

Scripting on Routers

On-box Automation with JUNOS



Phil Shafer

<phil@juniper.net>

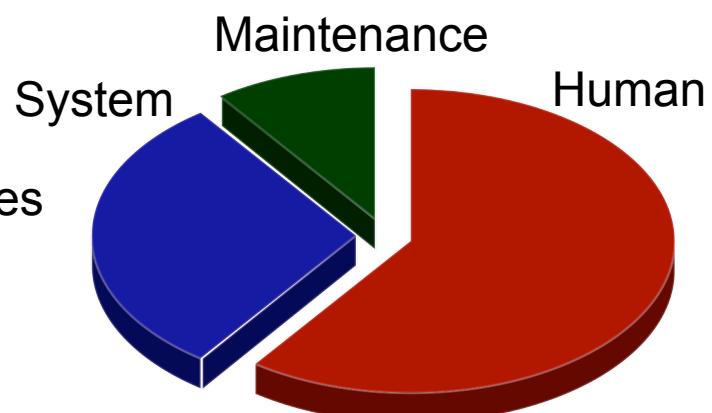
NANOG 47, 10/19/2009



Problem Statement

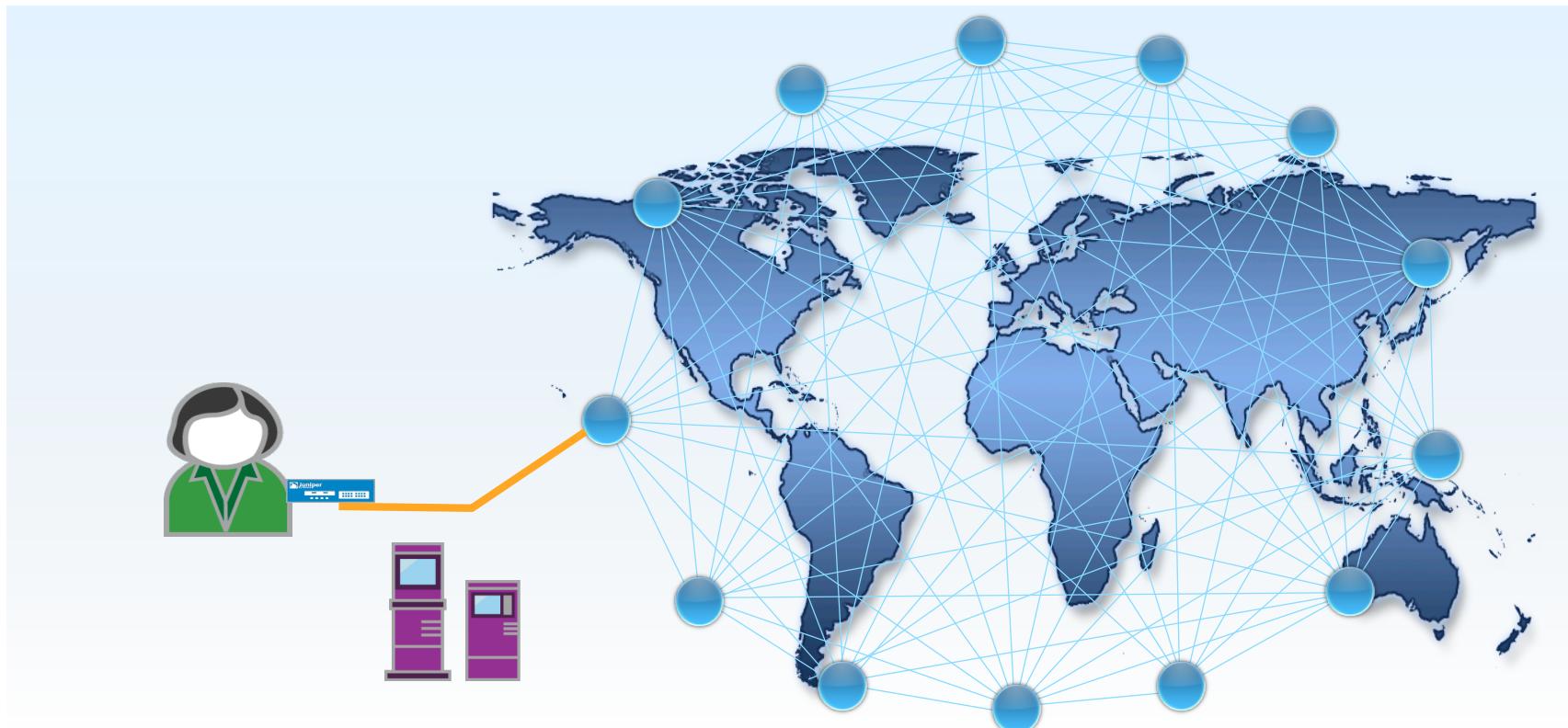
- Configuration is increasingly complex
 - Impedes deployment of complex services
- Network operators have custom design rules
 - Vary among our customers
- No enforcement mechanism
- Cost of failure is high and getting higher
 - Customer expectations, SLAs, convergence

Config errors are 50-70% of outages



Off-box Network Management

- "Database of Record" is a real database
 - Generate configs and push to devices



But.....



