

Technical Guide

Exinda Networks
www.exinda.com

Adaptive Response CLI Command Reference For Exinda Optimizer™

Optimizer version: 4.60
Document Release: February 2005

Command List

>CONF

config_script
config_rule
trigger
ar_rule

>conf

This is a reference guide for the conf command. The conf command is available via the Exinda Command Line Interface (CLI). To enter the conf mode, at the # prompt type conf:

DEFAULT# conf

Enter configuration information. Type 'exit' to submit.

CLI configuration manual available at <http://www.exinda.com/public/help>

conf>

Symbols

{ } = mandatory parameters

[] = optional parameters

Commands

bold = commands and reserved words

italics = user defined parameters

Pre-fixing certain commands with a “**no**” has the result of performing the inverse function of the command.

For example, the command “**network object** localServer 192.168.1.1 255.255.255.0 **monitor internal**” will establish the network object called “localServer” with an IP address of 192.168.1.1 and local monitoring. The command “**no network object** localServer” will remove this network object from the network objects list.

Abbreviations

CLI	Command Line Interface
WUI	Web User Interface

config_script

[no] config_script {*script-name*}

Example: no config_script myscript

Removes the file (*script-name*) of CLI conf commands.

Syntax Description

Reserved Words

no	Delete the script
-----------	-------------------

User Defined Parameters

<i>script-name</i>	Name of a file with user-defined CLI conf commands
--------------------	--

config_script {*script-name*}

{*config-statements*}

end_script

Example: config_script myscript
route 0.0.0.0 0.0.0.0 172.14.1.254
SMTP proxy 172.14.1.84
end_script

Save all the text (*config-statements*) between the **config_script** and **end_script** lines to *script-name*.

Syntax Description

Reserved Words

end_script	Defines the end of the script
-------------------	-------------------------------

User Defined Parameters

<i>script-name</i>	Name of a file with user-defined CLI conf commands
<i>config-statements</i>	CLI conf commands

config_rule

config_rule {*rule-name*} **script** [*script-name*]

Example: `config_rule use_route_b script myscript`

Updates the anywhere.rule table to specify the file of CLI conf commands (**script**) that will be called when the rule is activated. To specify that no script will be called, a *script-name* of " " can be used.

Syntax Description

Reserved Words

script	Defines that a script file is to be activated
---------------	---

User Defined Parameters

<i>rule-name</i>	Name of rule
<i>script-name</i>	Name of a file with user-defined CLI conf commands

config_rule {*rule-name*} **if** {*trigger-function*}[**&&** | **||**] [*trigger-function-additional*]

Example: `config_rule use_route_b if router_down && !heavy_load`

Updates the anywhere.rule table to specify the function that will trigger (*trigger-function*) the rule's activation (*rule-name*). The function is parsed to postfix notation and saved to the anywhere.rule_functions table.

Syntax Description

Reserved Words

if	Conditional statement for when a trigger function is activated
&&	Conditional statement to activate config_rule for when both <i>trigger function</i> and <i>trigger-function-additional</i> are triggered
 	Conditional statement to activate config_rule for when <i>trigger function</i> or <i>trigger-function-additional</i> is triggered

User Defined Parameters

<i>rule-name</i>	Name of rule
<i>trigger-function</i>	Parameter configured by user that activates the <i>rule_name</i> (see trigger)
<i>trigger-function-additional</i>	Additional parameter configured by user that activates the config_rule <i>rule_name</i> . It is included only when conditional statements }[&&] are used

[no] config_rule { *rule-name* } **alert** {*destination*}

Example: no config_rule use_route_b alert con@exinda.com

Updates the anywhere.rule_alert table to remove the specified alert.

Syntax Description

Reserved Words

no	Deletes the rule
-----------	------------------

User Defined Parameters

<i>rule-name</i>	Name of rule
------------------	--------------

<i>destination</i>	Email address to send alert
--------------------	-----------------------------

config_rule {*rule-name*} **alert** {*destination*} {"*subject*"} {"*message*"}

Example: `config_rule use_route_b alert
con@exinda.com,alert@exinda.com "Switching to config use_route_b"
"The router is down and the load isn't heavy"`

Updates the anywhere.rule_alert table to specify an alert that will occur when the rule is activated. Multiple alerts can be specified with a “,”.

Syntax Description

Reserved Words

alert	Definition that sends an alert to the user specified email address
,	Definition that separates multiple email addresses

User Defined Parameters

<i>rule-name</i>	Name of rule
<i>destination</i>	Email address to send alert
<i>subject</i>	Subject of alert specified by user that will be sent in the alert email
<i>message</i>	Message of alert specified by user that will be sent in the alert email

trigger

[no] trigger {*trigger-name*}

Example: `no trigger flooded_eth0`

Removes an entry from the anywhere.trigger table.

