Replacing Configuration on a Working Router

By Ivan Pepelnjak

1. 3. 2007

Have you ever faced a situation where you have badly misconfigured your router and had to roll back the configuration to a previous known state? Assuming that the working configuration was still saved in the NVRAM (unless you were trigger-happy and saved the new configuration as soon as you changed it), you had two options:

- Manually working out the configuration commands to bring the router back to the previous state (which might be a problem if your phone is constantly ringing due to a network down situation).
- Reloading the router and thus extending the downtime.

Cisco has addressed this problem with the Configuration Replace and Rollback introduced in IOS release 12.3(7)T and integrated in IOS release 12.4.

Note: This feature relies on Contextual Configuration Diff and Configuration Archive features described in previous IP Corner articles Router

Configuration Management … Too Good to be True? and Keep Track of Router Configurations with Configuration Archive.

Configuration Rollback Basics

The Configuration Rollback feature is conceptually very simple: whenever you mess up the router's configuration (for example, by changing an interface's IP address like I did in Listing 1), you can simply use the **configure replace** command to replace the current configuration with a previously saved one. I've used **nvram:startup-config** as the previously saved configuration, effectively bringing the current router configuration in synchronization with the startup configuration (Listing 2).

*Note: The **list** option of the **configure replace** command lists all commands that will be applied to the router's configuration. I would strongly recommend using it to track changes that Cisco IOS made to the current configuration.*

Listing 1: Changing interface IP addresses

```
fw#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
fw(config)#interface loopback 0
fw(config-if)#ip address 10.1.1.1 255.255.255.255
fw(config-if)#exit
fw(config)#interface FastEthernet 0/0
fw(config-if)#shutdown
fw(config-if)#ip address 1.2.3.4 255.255.0.0
fw(config-if)#^Z
```

Listing 2: Rollback to the startup configuration

```
fw#configure replace nvram:startup-config list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
```

```
!Pass 1
!List of Commands:
no interface Loopback0
interface FastEthernet0/0
 no ip address 1.2.3.4 255.255.0.0
 no shutdown
interface FastEthernet0/0
 ip address 10.0.0.1 255.255.255.0
end
Total number of passes: 1
Rollback Done
```

Most commonly, the configuration rollback would be used to replace the current configuration with the startup one. You can, however, use any file transfer method supported by Cisco IOS to fetch the target configuration. For example, if you have configured the Configuration Archive feature, you can use the **show archive** command to identify the URL of a previously saved router configuration and use that URL in the **configure replace** command (Listing 3).

Listing 3: Rollback to an archived configuration

```
fw#show archive
The next archive file will be named tftp://10.0.0.2/fw.cfg-67
 Archive #  Name
   0        tftp://10.0.0.2/fw.cfg-60
   1        tftp://10.0.0.2/fw.cfg-61
   2        tftp://10.0.0.2/fw.cfg-62
   3        tftp://10.0.0.2/fw.cfg-63
   4        tftp://10.0.0.2/fw.cfg-64
   5        tftp://10.0.0.2/fw.cfg-65
   6        tftp://10.0.0.2/fw.cfg-66 <- Most Recent
   7        tftp://10.0.0.2/fw.cfg-52
   8        tftp://10.0.0.2/fw.cfg-53
   9        tftp://10.0.0.2/fw.cfg-54
   10        tftp://10.0.0.2/fw.cfg-55
```

```
   11       tftp://10.0.0.2/fw.cfg-56
   12       tftp://10.0.0.2/fw.cfg-57
   13       tftp://10.0.0.2/fw.cfg-58
   14       tftp://10.0.0.2/fw.cfg-59
fw#configure replace tftp://10.0.0.2/fw.cfg-66 list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
Loading fw.cfg-66 from 10.0.0.2 (via FastEthernet0/0): !
[OK - 2150 bytes]
… rest deleted …
```

In-depth Look at Configuration Rollback

The Configuration Rollback feature relies on the Contextual Configuration Diff feature to identify changes between the running configuration and the target one. As Cisco IOS is not perfect in identifying differences between two router configurations, the Configuration Rollback has built-in safety measures:

Configuration rollback is attempted in multiple passes, in each pass identifying the differences and thus gradually bringing the configuration toward the target one.

Note: As the router has to generate running configuration in each pass, the configuration rollback might take a long time on a heavily loaded router with large configuration (for example, Cisco 7600 with thousands of interfaces).

The rollback process is stopped after the fifth pass, potentially leaving the router configuration in a partially recovered state.

For example, as described in a previous IP corner article, Contextual Configuration Diff is not very good in generating commands needed to change the order of lines in an IP access list. The Configuration Rollback

process handles this situation (as well as most other order-dependent configuration constructs) very gracefully – when faced with a changed access list (Table 1), it deletes the access list in the first pass and recreates it in the second pass.