Goals for Scripting

- Tighter integration
- Promote high-level concepts and views
- Enforce policies on the box
- Simplify configuration
- Give better insight into what's going on
- Help avoid common and repeated mistakes

Did you ever...

- Make a mistake?
 - Forget something?
 - Accidentally deleted something?
 - Something important? Something obvious?
 - Been surprised by what you configured?

- Make a plan, a design, a template, but...
 - Someone didn't follow it?
 - Someone didn't understand it?
 - Someone didn't like it?

- Spend hours trouble shooting a problem...
 - And it was something obvious?
 - That you've seen before?
 - And it's 3am? Again?

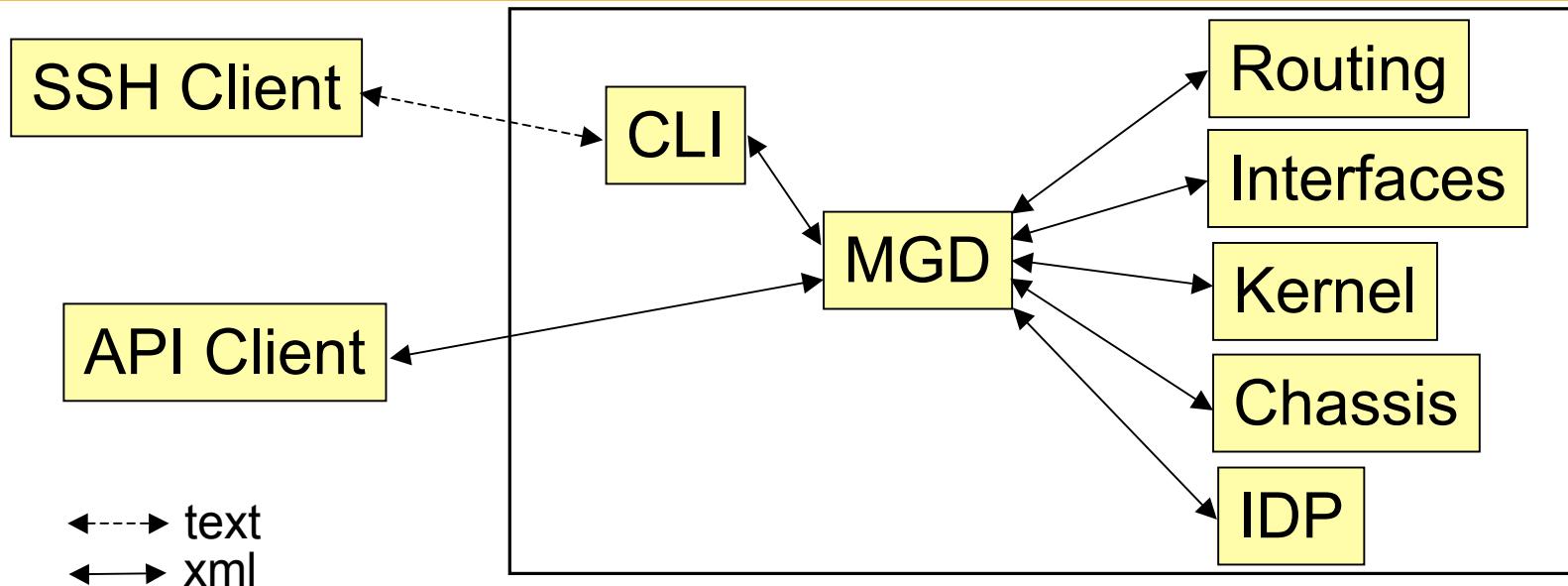
XML API

- JUNOScript XML API (2001)
 - Standard is RFC 4741 (NETCONF)
- XML-based
 - Send XML, get XML
 - Extensible
 - An RPC for every CLI command

```
cli@test> show route 10.4.9.1 protocol bgp best next-hop 10.5.14.2

<rpc>
  <get-route-information>
    <destination>10.4.9.1</destination>
    <protocol>bgp</protocol>
    <best/>
    <next-hop>10.5.14.2</next-hop>
  </get-route-information>
</rpc>
```

XML Traffic



- CLI renders XML into text
- Internally data is trafficked in XML
- API publishes that data to client apps

CLI Output and XML Replies

```
inet.0: 37 destinations, 46 routes (36 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.4.9.0/24      *[BGP/170] 1w2d 10:15:09, MED 3, localpref 100
                  AS path: 68 I
                  > to 10.5.14.2 via fe-0/3/0.0
```

```
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/9.5I0/junos">
  <route-information xmlns="http://xml.juniper.net/junos/9.5I0/junos-routing">
    <route-table>
      <table-name>inet.0</table-name>
      <destination-count>37</destination-count>
      <total-route-count>46</total-route-count>
      <active-route-count>36</active-route-count>
      <holddown-route-count>0</holddown-route-count>
      <hidden-route-count>1</hidden-route-count>
      <rt junos:style="brief">
        <rt-destination>10.4.9.0/24</rt-destination>
        <rt-entry>
          <active-tag>*</active-tag>
          <protocol-name>BGP</protocol-name>
          <preference>170</preference> ...
```