Syntax Description

Reserved Words

no	Delete the trigger
-----------	--------------------

User Defined Parameters

<i>trigger-name</i>	Name of trigger set by the user
---------------------	---------------------------------

trigger {*trigger-name*} **interface** {*interface*} {**inbound|outbound**} **exceeds** {*kilobits-per-second*}

Example: `trigger flooded_eth0 interface eth0 inbound exceeds 7500`

Update the anywhere.trigger table for an interface (Network Interface Card) on the Exinda appliance

Syntax Description

Reserved Words

interface	Defines that interface (NIC) is the trigger
inbound outbound	Defines if the traffic is inbound or outbound
exceeds	Defines that there is a limit on the amount of download of the interface

User Defined Parameters

<i>trigger-name</i>	Name of trigger
<i>interface</i>	Name of interface (NIC)
<i>kilobits-per-second</i>	Amount of exceeded traffic in kilobits-per-second

trigger {*trigger-name*} **policy** {*policy-name*} {**inbound|outbound**} **exceeds** {*“kilobits-per-second”*}

Example: `trigger heavy_http policy http inbound exceeds 7500`

Update the anywhere.trigger table for a policy defined by the user in the configuration of the Exinda appliance

Syntax Description

Reserved Words

policy	Defines that policy is the trigger
inbound outbound	Defines if the traffic is inbound or outbound
exceeds	Defines that there is a limit on the download of the interface

User Defined Parameters

<i>trigger-name</i>	Name of trigger
<i>policy-name</i>	Name of policy
<i>kilobits-per-second</i>	Amount of exceeded traffic in kilobits-per-second

trigger {*trigger-name*} **ip** {*subnet*} {**inbound|outbound**} {**day|week|month**} **exceeds** {*“megabytes”*}

Example: `trigger monthly_cap ip 172.14.1.0/24 inbound month exceeds 5000`

Update the anywhere.trigger table for an IP address or subnet defined by the user for the Exinda appliance

Syntax Description

Reserved Words

ip	Defines that subnet is the trigger
inbound outbound	Defines if the traffic is inbound or outbound
day week month	Defines the limit of the download daily, weekly or monthly
exceeds	Defines that there is a limit on the download of the subnet

User Defined Parameters

<i>trigger-name</i>	Name of trigger
<i>subnet</i>	Defined subnet
<i>kilobits-per-second</i>	Amount of exceeded traffic in kilobits-per-second

trigger {*trigger-name*} **sla** {*hostname*} **exceeds** {*millisecond-latency*}

Example: `trigger gateway_down sla 172.14.1.254 exceeds 30000`

Update the anywhere.trigger table for a Service Level Application defined by the user in the configuration of the Exinda appliance.

Syntax Description

Reserved Words

sla	Defines that Service Level Agreement is the trigger
exceeds	Defines that there is a limit on the download of the SLA

User Defined Parameters

<i>trigger-name</i>	Name of trigger
<i>hostname</i>	Defined hostname of the SLA
<i>millisecond-latency</i>	Latency of the SLA in milliseconds

trigger {*trigger-name*} **snmp** {*hostname*} {*oid*} **equals** {*value*}

Example: trigger link_up snmp 172.14.1.254 .1.3.6.2.1.2.2.1.8.1 equals 1

Update the anywhere.trigger table for a SNMP managed appliance defined by the user.

Syntax Description

Reserved Words

snmp	Defines that SNMP is the trigger
equals	Defines that the trigger is activated when the OID of the snmp managed appliance is equal to a value specified by the user

User Defined Parameters

<i>trigger-name</i>	Name of trigger
<i>hostname</i>	Defines the IP Address of the snmp managed appliance
<i>oid</i>	Object identifier of the snmp managed appliance
<i>value</i>	Value of the Object Identifier of the snmp managed appliance

trigger *{trigger-name}* **date** *{param}* **equals** *{value}*

Example: `trigger first_of_month date mday equals 1`

Adds an entry to the anywhere.trigger table.

Syntax Description

Reserved Words

date	Defines that date is the trigger
equals	Defines that the trigger is activated when the date is equal to a value specified by the user

User Defined Parameters

<i>trigger-name</i>	Name of trigger
<i>param</i>	Date parameter Valid date triggers: "year", "month", "mday", "wday" and "hour". For "wday", 0 = Sun, 1 = Mon, ..., 6 = Sat
<i>value</i>	Value of the specified date format

ar rule

Pattern: **[no] ar_rule** *{name}*

Example: `no ar_rule general`

Removes an entry/rule from the anywhere.trigger table.

Syntax Description

Reserved Words

no	Delete the ar_rule
-----------	--------------------

User Defined Parameters

<i>name</i>	Name of ar_rule
-------------	-----------------

ar_rule {name} if host in {subnet} exceeds {megabytes} per {day|week|month} add to {policy}

Example: ar_rule general if host in 210.50.6.0/24 exceeds 2000 per month add to 10.10.50

Adds an entry to the anywhere.trigger table. This rule adds host(s) to a policy specified by the user if the condition statements have been met.