Table 1: Access list change

Startup configuration	Running configuration
ip access-list extended Test	ip access-list extended Test
permit tcp any host 10.0.0.2	deny tcp any any eq smtp
deny tcp any any eq smtp	permit tcp any host 10.0.0.2
permit tcp any any eq www	permit tcp any any eq www
permit tcp any any established	permit tcp any any established
deny ip any any log	deny ip any any log

Listing 4: Access list rollback

```
fw#configure replace nvram:startup-config list force
!Pass 1
!List of Commands:
no ip access-list extended Test
end
!Pass 2
!List of Commands:
ip access-list extended Test
 permit tcp any host 10.0.0.2
 deny tcp any any eq smtp
 permit tcp any any eq www
 permit tcp any any established
 deny ip any any log
end
Total number of passes: 2
Rollback Done
```

*Note: The **force** option of the **configure replace** command eliminates all user prompting.*

Rollback Failures

Since the Contextual Configuration Diff feature is not aware that the **policy-map** commands are order-dependent, one would expect the configuration rollback to have a problem with the **class** statement reordering in a **policy-map**. Indeed, the reordering is not identified and thus the configuration rollback is incomplete. Consider, for example, the configuration snippets in Table 2: as the classes have been reordered, all web traffic toward the server will be treated the same as the Internet web surfing. However, the configuration rollback does not identify any changes and does not fix the **policy-map** configuration (Listing 5).

Table 2: Policy map change

Startup configuration	Running configuration
Policy-map Shaping	policy-map Shaping
class TrafficToServer	class WWW
set precedence 3	set precedence 0
class WWW	police rate percent 50
set precedence 0	conform-action transmit
police rate percent 50	exceed-action drop
conform-action transmit	class TrafficToServer
exceed-action drop	set precedence 3

Listing 5: Policy map rollback

```
fw#configure replace nvram:startup-config list force
Total number of passes: 0
Rollback Done
```

When faced with IOS configuration that cannot be changed (for example, the IP Service Level Agreement – SLA – configuration in Table 3), configuration rollback fares a lot better. It aborts the rollback process after five passes and gives you the list of commands it has attempted to execute in the last pass (Listing 6). You can then use the command list to fix the remaining configuration changes manually.

Table 3: IP SLA change

Startup configuration	Running configuration
ip sla 1	ip sla 1
icmp-echo 1.2.3.4	icmp-echo 2.3.4.5
timeout 3	timeout 6
ip sla schedule 1 life forever start-time now	ip sla schedule 1 life forever start-time now

Listing 6: Configuration rollback aborts when unable to change IP SLA configuration in five passes

```
fw#configure replace nvram:startup-config list force
Entry already running and cannot be modified
        (only can delete (no) and start over)
        (check to see if the probe has finished exiting)
… text deleted …
!Pass 1
!List of Commands:
ip sla 1
 no icmp-echo 2.3.4.5
 no timeout 6
ip sla 1
 icmp-echo 1.2.3.4
 timeout 3
end
… text deleted …
```

```
The rollback configlet from the last pass is listed below:
********
!List of Commands:
ip sla 1
 no icmp-echo 2.3.4.5
 no timeout 6
ip sla 1
 icmp-echo 1.2.3.4
 timeout 3
end
********
Rollback aborted after 5 passes
```

Configuration Replace Feature

The Configuration Replace feature is an interesting application of the Configuration Rollback. Assume that you have a tool that builds router configurations offline (for example, based on a central database). Whenever you download a generated configuration file into your router, you risk that the current configuration will be broken and you'll lose access to the router, making it impossible to fix the error. The Configuration Replace feature automates this process by adding the **time** option to the **configuration replace** command:

- Using the **configuration replace** *url* **time** *seconds* command, you initiate the configuration replacement process.
- The router generates an archive copy of the running configuration (this copy will be used during the rollback process if needed).

Note: The Configuration Archive must be configured for the Configuration Replace feature to work correctly.

The target configuration is downloaded and replaces the running configuration (using the Configuration Rollback feature).

You have to confirm the successful completion of the configuration replacement process with the **configure confirm** command within the

timeframe specified with the **time** option of the **configuration replace** command.

If the configuration replacement is not confirmed (for example, you've lost access to the router), Cisco IOS performs an automatic rollback to the archived configuration generated in Step 2 of this process.

The whole process (including the rollback step) is displayed in Listing 7.