Scripting technologies

Commit
Scripts



Automatic part of commit process

Op
Scripts



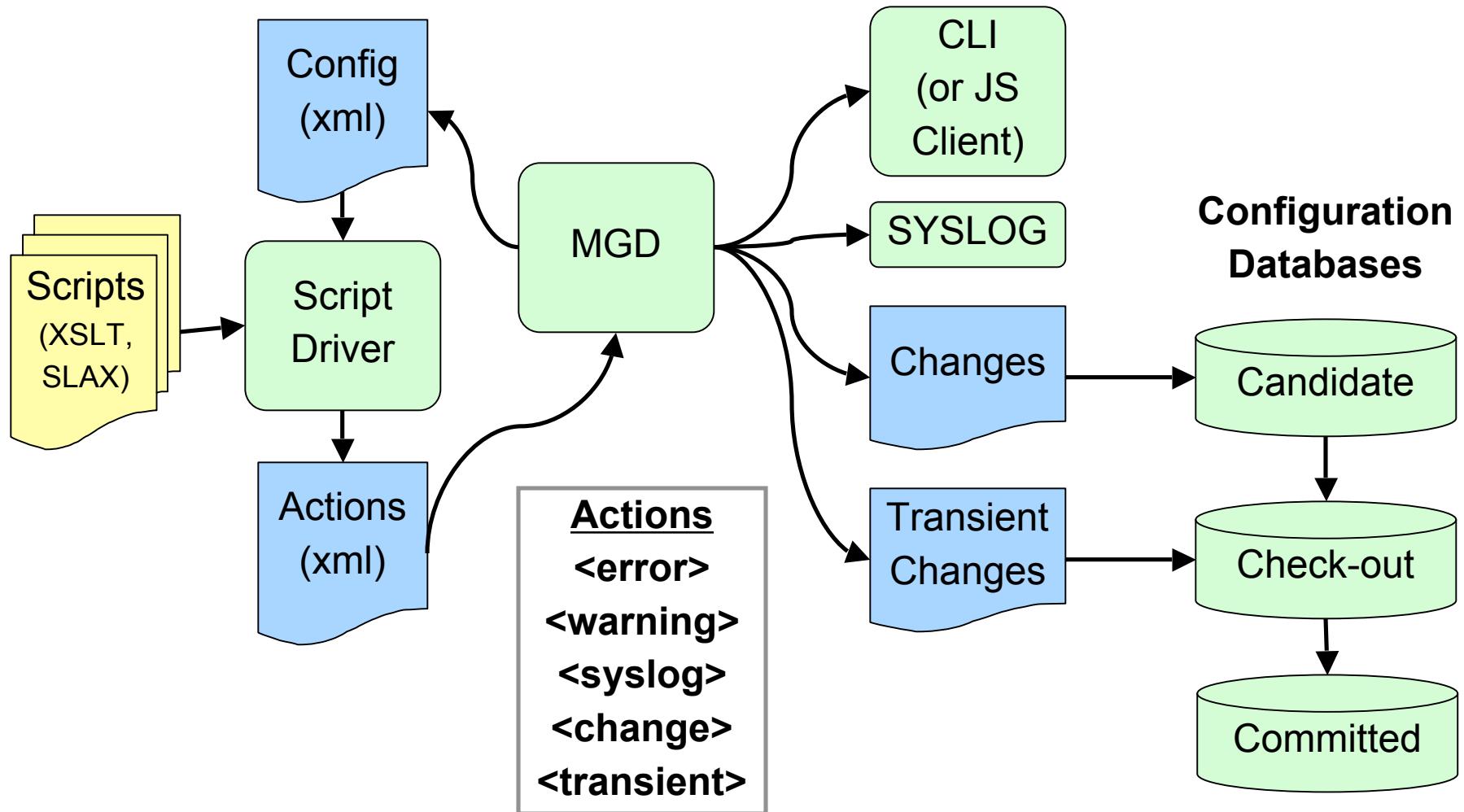
Add new commands to the CLI

Event
Scripts



Triggered when events occur

Commit Scripts Internals



Example Constraints

- Insist that all LDP-enabled interfaces are configured for an IGP
- Insist that each ATM interface not have more than 1000 PVCs configured
- Insist that an IGP does not use an import policy that will import the full routing table
- Insist that the "re0" and "re1" configuration groups are set up correctly and the nothing in the foreground configuration is blocking their proper inheritance

```
user@cli# commit
[edit protocols ospf area 10.4.0.0]
  'interface ge-3/2/1.0;'
    error: ldp not enabled for this interface
commit failed
```

apply-macro Statement

- Can appear anywhere in the configuration
- Takes a name as argument
- Contains optional set of name/value pairs
- Define any content you want/need
 - Customer name/number/class
- Used as input to commit script

```
[edit protocols ospf area 10.4.0.0]
apply-macro no-1dp;
```

```
apply-macro name1 {
    name2 value2;
    name3 value3;
    name4;
}
```

Change the Configuration

- Examples:
 - Automatically build a [protocols rip group] containing every t1 interface configured under [interfaces]
 - Automatically configure "family iso" on any interface with "family mpls"
 - Apply a configuration group for any sonet interface whose descriptions string matches a particular regular expression
 - Change "insist" on the previous slide to "automatically configure"
- Fix problems before they happen

```
<change>
  <protocols>
    <t1dp>
      <interface>
        <name> ge-3/2/1.0 </name>
      </interface>
    </t1dp> ...
```

