies the command name to match.

*command_name* must be one of the following:

- **apop**
- **dele**

- **list**
- **noop**
- **pass**
- **quit**
- **retr**
- **reset**
- **stat**
- **top**
- **uidl**
- **user**

**Usage Guidelines** Use this command to define rule expressions to match commands sent within POP3 packets.

### Example

The following command defines a rule expression to match the **list** command sent in POP3 packets:

```
pop3 command name = list
```

# pop3 mail-size

This command allows you to define rule expressions to match POP3 mail size.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] pop3 mail-size {** *operator mail_size* **| { range | !range }** *range_from* **to** *range_to* **}**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

**{ range | !range }** ***range_from*** **to** ***range_to***

Enables or disables the range criteria.

- **range**: Enables the range criteria.
- **!range**: Disables the range criteria.
- *range_from*: Specifies start of the range.

  *range_from* must be an integer from 1 through 4000000000.
- *range_to*: Specifies the end range.

  *range_to* must be an integer from 1 through 4000000000, and must be greater than *range_from*.

***mail_size***

Specifies the mail size to match.

*mail_size* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match POP3 mail size.

**Example**

The following command defines a rule expression to match POP3 mail size of *40000*:

```
pop3 mail-size = 40000
```

# pop3 pdu-length

This command allows you to define rule expressions to match the Protocol Data Unit (PDU) length of POP3 packets equal to the POP3 header plus POP3 payload.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] pop3 pdu-length { operator pdu_length | { { range | !range } range_from to range_to } }`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

**{ range | !range } *range_from* to *range_to***

Enables or disables the range criteria.

- **range**: Enables the range criteria.
- **!range**: Disables the range criteria.
- *range_from*: Specifies the start of range as an integer from 0 through 65535.
- *range_to*: Specifies the end range. *range_to* must be an integer from 0 through 65535, and must be greater than *range_from*.

***pdu_length***

Specifies the POP3 PDU length to match.

*pdu_length* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match POP3 PDU length (header + payload) in bytes.

**Example**

The following command defines a rule expression to match PDU length of *1000* bytes:

```
pop3 pdu-length = 1000
```

# pop3 pdu-type

This command allows you to define rule expressions to match POP3 Protocol Data Unit (PDU) type.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] pop3 pdu-type`** *`operator pdu_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***pdu_type***

Specifies the POP3 PDU type to match.

*pdu_type* must be one of the following:

- **command-packet**
- **data-packet**
- **relay-packet**

**Usage Guidelines** Use this command to define rule expressions to match POP3 PDU type.

**Example**

The following command defines a rule expression to match POP3 PDU type **relay-packet**:

```
pop3 pdu-type = relay-packet
```

# pop3 previous-state

This command allows you to define rule expressions to match the previous state of POP3 sessions.

**Product** ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] pop3 previous-state** *operator pop3_previous_state*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *pop3_previous_state*

Specifies the previous state to match.

*pop3_previous_state* must be one of the following:

- **connected**: Connected state
- **data transaction**: Data transaction state
- **init**: Initialized state
- **reply-error**: Reply error state
- **reply-ok**: Response ok state
- **waiting-for-reply**: Waiting for reply state

## Usage Guidelines

Use this command to define rule expressions to match a POP3 previous state.

### Example

The following command defines a rule expression to match user traffic for a POP3 previous state of **connected**:

```
pop3 previous-state = connected
```

# pop3 reply args

This command allows you to define rule expressions to match specified arguments with POP3 reply.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] pop3 reply args [ case-sensitive ]`** *`operator argument`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***argument***

Specifies the reply argument to match.

In 11.0 and earlier releases, *argument* must be an alphanumeric string of 1 through 512 characters, and may contain punctuation characters.

In 12.0 and later releases, *argument* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match specified arguments within a POP3 reply.

**Example**

The following command defines a rule expression to match the argument *test* with POP3 replies:

```
pop3 reply args = test
```

# pop3 reply id

This command allows you to define rule expressions to match POP3 reply ID.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] pop3 reply id** *operator reply_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***reply_id***

Specifies the POP3 reply ID to match.

*reply_id* must be one of the following:

- **0**: Unknown reply
- **1**: +OK
- **2**: -Error

**Usage Guidelines** Use this command to define rule expressions to match POP3 reply ID.

### Example

The following command defines a rule expression to match POP3 reply ID of *2*:

```
pop3 reply id = 2
```

# pop3 reply status

This command allows you to define rule expressions to match POP3 reply status.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] pop3 reply status`** *`operator reply_status`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *reply_status*

Specifies the reply status to match.

*reply_status* must be one of the following:

- **+OK**: Reply OK
- **-ERR**: Reply error

**Usage Guidelines** Use this command to define rule expressions to match POP3 reply status.

### Example

The following command defines a rule expression to match POP3 reply status **+OK**:

```
pop3 reply status = +OK
```

# pop3 session-length

This command allows you to define rule expressions to match POP3 session-length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] pop3 session-length { *operator session_length* | { range | !range } *range_from* to *range_to* }**

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *session_length*

Specifies the POP3 session length to match.

*session_length* must be an integer from 1 through 4000000000.

#### { range | !range } *range_from* to *range_to*

Enables or disables the range criteria for PoP3 session length.

- **range**: Enables the range criteria for POP3 session length.

- **!range**: Disables the range criteria for POP3 session length.
- *range_from*: Specifies the start of range of POP3 session as an integer from 1 through 4000000000, but less than or equal to *range_to*.
- *range_to*: Specifies the end of range of POP3 session as an integer from 1 through 4000000000, but greater than or equal to *range_from*.

**Usage Guidelines**

Use this command to define rule expressions to match the total length of POP3 sessions.

**Example**

The following command defines a rule expression to match a POP3 session length of *40000*:

```
pop3 session-length = 40000
```

# pop3 state

This command allows you to define rule expressions to match the current state of POP3 sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] pop3 state** *operator current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state to match.

*current_state* must be one of the following:

- **close**
- **connected**
- **data-transaction**
- **reply-error**
- **reply-ok**
- **waiting-for-reply**

**Usage Guidelines** Use this command to define rule expressions to match the current state of POP3 sessions.

#### Example

The following command defines a rule expression to match the POP3 current state **close**:

```
pop3 state = close
```

# pop3 user-name

This command allows you to define rule expressions to match POP3 user name.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] pop3 user-name [ case-sensitive ]`** *`operator user_name`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain

- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***user_name***

Specifies the POP3 user name to match.

*user_name* must be an alphanumeric string of 1 through 64 characters, and may contain punctuation characters and space.

**Usage Guidelines**

Use this command to define rule expressions to match POP3 user name.

**Example**

The following command defines a rule expression to match POP3 user name *test*:

```
pop3 user-name = test
```

# pptp any-match

This command allows you to defines a rule expression to match all Point-to-Point Tunneling Protocol (PPTP) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] pptp any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to specify a ruledef to analyze user traffic based on the PPTP any match status.

**Example**

The following command creates a PPTP ruledef for analyzing user traffic using a PPTP any match status of *FALSE*:

```
pptp any-match = FALSE
```

# pptp ctrl-msg-type

This command allows you to define rule expressions to match control message type in PPTP packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] pptp ctrl-msg-type =** *message_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***message_type***

*message_type* must be one of the following:

- **call-clear-request**
- **call-disconnect-notify**

- **echo-reply**
- **echo-request**
- **incoming-call-connected**
- **incoming-call-reply**
- **incoming-call-request**
- **outgoing-call-reply**
- **outgoing-call-request**
- **set-link-info**
- **start-control-connection-reply**
- **start-control-connection-request**
- **stop-control-connection-reply**
- **stop-control-connection-request**
- **wan-error-notify**

**Usage Guidelines** Use this command to define rule expressions to match the control message type in PPTP packets.

**Example**

The following command specifies to match **echo-reply** message type:

```
pptp ctrl-msg-type = echo-reply
```

# pptp gre any-match

This command allows you to define rule expressions to match all PPTP Generic Routing Encapsulation (GRE) packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] pptp gre any-match =** *condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***condition***

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match all PPTP GRE packets.

**Example**

The following command defines a rule expression to match all PPTP GRE packets:

```
pptp gre any-match = TRUE
```

# radius any-match

This command allows you to define rule expressions to match all RADIUS packets.

**Product**

GGSN

PDSN

PGW

**Privilege** Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] radius any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define an any-match rule expression to match all RADIUS packets.

**Example**

The following command defines an any-match rule expression to match all RADIUS packets:

```
radius any-match = TRUE
```

# radius error

This command allows you to define rule expressions to match for errors in RADIUS packets and errors in the RADIUS analyzer.

**Product**

GGSN

PDSN

P-GW

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] radius error`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match for errors in RADIUS packets and other errors in RADIUS analyzer.

**Example**

The following command defines a rule expression to match user traffic based on RADIUS error status of **TRUE**:

```
radius error = TRUE
```

# radius state

This command allows you to define rule expressions to match the current state of an RADIUS session.

**Product**

GGSN

PDSN

P-GW

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] radius state** *operator radius_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***radius_state***

Specifies the RADIUS state to match.

*radius_state* must be one of the following:

- **auth-req-rcvd**: Analyzer received the Access-Request message from the client.
- **auth-rsp-fail**: Analyzer received the Access-reject message from the server.
- **auth-rsp-success**: Analyzer received the Access-Accept message from the server as a reply to Access-request.

**Usage Guidelines**

Use this command to define rule expressions to match the current state of an RADIUS session.

**Example**

The following command defines a rule expression to match RADIUS current state **close**:

```
radius state = close
```

# rtcp any-match

This command allows you to define rule expressions to match all Real-Time Transport Control Protocol (RTCP) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtcp any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **TRUE**: The rule matches any RTCP traffic.
- **FALSE**: The rule does not match any RTCP traffic.

**Usage Guidelines**

Use this command to define rule expressions to match all RTCP packets.

**Example**

The following command defines a rule expression to match all RTCP packets:

```
rtcp any-match = TRUE
```

# rtcp jitter

This command allows you to define rule expressions to match the jitter parameter in RTCP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtcp jitter`** *operator jitter*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***jitter***

Specifies the RTCP inter-arrival jitter value (in milliseconds) to match.

*jitter* must be an integer from 0 through 4294967295.

**Usage Guidelines**

Use this command to define rule expressions to match jitter parameter found in the RTCP sender report or receiver report packets.

**Example**

The following command matches packets for jitter greater than or equal to 1295 milliseconds:

```
rtcp jitter >= 1295
```

# rtcp parent-proto

This command allows you to define rule expressions to match the parent protocol of the RTCP flow.

**Important** This command is available only in 8.1 and 9.0 and later releases.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtcp parent-proto`** *operator parent_protocol*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***parent_protocol***

Specifies the RTCP parent protocol to match.

*parent_protocol* must be one of the following:

- **rtsp**: Real Time Streaming Protocol
- **sip**: Session Initiation Protocol

**Usage Guidelines** Use this command to define rule expressions to match user traffic based on the parent protocol of the RTCP flow.

#### Example

The following command defines a rule expression to match user traffic based on SIP being the parent protocol of the RTCP flow:

```
rtcp parent-proto = sip
```

# rtcp pdu-length

This command allows you to define rule expressions to match Protocol Data Unit (PDU) length of RTCP packets, (RTCP header + RTCP payload).