Listing 7: Configuration replacement and rollback

```
fw#configure replace tftp://10.0.0.2/fw-confg force time 30
!!Timed Rollback: Backing up to tftp://10.0.0.2/fw.cfg-3
Loading fw-confg from 10.0.0.2 (via FastEthernet0/0): !
[OK - 1624 bytes]
% Class-map is being used
% Class-map is being used
Feb  1 12:33:15: Rollback:Acquired Configuration lock.
Total number of passes: 2
Rollback Done
fw#
Feb  1 12:33:19: %PARSER-3-
CONFIGNOTLOCKED: Unlock requested by process '183'. Configuration not locked
.
… after 30 seconds …
Timed Rollback: rolling to:tftp://10.0.0.2/fw.cfg-3
Loading fw.cfg-3 from 10.0.0.2 (via FastEthernet0/0): !
[OK - 2065 bytes]
Feb  1 12:33:49: Rollback:Acquired Configuration lock.
```

Event-Driven Rollback

While the *Configuration Replace and Rollback* feature addresses the needs of large networks using centralized router configuration management systems, it does not give us what we need in our daily network configuration jobs – the ability to have an automatic rollback to a known configuration if we lose access to the router during the configuration process (an alternative, of course, is to deploy CRS routers, as IOS XR

supports configuration commit and rollback). Yet again, we have to rely on Embedded Event Manager (EEM) to help us.

Note: The EEM policies in this section could be written in Tcl programming language. I've decided, however, to implement them in EEM applets to give you a few examples on what you can achieve without committing yourself to the complexities of Tcl.

We'll start by configuring two simple EEM applets that detect configuration changes. One of them is triggered by the **configure terminal** command, the other by the "%SYS-5-CONFIG_I: Configured from ..." *syslog* message (Listing 8). Both applets call a common EEM policy (*ScheduleRollback*) that schedules an automatic rollback.

*Note: All EEM applets in this section contain debugging messages produced by **action 99.*** configuration commands. You might want to remove them before deploying this solution in a production environment.*

Listing 8: EEM applets detecting configuration change

```
event manager applet DetectManualConfig
 event cli pattern "configure terminal" sync no skip no occurs 1
 action 1.0 policy ScheduleRollback
 action 99.99 syslog priority debugging msg "Detected conf term"
!
event manager applet DetectConfigurationChange
 event syslog pattern "CONFIG_I"
 action 1.0 policy ScheduleRollback
 action 99.99 syslog msg priority debugging "Detected config change"
```

Automatic configuration rollback is triggered with help of a periodically decrementing counter:

- The *ScheduleRollback* applet sets the *RollbackCounter* to a non-zero value (number of minutes until the automatic rollback plus one).
- The RollbackCountdown applet decrements the *RollbackCounter* once a minute.
- The *TriggerRollback* applet is triggered whenever the *RollbackCounter* reaches value 1 (and is re-enabled after the counter is reset to a value higher than two).

Note: See also the Advanced deployment scenarios section of the IP Corner article Keep Track of Router Configurations with Configuration Archive for a detailed description of this technique.

The *TriggerRollback* applet should invoke the automatic rollback only if requested by the operator. To make the rollback conditional, we use yet another counter (*DoRollback*) and the *TriggerRollback* applet simply decrements it indicating the need for automatic rollback (Listing 9)

Listing 9: Scheduling automatic rollback

```
event manager applet ScheduleRollback
 event none
 action 1.0 counter name RollbackCounter op set value 3
 action 99.99 syslog priority debugging msg "Rollback scheduled in
$_counter_value_remain minutes"
!
event manager applet RollbackCountdown
 event timer cron name RollbackCountdown cron-entry "* * * * *"
 action 1.0 counter name RollbackCounter op dec value 1
 action 99.98 counter name RollbackCounter op nop
 action 99.99 syslog priority debugging msg "Rollback Countdown ...
$_counter_value_remain"
!
event manager applet TriggerRollback
 event counter name RollbackCounter entry-val 1 entry-op eq exit-val 2 exit-
op gt
 action 1.0 counter name DoRollback op dec value 1
 action 99.99 priority debugging syslog msg "Triggering rollback operation"
```

The last part of the solution is the “user interface”:

- *BeginConfigTransaction* applet enables the automatic rollback setting of the *DoRollback* counter to a non-zero value.
- *CommitConfigTransaction* commits the configuration changes and disables the automatic rollback (setting the *DoRollback* counter to zero).
- *RollbackConfigTransaction* is triggered by the *DoRollback* counter and performs the actual rollback.

The *BeginConfigTransaction* and *RollbackConfigTransaction* applets need a working copy of the router configuration. In the simplest implementation, they use **startup-config** (NVRAM), but you could also store the running configuration into a flash file. All three applets are included in Listing 10.