Transient Changes

- Config that is published to software components, but not seen by the user
 - To see transients, use "show | display commit-scripts"
 - Automatically generate configuration
 - Remove the possibility of mistakes
 - Make the rule implicit and the enforcement assured

```
<transient-change>
  <protocols>
    <ldp>
      <interface>
        <name> ge-3/2/1.0 </name>
      </interface>
    </ldp> ...
```

Example: ex-iso.slax

- Enforces constraints on ISO and MPLS
- Rules the configuration must follow:
 - If interface has [family iso] configured but not [family mpls]
 - Then add family mpls
 - If interface not configured under [protocols mpls]
 - Then add the interface there
- Use the "jcs:emit-change" template (function)
- Both changes are accompanied with appropriate warning messages

```
interfaces {  
    so-0/0/0 {  
        unit 0 {  
            family iso;  
            ...  
        }  
        protocols {  
            mpls {  
                ...  
            }  
        }  
    }  
}
```

Before

```
interfaces {  
    so-0/0/0 {  
        unit 0 {  
            family iso;  
            family mpls;  
            ...  
        }  
        protocols {  
            mpls {  
                interface so-0/0/0.0;  
                ...  
            }  
        }  
    }  
}
```

After

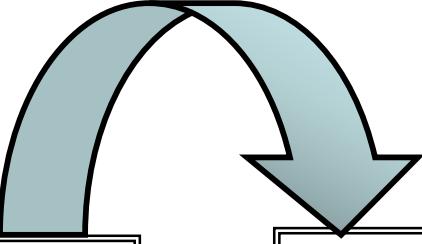
```
[edit]  
user@cli# commit  
[edit interfaces interface so-0/0/0 unit 0]  
  warning: Adding 'family mpls' to iso-enabled interface  
[edit interfaces interface so-0/0/0 unit 0]  
  warning: Adding iso-enabled interface so-0/0/0.0 to [protocols mpls]  
commit complete
```

How does it work?

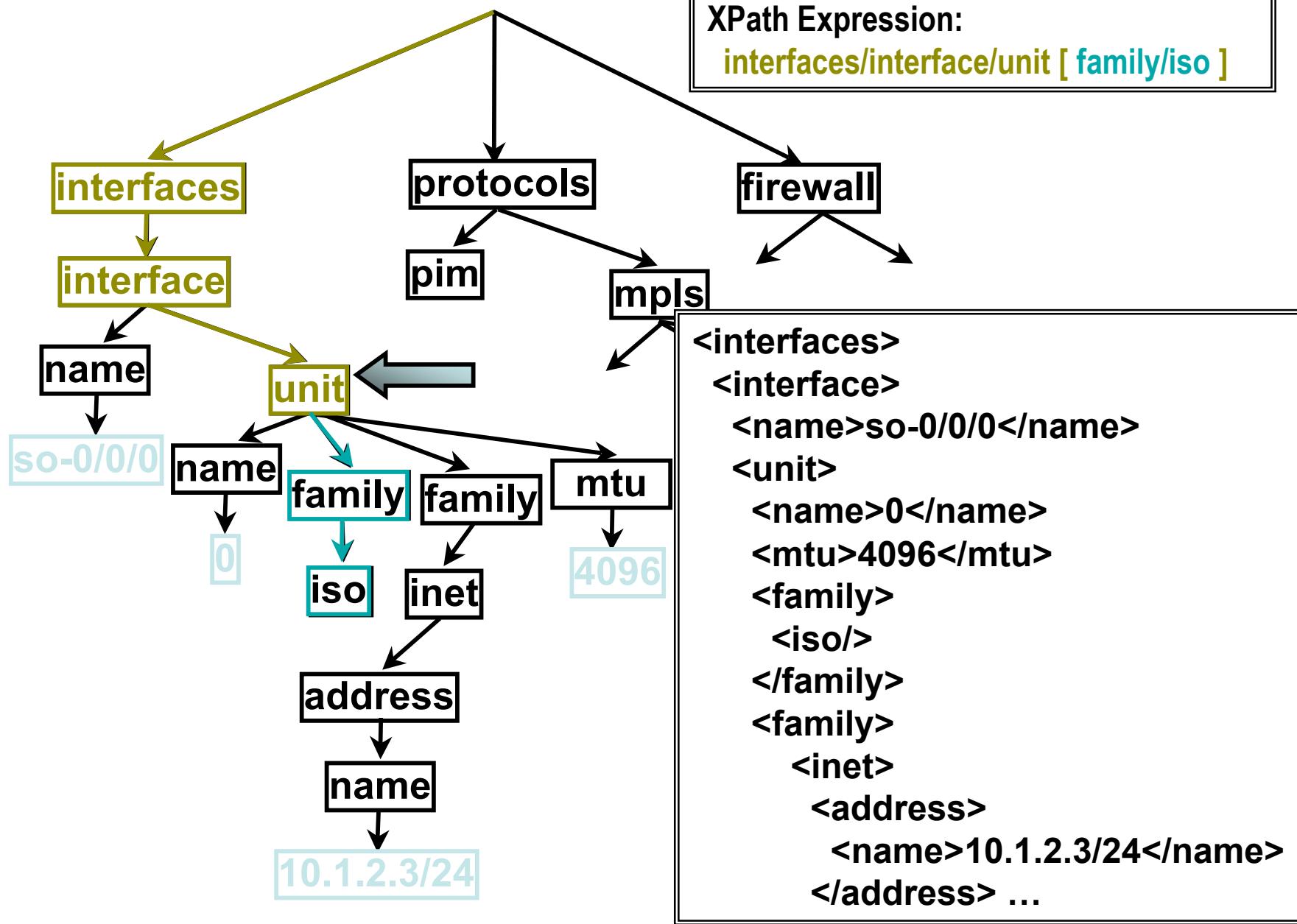
- Inspects configuration data in XML
 - Use XPath expressions to local interesting nodes
- Call jcs:emit-change
 - Pass in required arguments
 - \$content == the change to make (in xml)
 - \$dot == where to make the change
 - \$message == message to display to user
 - \$tag == 'change' or 'transient-change'
 - Emits warning message and complete change element
- MGD inspects the script's output
 - Looks for errors
 - Loads any changes
- Normal commit logic proceeds