Syntax Description

Reserved Words

if	Conditional statement for when ar_rule is activated
host	Defines that host activates the ar_rule
in	Defines the host in the specified subnet
exceeds	Defines that there is a limit on the download on each host specified
per	Defines the activation of the ar_rule every specified date interval
day week month	Intervals daily, weekly or monthly

add to	Defines that when the ar_rule is activated, add host to a particular policy
---------------	---

User Defined Parameters

<i>name</i>	Name of ar_rule
<i>subnet</i>	IP address and subnet of host
<i>megabytes</i>	Traffic exceeded in megabytes
<i>policy</i>	Policy number specified by the user in the configuration of the Exinda appliance

Example Script

Example Requirements:

Limit users' (172.14.1.0/25) download quota by implementing a policy 10.10.35. Restrict bandwidth of user to 64kbps once the user's quota of 50Mbytes of download has been exceeded. Flush users added to this policy daily at 1700hrs. Notify administrator if user 172.14.1.90/25 has exceeded inbound download of 50Mbytes.

Step 1:

- Create a trigger with name `users_a` to detect 172.14.1.90/25 when it exceeds an inbound download of 50 Mbytes daily.
- Create a trigger with name `flush_rules` that deletes the added hosts to policy 10.10.35 at 1700hrs daily

Trigger detecting exceeded download quota:

```
trigger {"trigger-name"} ip {"subnet"} {inbound|outbound} {day|week|month}
exceeds {"megabytes"}
```

Command line used

```
conf> trigger users_a ip 172.14.1.0/25 inbound day exceeds 50
```

Trigger which initiates a delete on the added hosts of policy 35 at 1700hrs:

```
trigger {trigger-name} date {param} equals {value}
```

Command line used

```
conf> trigger flush_rules date hour equals 17
```

Step 2:

- Create a `config_rule` named `users_exceed` that sends an alert if that quota has been exceeded.
- Delete the hosts added to policy 10.10.35 at 1700hrs with config rule `restrict_s`

If quota exceeded:

```
config_rule {rule-name} if {"trigger-function"}
```

Command line used:

```
conf> config_rule users_exceed if users_a
```

Send an alert with a subject and message:

```
config_rule {rule-name} alert {destination} {"subject"} {"message"}
```

Command line used:

```
conf> config_rule users_exceed alert alertmail@exinda.com  
"Alert Quota: Exceeded" "172.14.1.90/25"
```

Flush policy 35 if it is 1700hrs

```
config_rule {rule-name} if {"trigger-function"}
```

Command line used

```
conf> config_rule restrict_s if flush_rules
```

Config rule that runs restrict script:

```
config_rule {rule-name} script [script-name]
```

Command line used

```
conf> config_rule restrict_s script restrict_script
```

Step 3:

- Create an ar_rule that adds hosts to a policy when they exceed a download quota of 50 Mbytes
- Create a config_script with name restrict_script that flushes the users in policy 10.10.35

If quota exceeded add user to policy:

```
ar_rule {name} if host in {subnet} exceeds {megabytes} per {day|week|month} add to  
{policy}
```

Command line used:

```
conf> ar_rule general if host in 172.14.1.0/25 exceeds 50 per  
day add to 10.10.35
```

Run restrict script that adds a policy to restrict 172.14.1.0/25 bandwidth:

```
config_script {"script-name"}  
  
    {"config-statements"}  
end_script
```

Command line used (Refer to CLI conf manual under optimize command for different features):

```
conf> config_script restrict_script
conf> optimizer
conf> policy 35 restrict_bandwidth
conf> schedule 0
conf> minbw 64
conf> maxbw 64
conf> priority 1
conf> action OPTIMIZE
conf> end_script
```

Summary of Commands

Command	Description
conf> ar_rule general if host in 172.14.1.0/25 exceeds 50 per day add to 10.10.35	ar_rule that add users to policy 10.10.35 if user exceeds 50Mbytes per day
conf> trigger users_a ip 172.14.1.90/25 inbound day exceeds 50	trigger that activates when 172.14.1.90/25 exceeds 50Mbytes of inbound traffic
trigger flush_rules date hour equals 17	trigger that activates at 1700hrs
conf> config_rule users_exceed if users_a conf> config_rule users_exceed alert alertmail@exinda.com "Alert Quota: Exceeded" "172.14.1.0/25"	configure rule users_exceed that sends an alert if 172.14.1.90/25 exceeds 50 Mbytes per day
conf> config_rule restrict_s if flush_rules	Activates config_rule restrict_s if it is 1700hrs
conf> config_rule restrict_s script restrict_script	config_rule restrict_s that runs config_script restrict_script
conf> config_script restrict_script conf> optimizer conf> policy 35 restrict_bandwidth conf> schedule 0 conf> minbw 64 conf> maxbw 64 conf> priority 1 conf> action OPTIMIZE conf> end_script	start of result_script Set policy 10.10.35 restrict_bandwidth Scheduled "Always" Min and Max bandwidth of 64kbps Top priority and optimize bandwidth end of script