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] rtcp pdu-length** *operator pdu_length*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***pdu_length***

Specifies the RTCP length (in bytes) to match.

In 8.1 and later releases, *pdu_length* must be an integer from 1 through 65535.

In 8.0, *pdu_length* must be an integer from 1 through 2000.

**Usage Guidelines** Use this command to define rule expressions to match RTCP PDU length (header + payload) in bytes.

**Example**

The following command defines a rule expression to match user traffic based on an RTCP PDU length of *10000* bytes:

```
rtcp pdu-length = 10000
```

# rtcp rtsp-id

This command allows you to define rule expressions to match user traffic based on a Real-time Streaming Protocol (RTSP) ID associated with an RTCP flow.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] rtcp rtsp-id [ case-sensitive ]`** *`operator rtsp_id`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with

- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***rtsp_id***

Specifies the value to match.

*rtsp_id* must be an alphanumeric string of 1 through 32 characters.

**Usage Guidelines** Use this command to define rule expressions to match an RTSP ID associated with an RTCP flow.

**Example**

The following command defines a rule expression to match user traffic containing RTSP message ID of *test1*:

```
rtcp rtsp-id contains test1
```

# rtcp session-length

This command allows you to define rule expressions to match the total length of RTCP sessions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] rtcp session-length** *operator session_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal

- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***session_length***

Specifies the RTCP total session length (in bytes) to match.

In 8.1 and later releases, *session_length* must be an integer from 1 through 4000000000.

In 8.0, *session_length* must be an integer from 1 through 40000000.

**Usage Guidelines**

Use this command to define rule expressions to match RTCP total session length.

**Example**

The following command defines a rule expression to match user traffic for a total RTCP session length of *200000*:

```
rtcp session-length = 200000
```

# rtcp uri

This command allows you to define rule expressions to match URI associated with RTCP flows.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtcp uri [ case-sensitive ]`** *`operator uri`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***uri***

Specifies the URI to match.

*uri* must be an alphanumeric string of 1 through 127 characters and may include punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match URI associated with RTCP flow.

**Example**

The following command defines a rule expression to match user traffic for RTCP URI *rtsp://www.example.org*:

```
rtcp uri = rtsp://www.example.org
```

# rtp any-match

This command allows you to define rule expressions to match all Real-time Transport Protocol (RTP) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtp any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all RTP packets.

**Example**

The following command defines a rule expression to match all RTP packets:

```
rtp any-match = TRUE
```

# rtp parent-proto

This command allows you to define rule expressions to match the parent protocol of the RTP flow.

**Important**

This command is available only in 8.1 and in 9.0 and later releases.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtp parent-proto** *operator parent_protocol*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***parent_protocol***

Specifies the RTP parent protocol to match.

*parent_protocol* must be one of the following:

- **rtsp**: Real Time Streaming Protocol
- **sip**: Session Initiation Protocol

**Usage Guidelines** Use this command to define rule expressions to match user traffic based on the parent protocol of the RTP flow.

**Example**

The following command defines a rule expression to match user traffic with parent protocol of the RTP flow being SIP:

```
rtp parent-proto = sip
```

# rtp pdu-length

This command allows you to define rule expressions to match PDU length of RTP packets, equal to the RTP header + RTP payload.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] rtp pdu-length** *operator pdu_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***pdu_length***

Specifies the RTP PDU length (in bytes) to match.

In 8.1 and later releases, *pdu_length* must be an integer from 1 through 65535.

In 8.0, *pdu_length* must be an integer from 1 through 2000.

**Usage Guidelines**

Use this command to define rule expressions to match PDU length (header + payload) of RTP packets in bytes.

**Example**

The following command defines a rule expression to match an RTP PDU length of *1000* bytes:

```
rtp pdu-length = 1000
```

# rtp rtsp-id

This command allows you to define rule expressions to match RTSP ID associated with RTP flows.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtp rtsp-id [ case-sensitive ]** *operator rtsp_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***rtsp_id***

Specifies the RTSP ID to match.

*rtsp_id* must be an alphanumeric string of 1 through 32 characters.

**Usage Guidelines**

Use this command to define rule expressions to match RTSP ID associated with RTP flows.

**Example**

The following command defines a rule expression to match RTSP message ID of *test1*:

```
rtp rtsp-id contains test1
```

# rtp session-length

This command allows you to define rule expressions to match the total length of RTP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtp session-length** *operator session_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***session_length***

Specifies the RTP total session length (in bytes) to match.

In 8.1 and later releases, *session_length* must be an integer from 1 through 4000000000.

In release 8.0, *session_length* must be an integer from 1 through 40000000.

**Usage Guidelines**

Use this command to define rule expressions to match the RTP total session length. The session-length is calculated by adding together the "rtp pdu-length" values of all relevant packets.

**Example**

The following command defines a rule expression to match a total RTP session length of *200000*:

```
rtp session-length = 200000
```

# rtp uri

This command allows you to define rule expressions to match the media URI associated with RTP flows.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtp uri [ case-sensitive ]** *operator uri*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***uri***

Specifies the RTP URI to match.

*uri* must be an alphanumeric string of 1 through 127 characters. *uri* allows punctuation characters and excludes the "host" portion.

**Usage Guidelines**

Use this command to define rule expressions to match media URI associated with RTP flow.

**Example**

The following command defines a rule expression to match the RTP URI string *rtsp://www.example.org*:

```
rtp uri = rtsp://www.example.org
```

# rtsp any-match

This command allows you to define rule expressions to match all Real Time Streaming Protocol (RTSP) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] rtsp any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all RTSP packets.

**Example**

The following command defines a rule expression to match all RTSP packets:

```
rtsp any-match = TRUE
```

# rtsp content length

This command allows you to define rule expressions to match the content length field in RTSP header.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] rtsp content length`** *operator content_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***content_length***

Specifies the content length (in bytes) to match.

*content_length* must be an integer from 0 through 65535.

**Usage Guidelines** Use this command to define rule expressions to match "content length" field in RTSP headers.

**Example**

The following command defines a rule expression to match content length of *10000* in RTSP headers:

```
rtsp content length = 10000
```

# rtsp content type

This command allows you to define rule expressions to match the content type field in RTSP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] rtsp content type [ case-sensitive ]`** *operator content_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies the content type to match.

*content_type* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

**Usage Guidelines** Use this command to define rule expressions to match "content type" field in RTSP headers.

**Example**

The following command defines a rule expression to match RTSP content type *abc100*:

```
rtsp content type = abc100
```

# rtsp date

This command allows you to define rule expressions to match the date field in the RTSP message headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**`[ no ] rtsp date [ case-sensitive ]`** *`operator date`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *date*

Specifies the date in RTSP header to match.

*date* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match the "date" field in the RTSP message headers.

### Example

The following command defines a rule expression to match the date *12_04_2006* in RTSP message headers:

```
rtsp date = 12_04_2006
```

# rtsp previous-state

This command allows you to define rule expressions to match the previous state of RTSP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] rtsp previous-state`** *`operator rtsp_previous_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***rtsp_previous_state***

Specifies the previous state to match.

*rtsp_previous_state* must be one of the following:

- **init**
- **open**
- **play**
- **ready**
- **record**

**Usage Guidelines**

Use this command to define rule expressions to match the previous state of RTSP sessions.

**Example**

The following command defines a rule expression to match RTSP previous state **ready**:

**`rtsp previous-state = ready`**

# rtsp reply code

This command allows you to define rule expressions to match the return code in RTSP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtsp reply code`** *`operator reply_code`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***reply_code***

Specifies the RTSP reply code to match.

*reply_code* must be an integer from 100 through 599.

**Usage Guidelines**

Use this command to define rule expressions to match the return code in RTSP response.

**Example**

The following command defines a rule expression to match RTSP return code *302*:

```
rtsp reply code = 302
```

# rtsp request method

This command allows you to define rule expressions to match the method in RTSP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtsp request method`** *`operator request_method`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***request_method***

Specifies the RTSP request method to match.

*request_method* must be one of the following requests:

- **announce**
- **describe**
- **get-parameter**
- **options**
- **pause**
- **play**
- **record**
- **redirect**
- **set-parameter**
- **setup**

- **teardown**

**Usage Guidelines** Use this command to define rule expressions to match the method in RTSP responses.

**Example**

The following command defines a rule expression to match RTSP request method **announce**:

```
rtsp request method = announce
```

# rtsp request packet

This command allows you to define rule expressions to match all RTSP request messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] rtsp request packet`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **TRUE**: Is request
- **FALSE**: Is response

**Usage Guidelines** Use this command to define rule expressions to match all RTSP request messages.

### Example

The following command defines a rule expression to match all RTSP request messages:

```
rtsp request packet = TRUE
```

# rtsp rtp-seq

This command allows you to define rule expressions to match the "seq" field in the RTP-Info header of RTSP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtsp rtp-seq** *operator sequence_number*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *sequence_number*

Specifies the sequence number in the RTSP RTP-Info field to match.

*sequence_number* must be an alphanumeric string of 0 through 65535 characters in Normal Play Time (NPT) time format.

**Usage Guidelines**

Use this command to define rule expressions to match user traffic matching the "seq" field in the RTP-Info header of RTSP response for a PLAY request.

### Example

The following command defines a rule expression to match user traffic based on RTP-seq number *npt-12:34:59*:

```
rtsp rtp-seq = npt-12:34:59
```

# rtsp rtp-time

This command allows you to define rule expressions to match the "time" field in RTP-Info header of RTSP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtsp rtp-time** *operator time*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *time*

Specifies the time to match.

*time* must be an alphanumeric string of 1 through 2147483647 characters in Normal Play Time (NPT) time format.

**Usage Guidelines**

Use this command to define rule expressions to match the "time" field in the RTP-Info header of RTSP response for a PLAY request.

### Example

The following command defines a rule expression to match RTP timestamp of *20120123T153600Z*:

```
rtsp rtp-time = 20120123T153600Z
```

# rtsp rtp-uri

This command allows you to define rule expressions to match the URI field in the RTP-Info header of RTSP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtsp rtp-uri [ case-sensitive ]`** *`operator uri`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***uri***

Specifies the value to match with the URI in RTP-Info header of the RTSP message.

*uri* must be an alphanumeric string of 1 through 127 characters. *uri* allows punctuation characters and excludes the "host" portion.

**Usage Guidelines**

Use this command to define rule expressions to match the URI field in the RTP-Info header of the RTSP response for a PLAY request.

**Example**

The following command defines a rule expression to match user traffic based on RTP-URI string *rtsp://www.foo.com* in the RTP-info header of RTSP packet:

```
rtsp rtp-uri = rtsp://www.foo.com
```

# rtsp session-id

This command allows you to define rule expressions to match the session ID in RTSP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtsp session-id [ case-sensitive ]** *operator session_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with

- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***session_id***

Specifies the session ID to match.

*session_id* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the session ID in RTSP messages.

**Example**

The following command defines a rule expression to match the RTSP session ID *0123abc100*:

```
rtsp session-id = 0123abc100
```

# rtsp session-length

This command allows you to define rule expressions to match the total length of RTSP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtsp session-length** *operator session_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals

- =: Equals
- >=: Greater than or equals

***session_length***

Specifies the RTSP session length (in bytes) to match.