Expressing Data in XML

```
interfaces {  
    so-0/0/0 {  
        unit 0 {  
            mtu 4096;  
            family iso;  
            family inet {  
                address 10.1.2.3/24;  
            } } } }  
protocols {  
    pim;  
    mpls { ... }  
}
```



```
<interfaces>  
    <interface>  
        <name>so-0/0/0</name>  
        <unit>  
            <name>0</name>  
            <mtu>4096</mtu>  
            <family>  
                <iso/>  
            </family>  
            <family>  
                <inet>  
                    <address>  
                        <name>10.1.2.3/24</name>  
                    </address> ...  
                </inet>  
            </family>  
        </unit>  
    </interface>  
</interfaces>
```



```
match configuration {
    var $mpls = protocols/mpls;

    for-each (interfaces/interface/unit[family/iso]) {
        var $ifname = ../name . name;

        if (not(family/mpls)) {
            var $message = "Adding 'family mpls' to iso-enabled interface";
            var $content = <family> {
                <mpls>;
            }
            call jcs:emit-change($message, $content);
        }
        if ($mpls && not($mpls/interface[name == $ifname])) {
            var $message = "Adding iso-enabled interface "
                _ $ifname _ " to [protocols mpls]";
            var $content = <interface> {
                <name> $ifname;
            }
            call jcs:emit-change($dot = $mpls, $message, $content);
        }
    }
}
```

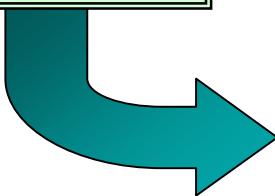
Macro Expansion

- Example: Use a basic interface class to drive configuration of protocols (IGP, CoS) and firewalls

```
apply-macro so-1/0/1.0 {  
    address 10.0.0.1/24;  
    description "Link to Hooverville";  
    igp ospf;  
    area 10.1.2.3;  
    class platinum;  
}
```

```
interfaces {  
    so-1/0/1 {  
        description "Link to Hooverville";  
        unit 0 {  
            family {  
                inet {  
                    address 10.0.0.1/24;  
                    ...  
                }  
            }  
        protocols {  
            ospf {  
                area 10.1.2.3 {  
                    interface so-1/0/1.0;  
                    ...  
                }  
            }  
            class-of-service { .... }  
            rpm { .... }  
        }  
    }  
}
```

apply-macro data can drive
(or prevent) commit scripts
expansion



Op Scripts (JUNOS 7.5)



Op scripts

- Add custom commands to JUNOS
- Same driver as commit scripts
 - Not passed anything for input
 - Output is displayed to CLI user
- Can perform an XML API call
 - Restricted by user's permissions/authorization
- Inspect API results
 - Find interesting bits (with XPath)
- Repeat cycle: API call, inspect results

Three Common Threads

- Get better information
 - Display to user
 - Diagnose issues
- Standard Operating Procedures
 - Codify Best Practices in a script
- Testing
 - Find violations of policies
 - Report and/or fix them

Op Scripts (Operational-mode)

- "op" command: op filename name1 val1 name2 val2
- Support simple name/value arguments

```
phil@work> op ?
```

Possible completions:

<script>

Name of script to run

dead-peers

Diagnose issues with dead peers

op-bchip

B-Chip dump

op-host

simple reachability tests

```
phil@work> op dead-peers ?
```

Possible completions:

<[Enter]>

Execute this command

<name>

Argument name

detail

Display detailed output

peer

Peer to diagnose

|

Pipe through a command

```
phil@work> op dead-peers peer 10.1.2.3
```

Configuration

```
[system scripts op]
traceoptions {
    flag all;
}
file dead-peers.slax {
    description "Diagnose issues with dead peers";
    arguments {
        peer {
            description "Peer to diagnose";
        }
    }
}
file op-bchip.slax {
    description "B-Chip dump";
}
file op-host.xls {
    description "simple reachability tests";
}
```

Argument Definitions

- May appear in configuration
- May appear in the script itself

```
var $arguments = {  
    <argument> {  
        <name> "interface";  
        <description> "Name of interface to display";  
    }  
    <argument> {  
        <name> "protocol";  
        <description> "Protocol to display (inet, inet6)";  
    }  
}  
param $interface = "fxp0";  
param $protocol;
```