Listing 10: Automatic rollback user interface

```
event manager applet BeginConfigTransaction
 event none
 action 1.0 syslog msg "Start the write memory command"
 action 1.1 cli command "write memory"
 action 1.2 syslog msg "Configuration transaction started"
 action 2.0 counter name DoRollback op set value 2
!
event manager applet CommitConfigTransaction
 event none
 action 1.0 counter name DoRollback op set value 0
 action 2.0 syslog msg "Configuration transaction committed"
!
event manager applet RollbackConfigTransaction
 event counter name DoRollback entry-val 1 entry-op eq exit-val 1 exit-op gt
 action 1.0 cli command "Configuration rollback triggered"
 action 1.1 cli command "configure replace nvram:startup-config force"
 action 1.2 syslog msg "Configuration rolled back to startup-config"
```

To add icing on the cake, you can define two command aliases (*start* and *commit* in my example) to make it easier for the operators to use the new functionality (Listing 11).

Listing 11: Exec-mode aliases

```
alias exec start event manager run BeginConfigTransaction
alias exec commit event manager run CommitConfigTransaction
```

Summary

The *Configuration Replace and Rollback* functionality is a tool that will help you in emergency situations when you need to change the router configuration to a previous known state without reloading the device. As the feature relies on *Contextual Configuration Diff* feature, it's not absolutely bulletproof; while it can handle most order-dependent IOS configuration constructs (like **access-lists** or **community-lists**), it fails when faced with a reordered **policy-map** statement.

This IOS feature also contains provisions for large networks that prepare the router configurations with centralized network management tools as it allows you to replace the current router configuration with a new one and perform a rollback if the new configuration is not confirmed within the specified timeframe. It does not, however, solve the problem we're commonly facing: how do you recover a router when your configuration change has cut you off from it. Fortunately, IOS already includes simple scripting tool (Embedded Event Manager applets) that you can use to implement the automatic rollback.

NIL – More Than Just a Training Company

NIL Learning delivers the leading-edge Cisco training to IT professionals and companies around the globe. Through field-proven experts — each both active engineer and instructor — NIL Learning enhances the standard learning curriculum with real-life experience and helps clients to maximize their training investment.

NIL Learning is part of NIL, a leading global IT solutions provider. Since 1992, NIL has been at the forefront of advanced contributors to strategic partner Cisco's technologies, learning curriculum and value-added solutions deployed to clients around the globe. Today, NIL has earned the highest certifications offered by Cisco, VMware, EMC, HP, IBM, Microsoft, F5, Jive,

MobileIron, RSA, VCE and others. Their portfolio of solutions consists of managed services, professional services and learning services.

NIL is headquartered in Slovenia, with regional offices in Croatia, Serbia, Saudi Arabia, the U.S., Turkey, South Africa, Morocco, Nigeria, Kenya and Botswana.

Why learn at NIL LEARNING?

- All NIL LEARNING instructors are **field-proven experts** - each both active engineer and instructor.
- **75% of NIL LEARNING engineers hold CCSI** certifica- tions, and **18 have already achieved the respected CCIE rank.**
- NIL LEARNING **enhances the standard learning curriculum** with real-life experience and helps clients to maximize their training investment.
- NIL has been a Cisco Training Partner for many years; it became a **Cisco Learning Partner in 1993**, and has been a **Cisco Gold Partner since 1995.**
- NIL was awarded the **Cisco Most Business Relevant Learning Partner in MEA in 2010 and the most innova- tive learning partner in MEA.**
- NIL received the **Innovation Award for its Technology Led Training and its extensive contribution to Cisco learning solutions** at the Cisco EMEAR Learning Partner Summit in 2012.
- NIL received the **Innovation Award for its Technology Led Training and Advanced Engineer Program** at the Cisco Global Learning Partner Summit in 2013.
- NIL LEARNING runs a **centralized training schedule across the whole EMEAR region.**

More Info

Slovenia

T: +386 1 4746 500

E: sales-support@nil.com

Botswana

T: +267 318 1684

E: training@it-iq.bw

Croatia

T: +385 (0)51 583 255

E: info-nilcroatia@nil.com

Kenya

T: +27 (0)11 575 4637

E: mea_sales@nil.com

Morocco

T: +212(0) 660 808 394

E: info-nilmorocco@nil.com

Nigeria

T: +27 (0)11 575 4637

E: mea_sales@nil.com

Saudi Arabia

T: +966 1 465 4641

E: info.nilme@nil.com

Serbia

T: +381 11 2282 818

E: info-nilserbia@nil.co.rs

South Africa

T: +27 (0)11 575 4637

E: mea_sales@nil.com

Turkey

T: +902 123 81 8639

E: info-nilturkey@nil.com

USA

T: +1 612 886 3900

E: info-nilusa@nil.com

www.learning.nil.com