

Congestion Point Expressions

You can enter a congestion point expression by using the syntax listed in this topic. You can also embed Python scripting expressions within the congestion point expression.

If you embed Python expressions within a congestion point expression, use the escape sequence `< - then - >` to enclose the Python expression. See “Methods for Use with Scripting Expressions” on page 1 and “Match Criteria for Congestion Point Classification” on page 2.

The syntax for a congestion point expression is:

`< NetworkDevice > / < NetworkInterface > [/ < CongestionPoint >]`

- `< NetworkDevice >` —Network device listed in the directory.
- `< NetworkInterface >` —Network interface listed in the directory.

For information about interfaces, see Overview of Classification Scripts .

- `< CongestionPoint >` —(Optional) Name of an instance of a congestion point that is automatically created.

If one of the elements with the path contains a slash (/), use a backslash (\) as an escape character for the slash. For example, V.

Expressions in Templates for Congestion Point Profiles

You can create a congestion point profile to be used as a template for other profiles. Templates simplify management of congestion points. Rather than configuring each congestion point individually, you can create templates to define common parameters for a class of individual congestion points.

For example, in an environment in which VLAN interfaces GigabitEthernet1/0.1 through GigabitEthernet1/0.1000 have the same available bandwidth, you can specify the characteristics of the VLAN interface once and have SRC ACP create the congestion points based on the template configuration.

When a congestion point expression has the third element (`< CongestionPoint >`), SRC ACP uses the `< NetworkDevice > / < NetworkInterface >` part of the expression to load the congestion point from the directory, and uses it as a template to create a congestion point in memory for subscriber. The `< CongestionPoint >` part of the expression distinguishes each congestion point (available bandwidth) created from this template.

Methods for Use with Scripting Expressions

SRC ACP provides the following methods to use in scripting expressions:

- `slot(nasPortId)`—Collects the slot number from the `nasPortId` or `interfaceName`
Example—`slot(" atm 4/5:0.32")` == " 4"
- `port(nasPortId)`—Collects the port number from the `nasPortId` or `interfaceName`
Example—`port(" atm 4/5:0.32")` == " 5"
- `l2id(nasPortId)`—Collects the layer 2 ID from the `nasPortId` (VLAN id or ATM vpi.vci)
Example—`l2id(" atm 4/5:0.32")` == " 0.32"
- `escape(string)`—Replaces any slash with the escape sequence `\`
Example—`escape("atm 4/5")` == "atm 4\5"

Match Criteria for Congestion Point Classification

You can use the match criteria in Python scripting expressions for a congestion point expression. For more information about the match criteria, see Congestion Point Classification Criteria.

- Related Topics**
- Overview of Congestion Point Classification
 - Classifying Congestion Points
 - Defining a Congestion Point Profile
 - Configuration Statements for Congestion Point Classification

Published: 2009-09-15