Example: dead-peers (slax)

```
param $peer;

match / {
    <op-script-results> {
        if ($peer) {
            call dead-peer($peer);
        } else {
            var $summary =
                jcs:invoke('get-bgp-summary-information');

            for-each ($summary/bgp-peer
                      [peer-state == 'Connect']) {
                call dead-peer($peer = peer-address);
            }
        }
    }
}
```

Example: dead-peers (cont'd)

```
template dead-peer ($peer) {
    var $query = <get-bgp-neighbor-information> {
        <neighbor-address> $peer;
    }
    var $answer = jcs:invoke($query);

    <output> "Peer: " _ $peer;

    var $info = $answer/bgp-peer;
    if ($info/last-error) {
        <output> "Last error was: " _ $info/last-error;
    }
    if ($info/last-state != 'Active') {
        <output> "Last state was: " _ $info/last-state;
    }
    /* ... */
}
```

Session: show-dead-peers

```
phil@work> op dead-peers peer 10.5.14.2
Peer: 10.5.14.2
Last error was: Cease
Last state was: OpenConfirm
Next-hop: 10.11.3.4
Interface: ge-1/2/3.0
Interface is up (14212 errors)
Next-hop is reachable via ping (5/5)
No authentication found in local configuration
No authentication errors found in syslog
...
phil@work>
```

Remote RPCs

- Network issues are seldom one sided
 - Diagnose scripts need access to remote device
 - Compare config to ensure both sides are consistent
 - Inspect remote state to see where the problem lies
 - Access to intermediate devices as well
- Configure services at one spot
 - Op script can change configuration on multiple boxes
 - Example: add a new device to my BGP mesh
 - Local config details all BGP peers (who need to change)
 - Loop thru all peers adding to new peer to each old peer's config while adding old peer to new peer's config
 - Test while you go to make sure peerage comes up

itr: Intelligent Trace Route

```
phil@dent> op itr target 10.5.20.2 sleep 0
Intelligent Traceroute for target 10.5.20.2

(0) dent is a m7i running 9.5I20090527_1849_phil (S/N A9673)
    Interfaces:
        out: fe-0/3/0.0 / 10.5.14.1 -> 10.5.14.2 (BGP route 10.5.20.0/24)
    Latency (ms):
        out: 2.199 +/- 2.545 (1.070 7.888 0.952 1.112 1.241 0.936 )

(1) iad is a m7i running 9.5I20090330_0030_danai-20 (S/N 39180)
    Interfaces:
        in: fe-0/0/3.0 / 10.5.14.2 -> 10.5.14.1 (Direct route 10.5.14.0/24)
        out: fe-0/0/2.0 / 10.5.16.1 -> 10.5.16.2 (OSPF route 10.5.20.0/24)
    Latency (ms):
        in: 1.409 +/- 0.423 (1.006 1.203 2.240 1.008 1.471 1.526 )
        out: 1.063 +/- 0.147 (1.042 1.248 1.145 0.993 1.161 0.791 )

(2) rdu is a m7i running 9.5R3.1 (S/N A9687)
    Interfaces:
        in: fe-0/2/0.0 / 10.5.16.2 -> 10.5.16.1 (Direct route 10.5.16.0/24)
        out: fe-0/2/2.0 / 10.5.20.1 -> 10.5.20.2 (Direct route 10.5.20.0/24)
    Latency (ms):
        in: 1.982 +/- 2.138 (1.010 1.025 0.822 6.758 1.195 1.082 )
        out: 0.850 +/- 0.105 (1.013 0.763 0.792 0.981 0.750 0.801 )

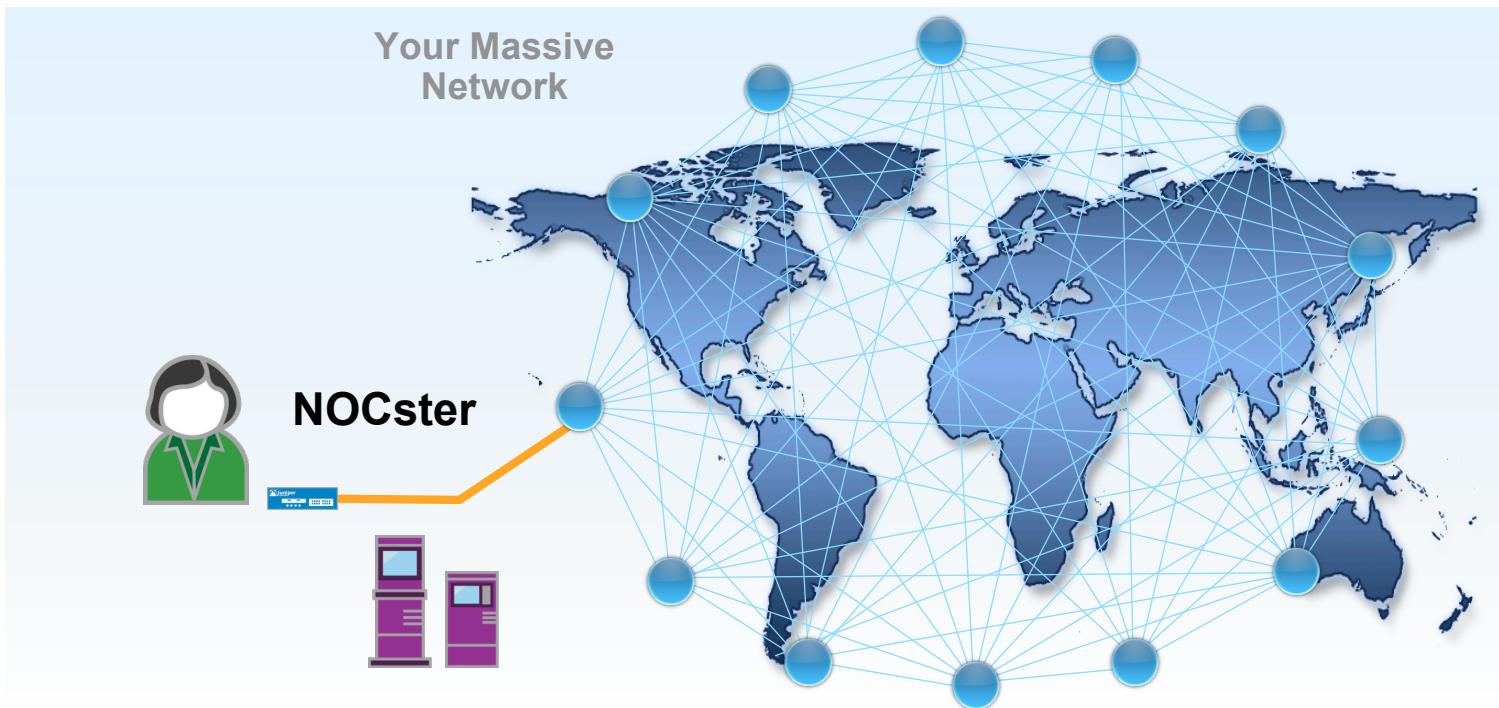
...
```