*session_length* must be an integer from 1 through 40000000.

**Usage Guidelines**

Use this command to define rule expressions to match the total length of RTSP sessions. That is, the sum of the "rtsp pdu-length" values of all relevant packets.

**Example**

The following command defines a rule expression to match RTSP session length of *3000000* bytes:

```
rtsp session-length = 3000000
```

# rtsp state

This command allows you to define rule expressions to match the current state of RTSP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtsp state** *operator current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state to match.

*current_state* must be one of the following:

- **end**
- **init**
- **open**
- **play**
- **ready**
- **record**

**Usage Guidelines**

Use this command to define rule expressions to match the current state of RTSP sessions.

**Example**

The following command defines a rule expression to match RTSP current state **init**:

```
rtsp state = init
```

# rtsp uri

This command allows you to define rule expressions to match URI in RTSP request message.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtsp uri [ case-sensitive ]`** *`operator uri`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***uri***

Specifies the URI to match.

*uri* must be an alphanumeric string of 1 through 127 characters. *uri* allows punctuation characters and excludes the "host" portion.

## Usage Guidelines

Use this command to define rule expressions to match URI in RTSP request.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 5: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |

| Regex Character | Description |
|---|---|
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match user traffic based on RTSP URI *rtsp://www.example.com:554/twister/audiotrack*:

```
rtsp uri = rtsp://www.example.com:554/twister/audiotrack
```

The following command defines a regex rule expression to match either of the following or similar values in the RTSP URI string: rtsp://pvs29p.cvf.fr:554/t1/live/Oui17, rtsp://pvs00p.cvf.fr:554/t1/live/Nrj12, rtsp://pvs90p.cvf.fr:554/t1/live/France24_fr.

```
rtsp uri regex
"rtsp://pvs([0-9][0-9])p.cvf.fr:554/t1/live/(Gulli|Tf1|Tmc|Nrj12|Star|France24_fr|Oui17)*"
```

# rtsp uri sub-part

This command allows you to define rule expressions to match user traffic by parsing sub-parts of the URI in an RTSP request message.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] rtsp uri sub-part { { absolute-path | host | query } [ case-sensitive ]** *operator string* **| port {** *port_operator port_value* **| { range | !range }** *range_from* **to** *range_to* **} }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**absolute-path**

Specifies the absolute path matching criteria to RTSP URI in an RTSP request message.

**host**

Specifies the host name matching criteria to RTSP URI in an RTSP request message.

**query**

Specifies the query string matching criteria to RTSP URI in an RTSP request message.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***string***

Specifies the absolute path/host name or query string to match with the URI in RTSP header.

*string* must be an alphanumeric string of 1 through 127 characters. *string* allows punctuation characters and excludes the "host" portion.

**port**

Specifies the port related matching for RTSP URI in an RTSP request message.

***port_operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***port_value***

Specifies the RTSP port number to match with port rule in the RTSP flow as an integer from 0 through 65535.

**{ range | !range }** ***range_from*** **to** ***range_to*** **}**

Enables or disables the range criteria for RTSP flow ports.

- **range**: Enables the range criteria for RTSP flow ports.
- **!range**: Disables the range criteria for RTSP flow ports.
- *range_from*: Specifies the start of range of RTSP flow ports as an integer from 0 through 65535, but less than or equal to *range_to*.
- *range_to*: Specifies the end of range of RTSP flow ports as an integer from 0 through 65535, but more than or equal to *range_from*.

**Usage Guidelines** Use this command to define rule expressions to match URI sub parts like host, absolute path, port, and query in RTSP request messages.

### Example

The following command defines a URI sub part rule expression to analyze user traffic based on an RTSP URI port number between *1023* and *1068*:

```
rtsp uri sub-part port range 1023 to 1068
```

# rtsp user-agent

This command allows you to define rule expressions to match the user-agent field in RTSP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] rtsp user-agent [ case-sensitive ]`** *`operator user_agent`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with

- **starts-with**: Starts with

***user_agent***

Specifies the user agent to match.

*user_agent* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the "user-agent" field in RTSP header.

**Example**

The following command defines a rule expression to match *test* in "user-agent" field of RTSP header:

```
rtsp user-agent = test
```

# rtsp-stream any-match

This command allows you to define rule expressions to match all user traffic of type RTSP, RTCP, and RTP to achieve an unified charging for RTSP correlated flows.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] rtsp-stream any-match`** *`operator condition`*

**no**

If previously configured, deletes the rtsp-stream any match rule definition.

**operator**

Specifies how to logically match the information in the analyzed field.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

**condition**

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to specify a rule definition to analyze all RTSP, RTCP, and RTP traffic.

### Example

The following command defines a rule expression to match all RTSP, RTCP, and RTP user traffic:

```
rtsp-stream any-match = TRUE
```

# rtsp-stream first-setup-url

This command allows you to define rule expressions to match user traffic of type RTSP, RTCP, and RTP on the first setup URL of the parent RTSP flow to achieve an unified charging for RTSP correlated flows.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] rtsp-stream first-setup-url [ case-sensitive ]** *operator url*

### no

If previously configured, deletes the rtsp-stream any match rule definition.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to logically match the information in the analyzed field.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: contains

- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***url***

Specifies the URL to match.

*url* must be an alphanumeric string of 1 through 127 characters.

## Usage Guidelines

Use this command to specify a rule definition to analyze RTSP, RTCP, and RTP traffic based on the first setup URL of the parent RTSP flow.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 6: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |

| Regex Character | Description |
|---|---|
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match all RTSP, RTCP, and RTP traffic when the parent RTSP's first setup URL contains *cisco.com* :

```
rtsp-stream first-setup-url contains cisco.com
```

The following command defines a rule expression to match all RTSP, RTCP, and RTP traffic when the parent RTSP's first setup URL matches the given regular expression: *rtsp://tvs100.google.fr/t1/M6*

```
rtsp-stream first-setup-url regex
rtsp://tvs(a|l|b)[0-9][0-9].google.(fr|:554)/t1/(M6|W9_)*
```

# rule-application

This command allows you to specify the purpose of a ruledef, such as for charging, post-processing, routing, and so on.

**Product** ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

```
rule-application { charging | post-processing | routing | tpo }
no rule-application
```

### no

Disables the rule application configuration.

### charging

Specifies that the current ruledef is for charging purposes.

Up to 2,048 rule definitions can be defined for the charging application in an Active Charging Service.

Default: Enabled

### post-processing

**Important** The **post-processing** keyword is available only in 8.3 and later releases.

Specifies that the current ruledef is for post-processing purposes. This enables processing of packets even if the rule matching for them has been disabled.

### routing

Specifies that the current ruledef is for routing purposes. Up to 256 rule definitions can be defined for routing in an Active Charging Service. Default: Disabled

### tpo

**Important** The Traffic Performance Optimization (TPO) in-line service is not supported in this release.

## Usage Guidelines

Use this command to specify the rule application for a rule definition.

If, when configuring a ruledef, the rule-application is not specified, by default the system configures the ruledef as a charging ruledef.

### Example

The following command configures the rule application "charging" to the current rule definition:

```
rule-application charging
```

# sdp any-match

This command allows you to define rule expressions to match all packets that contain Session Description Protocol (SDP) descriptions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] sdp any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all packets containing SDP descriptions.

### Example

The following command defines a rule expression to match all packets containing SDP descriptions:

```
sdp any-match = TRUE
```

# sdp connection-ip-address

This command allows you to define rule expressions to match the IP address in the connection field of SDP descriptions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] sdp connection-ip-address`** *`operator ipv4_address`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***ipv4_address***

Specifies the IP address to match.

*ipv4_address* must be in IPv4 dotted-decimal notation.

**Usage Guidelines**

Use this command to define rule expressions to match IP address in the connection field of SDP descriptions.

**Example**

The following command defines a rule expression to match the IP address *10.1.1.1* in the connection field of SDP descriptions:

```
sdp connection-ip-address = 10.1.1.1
```

# sdp media-audio-port

This command allows you to define rule expressions to match media audio ports specified in the media sections of SDP descriptions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] sdp media-audio-port** *operator port*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***port***

Specifies the port number to match.

*port* must be an integer from 0 through 65535.

**Usage Guidelines** Use this command to define rule expressions to match media audio ports specified in the media sections of SDP descriptions.

**Example**

The following command defines a rule expression to match media audio port *100* in the media sections of SDP descriptions:

```
sdp media-audio-port = 100
```

# sdp media-video-port

This command allows you to define rule expressions to match media video ports specified in the media sections of SDP descriptions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**[ no ] sdp media-video-port** *operator port*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***port***

Specifies the port number to match.

*port* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match media video ports specified in the media sections of SDP descriptions.

**Example**

The following command defines a rule expression to match media video port *100* in the media sections of SDP descriptions:

```
sdp media-video-port = 100
```

# sdp uplink

This command allows you to define rule expressions to match SDP descriptions in the uplink (subscriber to network) direction.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] sdp uplink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Is not uplink
- **TRUE**: Is uplink

**Usage Guidelines**

Use this command to define rule expressions to match SDP descriptions in uplink direction.

**Example**

The following command defines a rule expression to match all SDP descriptions in the uplink direction:

```
sdp uplink = TRUE
```

# secure-http any-match

This command allows to match traffic analyzed by the Secure HTTP (HTTPS) analyzer in uplink or downlink direction.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] secure-http any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match traffic analyzed by the Secure HTTP (HTTPS) analyzer in uplink or downlink direction. The analysis does not differentiate between HTTPS and non-HTTP packets if the traffic is analyzed by HTTPS analyzer.

**Example**

The following command defines a rule expression to match HTTPS packets analyzed by the HTTPS analyzer:

```
secure-http any-match = TRUE
```

# secure-http uplink

This command allows you to define rule expressions to match uplink (subscriber to network) HTTPS packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] secure-http uplink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Is not uplink
- **TRUE**: Is uplink

**Usage Guidelines** Use this command to define rule expressions to match uplink HTTPS packets.

**Example**

The following command defines a rule expression to match all uplink HTTPS packets:

```
secure-http uplink = TRUE
```

# sip any-match

This command allows you to define rule expressions to match all Session Initiation Protocol (SIP) packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] sip any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all SIP packets.

**Example**

The following command defines a rule expression to match all SIP packets:

```
sip any-match = TRUE
```

# sip call-id

This command allows you to define rule expressions to match the Call ID in SIP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] sip call-id [ case-sensitive ]** *operator call_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***call-id***

Specifies the call ID to match.

*call-id* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the call ID in SIP messages.

**Example**

The following command defines a rule expression to match the call ID *test* in SIP messages:

```
sip call-id = test
```

# sip content length

This command allows you to define rule expressions to match the content-length field in SIP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] sip content length`** *`operator content_length`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***content_length***

Specifies the SIP content length to match.

*content_length* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match the content-length field in SIP headers.

**Example**

The following command defines a rule expression to match the content length *10000* in SIP headers:

```
sip content length = 10000
```

# sip content type

This command allows you to define rule expressions to match the content type field in SIP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] sip content type [ case-sensitive ]`** *`operator content_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies the content type to match.

*content_type* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the content type field in SIP headers.