Interactive Scripts

- Op scripts can ask the user for input

```
var $response = jcs:get-input($prompt);  
var $password = jcs:get-secret("Enter password: ");
```
- Simple questions or complex interactions
- New "wizard" frameworks
 - Uses "guides" to build XML content
 - Can build JUNOS-style XML
 - Suitable for load-configuration
 - Can build arbitrary XML
 - Transform into JUNOS config for one or more boxes

Single point of configuration

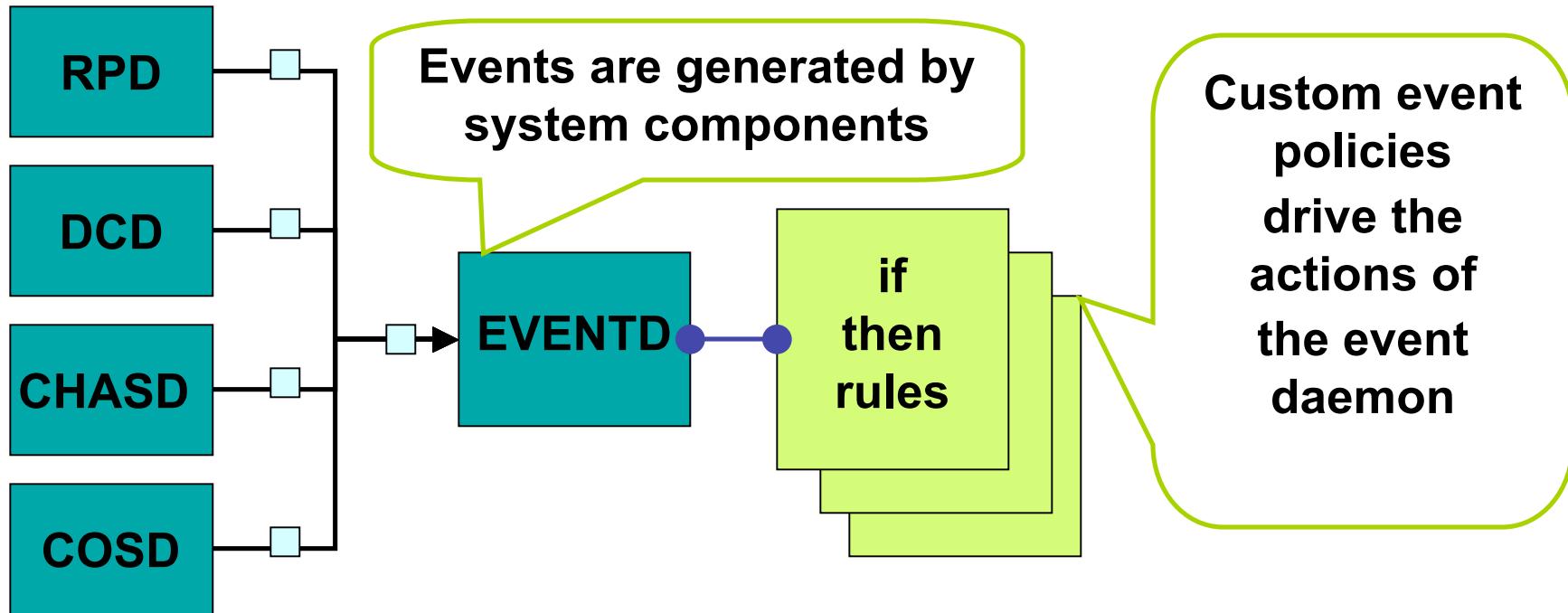


- Run an op script in one location to deploy a service on multiple devices
 - Config is generated by script
 - Then fed to other devices via remote RPCs
 - Config is guaranteed to be self-consistent
 - Fewer questions means fewer errors

Event Scripts (JUNOS 7.6)



Event Overview



Tag: RPD_LPD_SESSONDOWN

MSG: LDP session 10.1.2.3..1234 is down, reason: connection reset

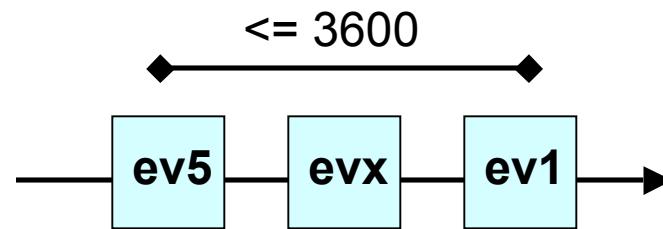
Attributes:

peer: 10.1.2.3..1234

reason: connection reset

Event Correlation

- Detect connected events
- Simple correlation
 - Dampen events



```
event-options {  
    policy dampen-policy {  
        events [ ev1 ev2 ev3 ];  
        within 3600 events [ ev4 ev5 ];  
        then {  
            ignore;  
        }  
    }  
}
```

Matching Attributes

- Test attribute values with attribute-match statement
 - "equals", "starts-with", and "matches" (regex)

```
[event-options policy test]
events RPD_RDISC_NOMULTI;
within 600 events RPD_KRT_IFDCHANGE;
attributes-match {
    RPD_RDISC_NOMULTI.interface-address
        equals RPD_KRT_IFDCHANGE.address;

    RPD_RDISC_NOMULTI.interface-name
        starts-with RPD_KRT_IFDCHANGE.ifd-index;
}
```

Example: interface-up-down

```
event-options {
    policy save-if-data {
        events [ SNMP_TRAP_LINK_DOWN SNMP_TRAP_LINK_UP ];
        then {
            execute-commands {
                commands {
                    "show interfaces {$$.interface-name}";
                    "show alarms";
                }
                output-filename if-status.txt;
                output-format text;
                destination my-server;
            }
        }
    }
}
```

Time-based Events

- Generate events
 - Periodic or time of day
- Use with event correlation
 - Ignore events during time window

```
[event-options]
generate-event MIDNIGHT {
    time-of-day 00:00:00;
}
generate-event HOURLY {
    time-interval 60;
}
```

Example: policy pic-restart

```
event-options {
    policy pic-restart {
        events [ PIC PFE ];
        attributes-match {
            PIC.message matches ".*PIC\(.).\.*chip . check_indq event 0xff.*";
            PFE.message matches ".*PIC\(.).\.*chip . check_indq event 0xff.*";
        }
        then {
            event-script pic-restart.slax {
                arguments {
                    syslog "{$$.message}";
                }
                output-filename pic-restart.txt;
                destination local;
            }
        }
    }
}
```

Example: pic-restart.slax

```
match / {  
    <op-script-results> {  
        var $fpc = jcs:regex('fpc([0-9])', $syslog);  
        var $pic = jcs:regex('ATMPIC\\(([0-9])\\)', $syslog);  
        var $pfe = jcs:regex('PFE-([0-9])', $syslog);  
  
        var $offline = <command> 'request chassis pic pic-slot '  
            _ $pic[2] _ ' fpc-slot ' _ $fpc[2] _ ' offline';  
        var $result_offline = jcs:invoke($offline);  
  
        expr jcs:sleep(15);  
  
        var $online = <command> 'request chassis pic pic-slot '  
            _ $pic[2] _ ' fpc-slot ' _ $fpc[2] _ ' online';  
        var $result_online = jcs:invoke($online);  
    }  
}
```

Event Scripts

- Possibilities
 - Change configuration
 - `jcs:dampen("change", count, minutes);`
 - SNMP
 - Generate traps
 - Fill in values in the JnxUtility MIB
 - Accounting Profiles
 - Record values in accounting profiles
 - Post-process and analyze data
 - Compare to yesterday or last week's data

Scripting Recap

Self-defense

Use Commit Script to fail the commit if the config breaks the rules

Self-healing

Change the config to fix problems at commit time so nothing is ever broken

Self-defining

Build your own data with apply-macro to encapsulate high-level constructs

Self-diagnosing

Op Scripts for trouble shooting determine root cause of errors

Self-monitoring

React to events in real time with Event Scripts

Additional Information:

[\(aka <http://tinyurl/ykb9xo7>\)](http://junos/juniper.net/content/Resources/Day_One_Guide/JSA1_DO_JUNOS_Auto_-_THIRD.pdf)

http://www.juniper.net/us/en/training/elearning/junos_scripting/index.html

<http://junos.juniper.net>

Thank you!

Phil Shafer
[<phil@juniper.net>](mailto:phil@juniper.net)