**Example**

The following command defines a rule expression to match content type *download_string* in SIP headers:

```
sip content type = download_string
```

# sip from

This command allows you to define rule expressions to match the from field in SIP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**`[ no ] sip from [ case-sensitive ]`** *`operator string`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *string*

Specifies the value to match.

*string* must be an alphanumeric string of 1 through 127 characters, and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match the "from" field in SIP messages.

### Example

The following command defines a rule expression to match *test1* in the "from" field in SIP messages:

```
sip from contains test1
```

# sip previous-state

This command allows you to define rule expressions to match previous state of SIP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] sip previous-state** *operator sip_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***sip_previous_state***

Specifies the previous state to match.

*sip_previous_state* must be one of the following:

- **init**
- **provisional-response**
- **request-sent**
- **response-fail**
- **response-ok**

**Usage Guidelines**

Use this command to define rule expressions to match a previous state of SIP sessions.

**Example**

The following command defines a rule expression to match user traffic based on the SIP previous state of **request-sent**:

```
sip previous-state = request-sent
```

# sip reply code

This command allows you to define rule expressions to match the reply code in SIP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] sip reply code`** *`operator reply_code`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***reply_code***

Specifies the SIP reply code to match.

*reply_code* must be an integer from 100 through 699.

**Usage Guidelines**

Use this command to define rule expressions to match the reply code in SIP responses.

**Example**

The following command defines a rule expression to match *180* in the reply code in SIP responses:

```
sip reply code = 180
```

# sip request method

This command allows you to define rule expressions to match the method in SIP requests.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] sip request method`** *`operator method`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***method***

Specifies the SIP method to match.

*method* must be one of the following:

- **ack**
- **bye**
- **cancel**
- **info**
- **invite**
- **message**
- **notify**
- **options**
- **prack**
- **publish**

- **refer**
- **register**
- **subscribe**
- **update**

**Usage Guidelines** Use this command to define rule expressions to match the method in SIP requests.

### Example

The following command defines a rule expression to match the method **bye** in SIP request messages:

```
sip request method = bye
```

# sip request packet

This command allows you to define rule expressions to match all SIP request packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] sip request packet`** *`operator condition`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **=**: Equals
- **!=**: Does not equal

### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Is a response
- **TRUE**: Is a request

**Usage Guidelines**

Use this command to define rule expressions to match all SIP request packets.

**Example**

The following command defines a rule expression to match all SIP request packets:

```
sip request packet = TRUE
```

# sip state

This command allows you to define rule expressions to match current state of the SIP session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] sip state** *operator current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state to match.

*current_state* must be one of the following:

- **ack-received**
- **provisional-response**

- **request-sent**
- **response-fail**
- **response-ok**

**Usage Guidelines** Use this command to define rule expressions to match the current SIP session.

### Example

The following command defines a rule expression to match user traffic based on SIP current state **request-sent**:

```
sip state = request-sent
```

# sip to

This command allows you to define rule expressions to match the "to" field in SIP messages.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] sip to [ case-sensitive ]** *operator to_address*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with

- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***to_address***

Specifies the "to" address/name to match.

*to_address* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the "to" field in SIP messages.

**Example**

The following command defines a rule expression to match *test1* in the "to" field of SIP messages:

```
sip to contains test1
```

# sip uri

This command allows you to define rule expressions to match the URI in SIP messages.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
[ no ] sip uri [ sub-part { headers | host | parameters | port | userinfo
 } ] [ case-sensitive ] operator uri
```

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**sub-part { headers | host | parameters | port | userinfo }**

This is an optional keyword that defines what sub-part of a SIP URI to check.

- **headers**: Apply the rule to SIP URI header field.
- **host**: Apply the rule the SIP URI host field.
- **parameters**: Apply the rule to the SIP URI parameters field.

- **port**: Apply the rule to the SIP URI port field.
- **userinfo**: Apply the rule to the SIP URI userinfo field.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

The string for sub-part keyword **port** must be an integer and requires different operators. Use the following operators with the **port** keyword:

- **!=**: Does not equal
- **<=**: Is less than
- **=**: Equals
- **>=**: Is greater than

#### *uri*

Specifies the SIP URI to match.

*uri* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

The string for sub-part keyword **port** must be an integer from 0 through 65535.

### Usage Guidelines

Use this command to define rule expressions to match the URI in SIP messages.

#### Example

The following command defines a rule expression to match the URI string *sip:10.1.1.1:5060* in SIP messages:

```
sip uri = sip:10.1.1.1:5060
```

The following command defines a rule expression to match the URI string *sip:nnnn@host:5060;user=phone* in SIP messages:

```
sip uri = sip:nnnn@host:5060;user=phone
```

# smtp any-match

This command allows you to define rule expressions to match all Simple Mail Transfer Protocol (SMTP) packets.

## Product

ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] smtp any-match** *operator condition*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

## Usage Guidelines

Use this command to define rule expressions to match all SMTP packets.

### Example

The following command defines a rule expression to match all SMTP packets:

```
smtp any-match = TRUE
```

# smtp command arguments

This command allows you to define rule expressions to match SMTP command arguments.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] smtp command arguments [ case-sensitive ]`** *`operator argument`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***argument***

Specifies the command argument to match.

*argument* must be an alphanumeric string of 1 through 63 characters and may contain punctuation characters.

**Usage Guidelines** Use this command to define rule expressions to match SMTP command arguments.

**Example**

The following command defines a rule expression to match SMTP command argument *test*:

```
smtp command arguments = test
```

# smtp command id

This command allows you to define rule expressions to match SMTP command IDs.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] smtp command id** *operator command_id*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***command_id***

Specifies the command argument to match.

*command_id* must be an integer from 0 through 10.

**Usage Guidelines** Use this command to define rule expressions to match SMTP command IDs.

### Example

The following command defines a rule expression to match SMTP command ID *8*:

```
smtp command id = 8
```

# smtp command name

This command allows you to define rule expressions to match commands sent in SMTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp command name`** *operator command_name*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *command_name*

Specifies the command name to match.

*command_name* must be one of the following:

- **bdat**
- **data**
- **ehlo**
- **expn**
- **helo**
- **mail-from**

- **noop**
- **quit**
- **rcpt-to**
- **rset**
- **vrfy**

**Usage Guidelines** Use this command to define rule expressions to match commands sent in SMTP packets.

### Example

The following command defines a rule expression to match **data** command in SMTP packets:

```
smtp command name = data
```

# smtp mail-size

This command allows you to define rule expressions to match the size of mail sent by a SMTP client.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] smtp mail-size {** *operator mail_size* **| { { range | !range }** *range_from* **to** *range_to* **} }**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***mail_size***

Specifies the mail size (in bytes) to match.

*mail_size* must be an integer from 1 through 40000000.

**{ range | !range }** ***range_from*** **to** ***range_to***

Enables or disables the range criteria.

- **range**: Enables the range criteria.
- **!range**: Disables the range criteria.
- *range_from*: Specifies the start of range as an integer from 1 through 40000000.
- *range_to*: Specifies the end range. *range_to* must be an integer from 1 through 40000000, and must be greater than *range_from*.

**Usage Guidelines**

Use this command to define rule expressions to match the size of mail sent by an SMTP client.

**Example**

The following command defines a rule expression to match mail size of *40000* bytes:

```
smtp mail-size = 40000
```

# smtp pdu-length

This command allows you to define rule expressions to match the Protocol Data Unit (PDU) length of SMTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp pdu-length {`** *`operator pdu_length`* **`| { { range | !range }`** *`range_from`* **`to`** *`range_to`* **`} }`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***pdu_length***

Specifies the SMTP PDU length (in bytes) to match.

*pdu_length* must be an integer from 1 through 65535.

**{ range | !range }** ***range_from*** **to** ***range_to***

Enables or disables the range criteria.

- **range**: Enables the range criteria.
- **!range**: Disables the range criteria.
- *range_from*: Specifies the start of range as an integer from 1 through 65535.
- *range_to*: Specifies the end range. *range_to* must be an integer from 1 through 65535, and must be greater than *range_from*.

**Usage Guidelines** Use this command to define rule expressions to match PDU length of SMTP packets, that is headers + payload.

**Example**

The following command defines a rule expression to match a PDU length of *1600* bytes:

```
smtp pdu-length = 1600
```

# smtp previous-state

This command allows you to define rule expressions to match previous state of SMTP command sessions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] smtp previous-state** *operator smtp_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***smtp_previous_state***

Specifies the previous state to match.

*smtp_previous_state* must be one of the following:

- **close**: Closed state
- **init**: Initialized state
- **response-error**: Reply error state
- **response-ok**: Response ok state
- **waiting-for-response**: Waiting for response state

**Usage Guidelines**

Use this command to define rule expressions to match a previous state of SMTP command sessions.

**Example**

The following command defines a rule expression to match user traffic based on SMTP previous state **close**:

```
smtp previous-state = close
```

# smtp recipient

This command allows you to define rule expressions to match the recipient e-mail ID in the current SMTP transaction.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp recipient [ case-sensitive ]`** *`operator argument`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***argument***

Specifies the response argument to match.

*argument* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the recipient e-mail ID in the current SMTP transaction.

**Example**

The following command defines a rule expression to match recipient e-mail ID containing *test* in the current SMTP transaction:

```
smtp recipient contains test
```

# smtp reply arguments

This command allows you to define rule expressions to match the arguments within SMTP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] smtp reply arguments [ case-sensitive ]** *operator argument*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *argument*

Specifies the reply argument to match.

*argument* must be an alphanumeric string of 1 through 63 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match the arguments with SMTP response.

### Example

The following command defines a rule expression to match reply argument *forward-path* in SMTP response:

```
smtp reply arguments = forward-path
```

# smtp reply id

This command allows you to define rule expressions to match reply ID assigned to SMTP responses.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp reply id`** *`operator reply_id`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***reply_id***

Specifies the reply ID to match.

*reply_id* must be one of the following:

- **0**: +NO reply
- **1**: +OK reply
- **2**: -ERR reply

**Usage Guidelines**

Use this command to define rule expressions to reply ID assigned to SMTP response.

**Example**

The following command defines a rule expression to match reply ID *2* assigned to SMTP response:

```
smtp reply id = 2
```

# smtp reply status

This command allows you to define rule expressions to match the reply status in SMTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp reply status`** *`operator reply_status`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***reply_status***

Specifies the SMTP reply status to match.

*reply_status* must be one of the following:

- **+OK**: Response OK
- **-ERR**: Response error

**Usage Guidelines**

Use this command to define rule expressions to match reply status in SMTP packets.

### Example

The following command defines a rule expression to match reply status **+OK** in SMTP packets:

```
smtp reply status = +OK
```

# smtp sender

This command allows you to define rule expressions to match sender e-mail ID in the current SMTP transaction.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp sender [ case-sensitive ]`** *`operator sender`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***sender***

Specifies the sender value to match.

*sender* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match sender e-mail ID in the current SMTP transaction.

### Example

The following command defines a rule expression to match sender e-mail ID containing *test* in the current SMTP transaction:

```
smtp sender contains test
```

# smtp session-length

This command allows you to define rule expressions to match total length of SMTP sessions.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] smtp session-length { `***`operator session_length`*** **` | { range | !range } `**
***`range_from`*** **`to`** ***`range_to`*** **`}`**

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *session_length*

Specifies the session length to match.

*session_length* must be an integer from 1 through 40000000.

#### { range | !range } *range_from* to *range_to*

Enables or disables the range criteria.

- **range**: Enables the range criteria.
- **!range**: Disables the range criteria.
- *range_from*: Specifies the start of range as an integer from 1 through 40000000.
- *range_to*: Specifies the end range. *range_to* must be an integer from 1 through 40000000, and must be greater than *range_from*.

**Usage Guidelines** Use this command to define rule expressions to match total length of SMTP session.

### Example

The following command defines a rule expression to match SMTP session length of *4000000*:

```
smtp session-length = 4000000
```

# smtp state

This command allows you to define rule expressions to match current state of a SMTP command session.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] smtp state** *operator current_state*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *current_state*

Specifies the current state to match.

*current_state* must be one of the following:

- **close**: Closed state
- **init**: Initialized state
- **response-error**: Response of error state
- **response-ok**: Response of ok state
- **waiting-for-response**: Waiting for response state

**Usage Guidelines**

Use this command to define rule expressions to match current state of SMTP command session.

**Example**

The following command defines a rule expression to match current state as **close** of SMTP command session:

```
smtp state = close
```

# tcp analyzed out-of-order

This command allows you to define rule expressions to determine whether the received TCP packet was received before all of the earlier sequenced packets have been received. This functionality is for whether the packet was analyzed or discarded because the earlier sequenced packet(s) was (were) not received before a timeout expired.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tcp analyzed out-of-order`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Not analyzed
- **TRUE**: Analyzed

**Usage Guidelines**

This command is used to set the status flag to 'analyzed' or 'not analyzed' for all TCP packets received at the ACSMgr/SessMgr prior to their earlier packets.

When a packet reaches ACSMgr/SessMgr prior to earlier packet(s), it and subsequent packets are buffered at ACSMgr/SessMgr as TCP out-of-order packets and ACSMgr/SessMgr waits for missing packet(s) until the time-out duration expires. If the packet(s) with the missing sequence number(s) arrives within the time-out duration, all buffered packets with the correct sequence will be presented to upper layers (HTTP etc.) for analysis; otherwise buffered TCP out-of-order packets will be sent to charging with analysis done flag at the TCP/IP layer only.

If this command is enabled the TCP out-of-order packets are marked and sent to TCP analyzer as analyzed for charging action, otherwise they are discarded.

**Example**

The following command sets to analyze TCP out-of-order packets:

```
tcp analyzed out-of-order = TRUE
```

# tcp any-match

This command allows you to define rule expressions to match all TCP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tcp any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Not analyzed
- **TRUE**: Analyzed

**Usage Guidelines**

Use this command to define rule expressions to match all TCP packets.

**Example**

The following command defines a rule expression to match all TCP packets:

```
tcp any-match = TRUE
```

# tcp client-port

This command allows you to define rule expressions to match client port number in TCP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tcp client-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal

- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**range | !range**

Specifies the range criteria.

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match a client port number in TCP headers.

**Example**

The following command defines a rule expression to analyze user traffic matching TCP client port *5000*:

```
tcp client-port = 5000
```

# tcp connection-initiator

This command allows you to define rule expressions to match the TCP connection initiator.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tcp connection-initiator`** *`operator`* **`subscriber`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

**subscriber**

Specifies that the connection is being initiated by the subscriber.

**Usage Guidelines**

Use this command to define rule expressions to match the TCP connection initiator, and to allow the operator to differentiate when the connection initiated by subscriber or the subscriber is acting as a Transaction Control Server (TCS) server.

**Example**

The following command defines a rule expression to match user traffic based on TCP connection initiator **subscriber**:

```
tcp connection-initiator = subscriber
```

# tcp downlink

This command allows you to define rule expressions to match downlink (network to subscriber) TCP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tcp downlink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match downlink (to subscriber) TCP packets.

**Example**

The following command defines a rule expression to match downlink TCP packets:

```
tcp downlink = TRUE
```

# tcp dst-port

This command allows you to define rule expressions to match destination port number in TCP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] tcp dst-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**range | !range**

Specifies the range criteria:

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the range of destination TCP ports.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match destination port number in TCP headers.

**Example**

The following command defines a rule expression to match destination port number *10* in TCP headers:

```
tcp dst-port = 10
```

# tcp duplicate

This command allows you to define rule expressions to match TCP retransmissions.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] tcp duplicate`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Not duplicated/retransmitted
- **TRUE**: Duplicated/retransmitted

**Usage Guidelines** Use this command to specify rule expressions to match TCP retransmission.

**Example**

The following command defines a rule expression to match TCP retransmissions:

```
tcp duplicate = TRUE
```

# tcp either-port

This command allows you to define rule expressions to match either a destination or source port number in TCP headers.

**Product** ACS

## Privilege

Security Administrator, Administrator

## Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] tcp either-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

### *port_number*

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

### range | !range

Specifies the range criteria:

- **!range**: Not in the range
- **range**: In the range

### *start_range* to *end_range*

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

### port-map *port_map_name*

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match either a destination or source port number in TCP headers.

This command expression allows you to create a single ruledef using either-port, rather than needing two ruledefs (one with dst-port and one with src-port).

### Example

The following command defines a rule expression to match destination/source port number *10* in TCP header:

```
tcp either-port = 10
```

# tcp error

This command allows you to define rule expressions to identify errors, either in the packet (for example, TCP checksum error) or in the TCP analyzer's Finite State Machine (FSM).

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tcp error`** *`operator condition`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define a rule expression to identify errors, either in the packet (for example, TCP checksum error) or in the TCP analyzer's FSM.

### Example

The following command defines a rule expression to match TCP errors:

```
tcp error = TRUE
```

# tcp flag

This command allows you to define rule expressions to match bit within the flag field of TCP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] tcp flag** *operator flag*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!contains**: Does not contain
- **contains**: Contains
- **!=**: Does not equal
- **=**: Equals

### *flag*

Specifies the flag value to match.

*flag* must be one of the following:

- **ack**: TCP FLAG ACK
- **fin**: TCP FLAG FIN

- **push**: TCP FLAG PUSH
- **reset**: TCP FLAG RESET
- **syn**: TCP FLAG SYN

**Usage Guidelines** Use this command to define rule expressions to match a bit within the flag field of TCP headers.

### Example

The following command defines a rule expression to match **reset** within flag field of TCP headers:

```
tcp flag = reset
```

# tcp initial-handshake-lost

This command allows you to define rule expressions to match data packets when there has been no TCP handshaking to establish TCP connection.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] tcp initial-handshake-lost`** *`operator condition`*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**

- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match data packets when there has been no TCP handshaking to establish TCP connection.

### Example

The following command defines a rule expression to identify TCP flow where the initial handshake was not seen:

```
tcp initial-handshake-lost = TRUE
```

# tcp payload

This command allows you to define rule expressions to match hexadecimal or ASCII string content in the payload protocol-signature field of the TCP payload.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] tcp payload starts-with { hex-signature** *hex_string* **| string-signature** *string* **}**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### hex-signature *hex_string*

Specifies hexadecimal protocol signature in payload field.

*hex_string* must be a dash-delimited list of hex data of size smaller than 32.

### string-signature *string*

Specifies protocol signature in payload field.

*string* must be an alphanumeric string of 1 through 32 characters.

**Usage Guidelines** Use this command to define rule expressions to match for Hex/ASCII string content in payload protocol-signature field.

This rule expression is useful for detecting certain applications.

### Example

The following command defines a rule expression to identify user traffic based on TCP protocol signature *tcp1*:

```
tcp payload starts-with string-signature tcp1
```

# tcp payload-length

This command allows you to define rule expressions to match the length of a TCP payload.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tcp payload-length`** *`operator payload_length`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***payload_length***

Specifies the TCP payload length to match.

*payload_length* must be an integer from 0 through 40000000.

**Usage Guidelines**

Use this command to define rule expressions to match length of TCP payload, excluding the TCP or lower layer headers.

To match TCP control packets configure a payload-length of 0 (zero).

### Example

The following command defines a rule expression to match TCP payload length of *10000*:

```
tcp payload-length = 10000
```

# tcp previous-state

This command allows you to define rule expressions to match previous state of TCP connections.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tcp previous-state`** *`operator tcp_previous_state`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *tcp_previous_state*

Specifies the previous state to match.

*tcp_previous_state* must be one of the following:

- **close**
- **close-wait**
- **closing**
- **established**
- **fin-wait1**
- **fin-wait2**

- **last-ack**
- **listen**
- **syn-received**
- **syn-sent**
- **time-wait**

**Usage Guidelines** Use this command to define rule expressions to match a TCP previous state.

### Example

The following command defines a rule expression to match user traffic based on previous state **time-wait**:

```
tcp previous-state = time-wait
```

# tcp proxy-prev-state

This command allows you to define rule expressions to match TCP previous state on the ingress side of the TCP proxy.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] tcp proxy-prev-state** *operator previous_state*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***previous_state***

Specifies the previous state to match.

*previous_state* must be one of the following:

- **close**
- **close-wait**
- **closing**
- **established**
- **fin-wait1**
- **fin-wait2**
- **last-ack**
- **listen**
- **syn-received**
- **syn-sent**
- **time-wait**

**Usage Guidelines**

If there is no TCP proxy configured, this configuration is not applicable.

For proxy-enabled flows, TCP state handling interprets the ingress side as the radio side and the egress side as the Internet side of the TCP connection.

**tcp state** and **tcp prev-state** is the state of the client stack, which would be either the state of the subscriber's stack (if flow is not proxy enabled) or the MS state of proxy on the egress-side (if flow is proxy-enabled).

**tcp proxy-state** and **tcp proxy-prev-state** is the state of the embedded TCP proxy server, that is the proxy ingress-side.

So, depending on the use case, if using **tcp state** and **tcp prev-state** an existing configuration may work fine regardless of whether proxy is enabled. For other use cases, other ruledefs may have to be created.

Both **tcp state** and **tcp proxy-state** can be used in the same ruledef. If proxy was being used, they would map to the egress-side and ingress-side, respectively. If proxy was not being used, then this would not match ruledef because proxy state would not be applicable.

**Example**

The following command defines a rule expression to match user traffic based on TCP proxy previous state of established:

```
tcp proxy-prev-state = established
```

# tcp proxy-state

This command allows you to define rule expressions to match the TCP state on the ingress side of the TCP proxy.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tcp proxy-state** *operator state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***state***

Specifies the state to match.

*state* must be one of the following:

- **close**
- **close-wait**
- **closing**
- **established**
- **fin-wait1**
- **fin-wait2**
- **last-ack**
- **listen**
- **syn-received**
- **syn-sent**
- **time-wait**

**Usage Guidelines**

If there is no TCP proxy configured, this configuration is not applicable.

For proxy-enabled flows, TCP state handling interprets the ingress side as the radio side and the egress side as the Internet side of the TCP connection.

**tcp state** and **tcp prev-state** is the state of the client stack, which would be either the state of the subscriber's stack (if flow is not proxy enabled) or the MS state of proxy on egress-side (if flow is proxy-enabled).

**tcp proxy-state** and **tcp proxy-prev-state** is the state of the embedded TCP proxy server, that is the proxy ingress-side.

So, depending on the use case, if using **tcp state** and **tcp prev-state** an existing configuration may work fine regardless of whether proxy is enabled. For other use cases, other ruledefs may have to be created.

Both **tcp state** and **tcp proxy-state** can be used in the same ruledef. If proxy was being used, they would map to the egress-side and ingress-side, respectively. If proxy was not being used, then this would not match the ruledef because proxy state would not be applicable.

#### Example

The following command defines a rule expression to match user traffic based on TCP proxy previous state of established:

```
tcp proxy-state = established
```

# tcp server-port

This command allows you to define rule expressions to match server port number in TCP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tcp server-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals

- =: Equals
- >=: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**range | !range**

Specifies the range criteria:

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match server port number in TCP headers.

**Example**

The following command defines a rule expression to analyze user traffic matching TCP server port *10*:

```
tcp server-port = 10
```

# tcp session-length

This command allows you to define rule expressions to match the total length of a TCP session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tcp session-length** *operator session_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***session_length***

Specifies the TCP session length (in bytes) to match as be an integer from 0 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match the total length of a TCP session.

The session-length is calculated by adding together the TCP payload-length values of all relevant packets.

**Example**

The following command defines a rule expression to match user traffic based on TCP session length of *2000* bytes:

```
tcp session-length = 2000
```

# tcp src-port

This command allows you to define rule expressions to match source a port number in TCP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] tcp src-port { *operator port_number* | { !range | range } { *start_range* to *end_range* | port-map *port_map_name* } }**

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

### *port_number*

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

### range | !range

Specifies the range criteria:

- **!range**: Not in the range
- **range**: In the range

### *start_range* to *end_range*

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

### port-map *port_map_name*

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

## Usage Guidelines

Use this command to define rule expressions to match source a port number in TCP headers.

### Example

The following command defines a rule expression to analyze user traffic matching TCP source port *10*:

```
tcp src-port = 10
```

# tcp state

This command allows you to define rule expressions to match current state of TCP connections.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tcp state** *operator current_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state to match.

*current_state* must be one of the following:

- **close**
- **close-wait**
- **closing**
- **established**
- **fin-wait1**
- **fin-wait2**
- **last-ack**
- **listen**
- **syn-received**
- **syn-sent**

- **time-wait**

**Usage Guidelines** Use this command to define rule expressions to match a current state of TCP connections.

**Example**

The following command defines a rule expression to match user traffic based on current state **close**:

```
tcp state = close
```

# tcp uplink

This command allows you to define rule expressions to match uplink (subscriber to network) TCP packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] tcp uplink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to uplink TCP packets.

### Example

The following command defines a rule expression to uplink TCP packets:

```
tcp uplink = TRUE
```

# tethering-detection

This command allows you to define rule expressions to match tethered or non-tethered flows.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

```
tethering-detection [ application | dns-based | ip-ttl | os-ua ] {
flow-not-tethered | flow-tethered }
no tethering-detection
```

#### no

Deletes the tethering detection configuration from the ruledef.

#### application

Specifies to select flows that were tethered or non-tethered based on App-based detection solution.

With release 21.1.3, the App-based Tethering Detection is introduced only for Netflix and YouTube.

#### dns-based

Specifies to select flows that were tethered or non-tethered based on DNS-based detection solution.

#### ip-ttl

Specifies to select flows that were tethered or non-tethered as per IP-TTL values.

#### os-ua

Specifies to select flows that were tethered or non-tethered as per OS-UA lookups.

In 18 and later releases, IPv6 OS-based tethering detection is supported.

#### flow-not-tethered

Specifies to match if tethering is not detected on flow.

#### flow-tethered

Specifies to match if tethering is detected on flow.

### Usage Guidelines

Use this command to define rule expressions to match tethered/non-tethered flows.

Note that in order for the rule containing the tethering-detection configuration to get matched, at least one valid rule line has to be present in it.

This configuration is treated in a special manner by the rule matching engine in that it is excluded from the condition **multi-line-or all-lines**. For example, if there are three rule-lines in a ruledef and multi-line-or is enabled as follows:

```
ruledef all-tethered-web-traffic
     http any-match = TRUE
     wsp any-match = TRUE
     multi-line-or all-lines
     tethering-detection flow-tethered
     exit
```

In this case, if for a packet only the rule line **tethering-detection flow-tethered** matches, it is not sufficient to result in a rule match even though **multi-line-or all-lines** is enabled in the ruledef.

#### Example

The following command defines a rule expression to match tethered flows:

**tethering-detection flow-tethered**

# tftp any-match

This command allows you to define rule expressions to match all Trivial File Transfer Protocol (TFTP) packets.

### Product

ACS

### Privilege

Security Administrator, Administrator

### Command Modes

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

### Syntax Description

**[ no ] tftp any-match** *operator condition*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Not analyzed
- **TRUE**: Analyzed

**Usage Guidelines**

Use this command to define rule expressions to match all TFTP packets.

**Example**

The following command defines a rule expression to match all TFTP packets:

```
tftp any-match = TRUE
```

# tftp data-any-match

This command allows you to define rule expressions to match all TFTP data packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] tftp data-any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**: Not analyzed
- **TRUE**: Analyzed

**Usage Guidelines**

Use this command to define rule expressions to match all TFTP data packets.

**Example**

The following command defines a rule expression to match all TFTP data packets:

```
tftp data-any-match = TRUE
```

# tls

This command allows to configure TLS/SSL Server Name Indication (SNI) and corresponding custom defined protocol (CDP).

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] tls { set-app-proto** *cdp_name_string* **| sni** *operator server_name_string* **}**

**no**

If previously configured, deletes the configuration in the current ruledef.

**set-app-proto *cdp_name_string***

Specifies the name of the custom defined protocol (CDP) for TLS/SSL flows matching the ruledef.

*cdp_name_string* must be an alphanumeric string of 1 through 19 characters.

**sni *operator server_name_string***

Specifies the TLS/SSL Server Name Indication (SNI) field value in the Client Hello packet.

*operator*: Specifies how to match and must be one of the following:

- **!=**: Does not equal

- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

*server_name_string*: Specifies the server name and must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to configure the TLS/SSL SNI and corresponding CDP. The CDP name for a TLS/SSL flow must match a set of SNI rule lines in multiline-and or multiline-or manner.

**Example**

The following command configures the SNI to *facebook.com*:

```
tls sni = facebook.com
```

The following command configures the name of the corresponding protocol to *facebook*:

```
tls set-app-proto facebook
```

# udp any-match

This command allows you to define rule expressions to match all UDP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] udp any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match all UDP packets.

**Example**

The following command defines a rule expression to match all UDP packets:

```
udp any-match = TRUE
```

# udp client-port

This command allows you to define rule expressions to match client port number in UDP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] udp client-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- <=: Lesser than or equals
- =: Equals
- >=: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**range | !range**

Specifies the range criteria.

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match client port number in UDP headers.

**Example**

The following command defines a rule expression to analyze user traffic matching UDP client port *500*:

```
udp client-port = 500
```

# udp downlink

This command allows you to define rule expressions to match downlink (network to subscriber) UDP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] udp downlink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match downlink UDP packets.

**Example**

The following command defines a rule expression to match downlink UDP packets:

```
udp downlink = TRUE
```

# udp dst-port

This command allows you to define rule expressions to match destination port number in UDP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] udp dst-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**!range | range**

Specifies the range criteria.

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match destination port number in UDP headers.

**Example**

The following command defines a rule expression to match user traffic based on destination port number *10*:

```
udp dst-port = 10
```

# udp either-port

This command allows you to define rule expressions to match either a destination or source port number in UDP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] udp either-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**!range | range**

Specifies the range criteria.

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

*start_range* must be an integer from 1 through 65535.

*end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match either destination or source port number in UDP headers.

**Example**

The following command defines a rule expression to match user traffic based on match either source/destination port number *10*:

```
udp either-port = 10
```

# udp payload starts-with

This command allows you to define rule expressions to match hex/ASCII string content in UDP payload protocol-signature field.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] udp payload starts-with { hex-signature`** *`hex_string`* **`| string-signature`** *`string`* **`}`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**hex-signature *hex_string***

Specifies hexadecimal protocol signature in payload field.

*hex_string* must be a dash-delimited list of hex data of size smaller than 32.

**string-signature *string***

Specifies protocol signature in payload field.

*string* must be an alphanumeric string of 1 through 32 characters.

**Usage Guidelines**

Use this command to define rule expressions to match for Hex/ASCII string content in UDP payload protocol-signature field.

This rule expression is useful for detecting certain applications.

### Example

The following command defines a UDP rule expression to analyze user traffic based on UDP protocol signature *udp1*:

```
udp payload starts-with string-signature udp1
```

# udp server-port

This command allows you to define rule expressions to match server port number in UDP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] udp server-port {** *operator port_number* **| { !range | range } {** *start_range* **to** *end_range* **| port-map** *port_map_name* **} }**

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *port_number*

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

#### range | !range

Specifies the range criteria.

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

- *start_range* must be an integer from 1 through 65535.
- *end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match source a server port number in UDP headers.

**Example**

The following command defines a rule expression to analyze user traffic matching UDP server port *53*:

```
udp server-port = 53
```

# udp src-port

This command allows you to define rule expressions to match source port number in UDP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] udp src-port {`** *`operator port_number`* **`| { !range | range } {`** *`start_range`* **`to`** *`end_range`* **`| port-map`** *`port_map_name`* **`} }`**

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***port_number***

Specifies the port number to match.

*port_number* must be an integer from 1 through 65535.

**!range | range**

Specifies the range criteria.

- **!range**: Not in the range
- **range**: In the range

***start_range* to *end_range***

Specifies the starting and ending port numbers for the port range.

*start_range* must be an integer from 1 through 65535.

*end_range* must be an integer from 1 through 65535, and must be greater than *start_range*.

**port-map *port_map_name***

Specifies the port map for the port range. *port_map_name* must be an alphanumeric string of 1 through 63 characters.

**Usage Guidelines**

Use this command to define rule expressions to match source port number in UDP headers.

**Example**

The following command defines a rule expression to match source port number *10* in UDP headers:

```
udp src-port = 10
```

# udp uplink

This command allows you to define rule expressions to match uplink (subscriber to network) UDP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] udp uplink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- =: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match uplink UDP packets.

**Example**

The following command defines a rule expression to match uplink (from subscriber) UDP packets:

```
udp uplink = TRUE
```

# wsp any-match

This command allows you to define rule expressions to match all Wireless Session Protocol (WSP) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] wsp any-match`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to specify a rule expression to match all WSP packets.

**Example**

The following command defines a rule expression to match all WSP packets:

```
wsp any-match = TRUE
```

# wsp content type

This command allows you to define rule expressions to match the content type field in WSP headers.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **`[ no ] wsp content type [ case-sensitive ]`** *`operator content_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies content type to match.

*content_type* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines** Use this command to define rule expressions to match "content type" field in WSP headers.

**Example**

The following command defines a rule expression to WSP content type *test*:

```
wsp content type = test
```

# wsp domain

This command allows you to define rule expressions to match domain portion of the URI for WSP packets.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

## Syntax Description

**[ no ] wsp domain [ case-sensitive ]** *operator domain*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

### *domain*

Specifies the domain to match.

*domain* must be an alphanumeric string of 1 through 127 characters.

## Usage Guidelines

Use this command to define rule expressions to match the domain portion of URIs in WSP packets.

From the URL, after http:// (if present) is removed, everything until the first "/" is the domain.

### Example

The following command defines a rule expression to match user traffic based on domain name *testdomain*:

```
wsp domain = testdomain
```

# wsp downlink

This command allows you to define rule expressions to match downlink (network to subscriber) WSP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp downlink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the downlink (from the Mobile Node direction) status to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match downlink WSP packets.

### Example

The following command defines a rule expression to match downlink WSP packets:

```
wsp downlink = TRUE
```

# wsp first-request-packet

This command allows you to define rule expressions to match WSP first-request-packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wsp first-request-packet** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match the GET or POST request, if it is the first WSP request for the subscriber's session.

### Example

The following command defines a rule expression to match WSP first-request-packet:

```
wsp first-request-packet = TRUE
```

# wsp host

This command allows you to define rule expressions to match the host name header field in WSP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] wsp host [ case-sensitive ]`** *`operator host_name`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***host_name***

Specifies the WSP host name to match.

*host_name* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match host name header field in WSP headers.

### Example

The following command defines a rule expression to match host name *host1* in WSP headers:

```
wsp host contains host1
```

# wsp pdu-length

This command allows you to define rule expressions to match WSP PDU length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp pdu-length`** *`operator pdu_length`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *pdu_length*

Specifies the WSP PDU length (in bytes) to match.

*pdu_length* must be an integer from 1 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match WSP PDU length (header + payload) in bytes.

### Example

The following command defines a rule expression to match user traffic based on WSP PDU length of *10000* bytes:

```
wsp pdu-length = 10000
```

# wsp pdu-type

This command allows you to define rule expressions to match WSP PDU type in the current packet.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wsp pdu-type** *operator pdu_type*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *pdu_type*

Specifies the WSP PDU type to match.

*pdu_type* must be one of the following:

- **confirmed push**
- **connect-reply**
- **connect-request**
- **data-fragment**
- **delete**

- **disconnect**
- **get**
- **head**
- **options**
- **post**
- **push**
- **put**
- **redirect**
- **reply**
- **resume**
- **suspend**
- **trace**

**Usage Guidelines** Use this command to define rule expressions to match WSP PDU type value in current packet.

**Example**

The following command defines a rule expression to match WSP PDU type **resume**:

```
wsp pdu-type resume
```

# wsp previous-state

This command allows you to define rule expressions to match previous WSP method invocation state.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] wsp previous-state** *operator wsp_previous_state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***wsp_previous_state***

Specifies the previous state to match.

*wsp_previous_state* must be one of the following:

- **init**
- **response-error**
- **response-ok**
- **waiting-for-response**

**Usage Guidelines**

Use this command to define rule expressions to match WSP previous state.

**Example**

The following command defines a rule expression to match WSP previous state of *response-ok*:

```
wsp previous-state = response-ok
```

# wsp reply code

This command allows you to define rule expressions to match WSP reply code.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wsp reply code** *operator reply_code*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***reply_code***

Specifies the WSP reply code to match.

*reply_code* must be an integer from 0 through 101.

**Usage Guidelines**

Use this command to define rule expressions to match WSP reply code.

**Example**

The following command defines a rule expression to match WSP reply code of *50*:

```
wsp reply code = 50
```

# wsp session-length

This command allows you to define rule expressions to match total length of a WSP session.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wsp session-length** *operator session_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: less than equals
- **=**: Equals
- **>=**: greater than equals

***session_length***

Specifies the WSP session length (in bytes) to match.

*session_length* must be an integer from 1 through 65535.

**Usage Guidelines** Use this command to define rule expressions to match total length of WSP session.

**Example**

The following command defines a rule expression to match WSP session length of *2000* bytes:

```
wsp session-length = 2000
```

# wsp session-management

This command allows you to define rule expressions to match WSP Session Management state.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] wsp session-management { previous-state | state }** *operator state*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**previous-state**

Specifies the previous WSP Session Management state.

**state**

Specifies current WSP Session Management Finite State Machine (FSM) state.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***state***

Specifies the state to match.

For **previous-state**, *state* must be one of the following:

- **connected**
- **connecting**
- **init**
- **resuming**
- **suspended**

For **state**, *state* must be one of the following:

- **close**
- **connected**
- **connecting**
- **init**
- **resuming**
- **suspended**

**Usage Guidelines**

Use this command to define rule expressions to match a WSP Session Management state.

**Example**

The following command defines a rule expression to match previous WSP Session Management state of **connecting**:

```
wsp session-management previous-state = connecting
```

# wsp state

This command allows you to define rule expressions to match WSP Method Invocation state.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] wsp state`** *`operator current_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state to match.

*current_state* must be one of the following:

- **close**
- **response-error**
- **response-ok**
- **waiting-for-response**

**Usage Guidelines**

Use this command to define rule expressions to match WSP Method Invocation state.

**Example**

The following command defines a rule expression to match a WSP Method Invocation state **close**:

```
wsp state = close
```

# wsp status

This command has been deprecated. See the **wsp reply-code** command.

# wsp tid

This command allows you to define rule expressions to match Transaction Identifier (TID) field for connection-less WSP.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp tid`** *`operator transaction_id`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***transaction_id***

Specifies the transaction identifier to match.

*transaction_id* must be an integer from 0 through 255.

**Usage Guidelines**

Use this command to define rule expressions to match TID field for connection-less WSP.

**Example**

The following command defines a rule expression to match a TID value of *22* for connection-less WSP:

```
wsp tid = 22
```

# wsp total-length

This command has been deprecated. See the **wsp session-length** command.

# wsp transfer-encoding

This command allows you to define rule expressions to match transfer encoding present in WSP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp transfer-encoding [ case-sensitive ]`** *`operator transfer_encoding`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***transfer_encoding***

This must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match transfer encoding present in WSP header.

### Example

The following command defines a rule expression to match user traffic based on WSP transfer encoding *7*:

```
wsp transfer-encoding contains 7
```

# wsp uplink

This command allows you to define rule expressions to match uplink (subscriber to network) WSP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp uplink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the uplink (to the Mobile Node direction) status to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match uplink WSP packets.

### Example

The following command defines a rule expression to match uplink WSP packets:

```
wsp uplink = TRUE
```

# wsp url

This command allows you to define rule expressions to match WSP URL.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp url [ case-sensitive ]`** *`operator url`*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***url***

Specifies the URL to match.

*url* must be an alphanumeric string of 1 through 127 characters.

## Usage Guidelines

Use this command to define rule expressions to match the complete URL, including the host portion.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 7: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |

| Regex Character | Description |
|---|---|
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match user traffic based on WSP URL *wsp://wiki.tcl.tk*:

```
wsp url = wsp://wiki.tcl.tk
```

The following command defines a regex rule expression to match any of the following or similar values in the WSP URL string: wsp://home.opera.yahoo.com, wsp://dwld.yahoo.com, wsp://dwld2.yahoo.com.

```
wsp url regex "wsp://(dwld|opera|home.opera|dwld[1-3]).yahoo.com"
```

# wsp user-agent

This command allows you to define rule expressions to match user agent field in WSP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wsp user-agent [ case-sensitive ]`** *`operator user_agent`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***user_agent***

Specifies the WSP user agent to match.

*user_agent* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match a user agent field in WSP headers.

**Example**

The following command defines a rule expression to match value *test* in user agent field in WSP headers:

```
wsp user-agent contains test
```

# wsp x-header

This command allows you to define rule expressions to match WSP extension-headers (x-headers).

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wsp x-header** *name* **[ case-sensitive ]** *operator string*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***name***

Specifies the x-header value as an alphanumeric string of 1 through 31 characters.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***string***

Specifies the value of the extension header as an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to configure any x-header field in WSP and parse it. The extension-header mechanism allows additional header fields to be defined without changing the protocol. The extension-header can be any header fields that are not specified in the RFC standard.

### Example

The following command defines a rule expression to analyze user traffic containing WSP extension-header of *test_field* and value of *test_string*:

```
wsp x-header test_field = test_string
```

# wtp any-match

This command allows you to define rule expressions to match all Wireless Transaction Protocol (WTP) packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wtp any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all WTP packets.

### Example

The following command defines a rule expression to match all WTP packets:

```
wtp any-match = TRUE
```

# wtp downlink

This command allows you to define rule expressions to match downlink (network to subscriber) WTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wtp downlink** *operator condition*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *condition*

Specifies the downlink (from the Mobile Node direction) status to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match downlink WTP packets.

### Example

The following command defines a rule expression to match all downlink WTP packets:

```
wtp downlink = TRUE
```

# wtp gtr

This command allows you to define rule expressions to match Group Transmission (GTR) flag in the current WTP PDU.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wtp gtr** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match the GTR flag (that indicates the last packet of a packet group) in the current WTP PDU.

### Example

The following command defines a rule expression to match WTP user traffic based on WTP GTR:

```
wtp gtr = TRUE
```

# wtp pdu-length

This command allows you to define rule expressions to match WTP PDU length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wtp pdu-length`** *operator pdu_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***pdu_length***

Specifies the WTP PDU length (in bytes) to match.

*pdu_length* must be an integer from 1 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match WTP PDU length (header + payload) in bytes.

**Example**

The following command defines a rule expression to match WTP PDU length of *9647* bytes:

```
wtp pdu-length = 9647
```

# wtp pdu-type

This command allows you to define rule expressions to match WTP PDU type.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wtp pdu-type** *operator pdu_type*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***pdu_type***

Specifies the WTP PDU type to match.

*pdu_type* must be one of the following:

- **abort**
- **ack**
- **invoke**
- **negative-ack**
- **result**
- **segment-invoke**
- **segment-result**

**Usage Guidelines**

Use this command to define rule expressions to match WTP PDU type.

**Example**

The following command defines a rule expression to match the WTP PDU type **result**:

```
wtp pdu-type = result
```

# wtp previous-state

This command allows you to define rule expressions to match previous WTP state.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wtp previous-state`** *`operator wtp_previous_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***wtp_previous_state***

Specifies the previous state to match.

*wtp_previous_state* must be one of the following:

- **ack-sent**
- **init**
- **invoke-sent**
- **rcvd**
- **result-rcvd**

**Usage Guidelines**

Use this command to define rule expressions to match WTP previous state.

**Example**

The following command defines a rule expression to match user traffic based on WTP previous state of **ack-sent**:

```
wtp previous-state = ack-sent
```

# wtp rid

This command allows you to define rule expressions to match Re-transmission Indicator (RID) flag set in WTP traffic.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] wtp rid** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match WTP RID flag.

### Example

The following command defines a rule expression to match user traffic containing WTP RID flag:

```
wtp rid = TRUE
```

# wtp state

This command allows you to define rule expressions to match current WTP state.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] wtp state`** *`operator current_state`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***current_state***

Specifies the current state to match.

*current_state* must be one of the following:

- **ack-sent**
- **close**
- **init**
- **invoke-sent**
- **rcvd**
- **result-rcvd**

**Usage Guidelines**

Use this command to define rule expressions to match current WTP state.

### Example

The following command defines a rule expression to match user traffic based on current WTP state **close**:

```
wtp state = close
```

# wtp tid

This command allows you to define rule expressions to match WTP Transaction Identifier (TID).

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wtp tid`** *`operator transaction_id`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***transaction_id***

Specifies the transaction identifier to match.

*transaction_id* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match WTP TID. This expression ignores the high order bit in the protocol that indicates the direction.

### Example

The following command defines a rule expression to match user traffic containing WTP TID value of *22*:

```
wtp tid = 22
```

# wtp transaction class

This command allows you to define rule expressions to match WTP Transaction Class (TCL) state.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] wtp transaction class`** *`operator transaction_class`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

**operator** must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***transaction_class***

Specifies the WTP TCL to match.

*transaction_class* must be an integer from 0 through 2.

**Usage Guidelines**

Use this command to define rule expressions to match WTP transaction class.

**Example**

The following command defines a rule expression to match WTP traffic based on WTP transaction class *2*:

```
wtp transaction class = 2
```

# wtp ttr

This command allows you to define rule expressions to match WTP Trailer Transmission (TTR) flag.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wtp ttr`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match TTR flag (used to indicate the last packet in a segmented message) in the current WTP PDU.

### Example

The following command defines a rule expression to match WTP traffic based on the presence of the WTP TTR flag:

```
wtp ttr = TRUE
```

# wtp uplink

This command allows you to define rule expressions to match uplink (subscriber to network) WTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**`[ no ] wtp uplink`** *`operator condition`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match uplink WTP packets.

### Example

The following command defines a rule expression to match all uplink WTP packets:

```
wtp uplink = TRUE
```

# www any-match

This command allows you to define rule expressions to match all WWW packets. It is true for HTTP, WAP1.x, and WAP2.0 protocols.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www any-match** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines**

Use this command to define rule expressions to match all WWW packets. This expression is true for HTTP, WAP1.x, and WAP2.0 protocols

**Example**

The following command defines a rule expression to match all WWW packets:

```
www any-match = TRUE
```

# www content type

This command allows you to define rule expressions to match the Content-Type field of HTTP/WSP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-acs-ruledef)#`

**Syntax Description**

**`[ no ] www content type [ case-sensitive ]`** *`operator content_type`*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

**case-sensitive**

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

***operator***

Specifies how to match.

**operator** must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***content_type***

Specifies the value to match.

*content_type* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the "content type" field of HTTP/WSP header.

### Example

The following command defines a rule expression to match the WWW content type *Accept*:

```
www content type = Accept
```

# www domain

This command allows you to define rule expressions to match the domain portion of URIs in WSP/HTTP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www domain [ case-sensitive ]** *operator domain*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***domain***

Specifies the domain to match.

*domain* must be an alphanumeric string of 1 through 127 characters.

**Usage Guidelines**

Use this command to define rule expressions to match the domain portion of URIs in WSP/HTTP packets.

From the URL, after http:// (if present) is removed, everything until the first "/" is the domain.

**Example**

The following command defines a rule expression to match user traffic based on domain name *testdomain*:

```
www domain = testdomain
```

# www downlink

This command allows you to define rule expressions to match downlink (network to subscriber) HTTP/WSP packets.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www downlink** *operator condition*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

***condition***

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match downlink HTTP/WSP packets.

### Example

The following command defines a rule expression to match all downlink WWW packets:

```
www downlink = TRUE
```

# www first-request-packet

This command allows you to define rule expressions to match the GET or POST request, if it is the first WSP/HTTP request for the subscriber's session.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] www first-request-packet** *operator condition*

### no

If previously configured, deletes the specified rule expression from the current ruledef.

### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

### *condition*

Specifies the condition to match.

*condition* must be one of the following:

- **FALSE**
- **TRUE**

**Usage Guidelines** Use this command to define rule expressions to match the GET or POST request, if it is the first WSP/HTTP request for the subscriber's session.

### Example

The following command defines a rule expression to match user traffic based on the WWW first-request-packet:

```
www first-request-packet = TRUE
```

# www header-length

This command allows you to define rule expressions to match WWW packet header length.

**Product** ACS

**Privilege** Security Administrator, Administrator

**Command Modes** Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description** **[ no ] www header-length** *operator header_length*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *header_length*

Specifies the WWW packet header length (in bytes) to match, *header_length* must be an integer from 0 through 65535.

**Usage Guidelines** Use this command to define rule expressions to match WWW packet header length.

### Example

The following command defines a rule expression to match user traffic based on WWW packet header length of *10000* bytes:

```
www header-length = 10000
```

# www host

This command allows you to define rule expressions to match the "host name" header field present in HTTP/WSP headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www host [ case-sensitive ]** *operator host_name*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

***host_name***

Specifies the WWW host name to match.

*host_name* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the host name header field present in HTTP/WSP headers.

**Example**

The following command defines a rule expression to match user traffic based on WWW host name *host1*:

```
www host = host1
```

# www payload-length

This command allows you to define rule expressions to match WWW payload length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www payload-length** *operator payload_length*

**no**

If previously configured, deletes the specified rule expression from the current ruledef.

***operator***

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

***payload_length***

Specifies the payload length (in bytes) to match.

*payload_length* must be an integer from 1 through 4000000000.

**Usage Guidelines**

Use this command to define rule expressions to match WWW payload length.

### Example

The following command defines a rule expression to match user traffic based on WWW payload length of *10000*:

```
www payload-length = 10000
```

# www pdu-length

This command allows you to define rule expressions to match WWW PDU length.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www pdu-length** *operator pdu_length*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *pdu_length*

Specifies the WWW PDU length (in bytes) to match.

*pdu_length* must be an integer from 0 through 65535.

**Usage Guidelines**

Use this command to define rule expressions to match WWW PDU length (header + payload) in bytes.

### Example

The following command defines a rule expression to match user traffic based on WWW PDU length of *9767* bytes:

```
www pdu-length = 9767
```

# www previous-state

This command allows you to define rule expressions to match previous HTTP/WSP(HTTP) state.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www previous-state** *operator www_previous_state*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *www_previous_state*

Specifies the previous state to match.

*www_previous_state* must be one of the following:

- **init**
- **response-error**
- **response-ok**
- **waiting-for-response**

**Usage Guidelines**

Use this command to define rule expressions to match a previous HTTP/WSP(HTTP) state.

### Example

The following command defines a rule expression to match user traffic based on WWW previous state **init**:

```
www previous-state = init
```

# www reply code

This command allows you to define rule expressions to match WWW reply code arguments.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www reply code** *operator reply_code*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **<=**: Lesser than or equals
- **=**: Equals
- **>=**: Greater than or equals

#### *reply_code*

Specifies the reply code to match.

*reply_code* must be an integer from 100 through 599.

**Usage Guidelines**

Use this command to define rule expressions to match HTTP 1.1 status code, or WSP status code that has been remapped to the corresponding HTTP value.

WSP status codes 0 – 101 are automatically remapped to the HTTP status code values, as defined by Table 36 WAP-230-WSP Version 5.

### Example

The following command defines a rule expression to analyze WWW user traffic based on reply code of *125*:

```
www reply code = 125
```

# www state

This command allows you to define rule expressions to match current HTTP/WSP(HTTP) state.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www state** *operator current_state*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **=**: Equals

#### *current_state*

Specifies the current state to match.

*current_state* must be one of the following:

- **close**
- **response-error**
- **response-ok**
- **waiting-for-response**

**Usage Guidelines**

Use this command to define rule expressions to match current HTTP/WSP state.

### Example

The following command defines a rule expression to match user traffic based on the current WWW state **close**:

```
www state = close
```

# www transfer-encoding

This command allows you to define rule expressions to match the transfer encoding field present in HTTP/WSP(HTTP) headers.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* **> ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www transfer-encoding [ case-sensitive ]** *operator transfer_encoding*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

*operator* must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with
- **!starts-with**: Does not start with
- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

#### *transfer_encoding*

Specifies the WWW transfer encoding to match.

*transfer_encoding* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

**Usage Guidelines**

Use this command to define rule expressions to match the "transfer encoding" field present in HTTP/WSP(HTTP) headers.

#### Example

The following command defines a rule expression to match user traffic based on the WWW transfer encoding *user1*:

```
www transfer-encoding = user1
```

# www url

This command allows you to define rule expressions to match URL for any Web protocol analyzer—HTTP, WAP1.X, WAP2.0.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > ACS Configuration > Ruledef Configuration

**active-charging service** *service_name* > **ruledef** *ruledef_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-acs-ruledef)#
```

**Syntax Description**

**[ no ] www url [ case-sensitive ]** *operator url*

#### no

If previously configured, deletes the specified rule expression from the current ruledef.

#### case-sensitive

Specifies that the rule expression be case-sensitive. By default, rule expressions are not case-sensitive.

#### *operator*

Specifies how to match.

**operator** must be one of the following:

- **!=**: Does not equal
- **!contains**: Does not contain
- **!ends-with**: Does not end with

- **!starts-with**: Does not start with
- =: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **regex**: Regular expression
- **starts-with**: Starts with

***url***

Specifies the URL to match.

*url* must be an alphanumeric string of 1 through 127 characters and may contain punctuation characters.

## Usage Guidelines

Use this command to define rule expressions to match the URL for any Web protocol analyzer—HTTP, WAP1.X, WAP2.0.

The following table lists the special characters that you can use in regex rule expressions. For more information on regex support, refer to the *Enhanced Charging Service Administration Guide*.

***Table 8: Special Characters Supported in Regex Rule Expressions***

| Regex Character | Description |
|---|---|
| * | Zero or more characters |
| + | Zero or more repeated instances of the token preceding the + |
| ? | Match zero or one character<br>**Important** The CLI does not support configuring "?" directly, you must instead use "\077".<br>For example, if you want to match the string "xyz<any one character>pqr", you must configure it as:<br>**http host regex** *"xyz\077pqr"*<br>In another example, if you want to exactly match the string "url?resource=abc", you must configure it as:<br>**http uri regex** *"url\\077resource=abc"*<br>Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI. |
| \character | Escaped character |
| \? | Match the question mark (\<ctrl-v>?) character |
| \+ | Match the plus character |
| * | Match the asterisk character |

| Regex Character | Description |
|---|---|
| \a | Match the Alert (ASCII 7) character |
| \b | Match the Backspace (ASCII 8) character |
| \f | Match the Form-feed (ASCII 12) character |
| \n | Match the New line (ASCII 10) character |
| \r | Match the Carriage return (ASCII 13) character |
| \t | Match the Tab (ASCII 9) character |
| \v | Match the Vertical tab (ASCII 11) character |
| \0 | Match the Null (ASCII 0) character |
| \\ | Match the backslash character |
| Bracketed range [0-9] | Match any single character from the range |
| A leading ^ in a range | Do not match any in the range. All other characters represent themselves. |
| .\x## | Any ASCII character as specified in two-digit hex notation.<br>For example, \x5A yields a "Z". |
| \| | Specify OR regular expression operator<br>**Important** When using the regex operator "\|" in regex expressions, always wrap the string in double quotes.<br>For example, if you want to match the string "pqr" OR "xyz", you must configure it as:<br>**http host regex** *"pqr\|xyz"*. |

### Example

The following command defines a rule expression to match user traffic based on WWW URL *www.abc.com*:

```
www url = www.abc.com
```

The following command defines a regex rule expression to match either of the following values in the WWW URL string:
http://tp2.site.com/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/,
http://134.210.11.13/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/.

```
www url regex
"http://(tp2.site.com|134.210.11.3)/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/"
```