XMLCONF

IETF 57 – Vienna
Rob Enns (rpe@juniper.net)
XMLCONF Draft

- Current is draft-enns-xmlconf-spec-01
- Agenda
  - Very brief introduction to XMLCONF
  - Draft-01 changes
XMLCONF Authors
(in no particular order)

- Andy Bierman <abierman@cisco.com>
- Eliot Lear <lear@cisco.com>
- David Perkins <dperkins@dsperkins.com>
- Ted Goddard <ted.goddard@windriver.com>
- Phil Shafer <phil@juniper.net>
- Rob Enns <rpe@juniper.net>
- Ken Crozier <kcrozier@cisco.com>
- Steve Waldbusser <waldbusser@nextbeacon.com>
- Margaret Wasserman <mrw@windriver.com>
XMLCONF Strategy

• Create a standard operational framework for configuration
  – Allow for monitoring and notifications, but focus on configuration

• Separate the protocol from the data model
  – Allow for standard and proprietary content
  – Standardize the protocol first, and then start on content

• Create a transport independent, RPC-based configuration mechanism
  – XMLCONF over BEEP, SOAP, console

• Develop high level protocol operations common to most devices
  – Focus on transactions
XMLCONF Strategy

• Allow implementation to mirror native capabilities of device
  – Text-based technology such as XML permits tight integration with CLI
  – No feature lag between XMLCONF and CLI
Session Management

- **Management channel**
  - Session control; creation of other channels
  - Abort command kills current command on the operations channel
  - Kill-session used to terminate the session of another user
- **Operations channel**
  - Used for RPC requests and replies
- **Notification channel**
  - Optional channel for asynchronous messages
RPC Model

- `<rpc>`
  - Request on operations channel
- `<rpc-reply>`
  - Reply sent on operations channel
- `<rpc-progress>`
  - Provides progress reports (percentage completion) for long duration RPC operations, sent on the management channel
- `<rpc-abort>`
  - Provides a way to abort an RPC in progress, or queued for processing, sent on the management channel
- `<rpc-abort-reply>`
  - Abort RPC reply, sent on the management channel
RPC Model: Error Reporting

- `<rpc-error>`
  - Included in `<rpc-reply>` if an error occurs during processing of an RPC request

- `<ok>`
  - Included in `<rpc-reply>` if no error occurred during processing of an RPC request
Operational Model

• Configuration datastores

- Different variants of this model are possible
Protocol Operations
<get-config>

• Used to retrieve all or part of a configuration

<rpc message-id="101">
  <get-config>
    <source><running/></source>
    <config xmlns="http://example.com/schema/v2.1/config">
      <users/>
    </config>
  </get-config>
</rpc>
Protocol Operations
<get-config>

<rpc-reply message-id="101">
  <get-config>
    <config xmlns="http://example.com/schema/v2.1/config">
      <users>
        <user>
          <name>root</name>
          <type>superuser</type>
        </user>
        <user>
          <name>fred</name>
          <type>admin</type>
        </user>
      </users>
    </config>
  </get-config>
</rpc-reply>
Protocol Operations
<edit-config>

- Used to modify a configuration
- Parameters:
  - target: (@config-name)
  - test-option: (test-then-set | set) [default: set]
  - error-option: (stop-on-err | ignore-err) [default: stop-on-err]
  - format: (xml | text)
  - config: (@URL | @config-name | @element-tree | text)
Protocol Operations

<edit-config>

<rpc message-id="102">
  <edit-config>
    <target><running/></target>
    <config xmlns="http://example.com/schema/v1.2/config">
      <interface>
        <name>Ethernet0/0</name>
        <address>
          <ipv4>1.2.3.4</ipv4>
          <ipv4-mask>255.0.0.0</ipv4-mask>
        </address>
      </interface>
    </config>
  </edit-config>
</rpc>

Reply is:

<rpc-reply message-id="102">
  <ok/>
</rpc-reply>
Protocol Operations

- `<copy-config>`
  - Copy configuration to/from a configuration datastore
- `<delete-config>`
  - Delete a configuration datastore
- `<get-state>`
  - Get operational state
Protocol Operations

• `<lock>`, `<unlock>`
  – Locking for configuration datastores
  – Requires lock capability

• `<validate>`, `<commit>`
  – Validate configuration without committing
  – Commit (activate) configuration
  – Requires candidate configuration capability
Notifications

- An XMLCONF peer advertising the 'notifications' capability supports the notification channel
- Used for sending asynchronous messages
- `<open-notifications>` operation requests opening notification channel with specific parameters
  - format: rfc3195 is the only legal value (in v1.0)
- `<close-notifications>` requests that the notification channel be closed
draft-01 changes

• Section 3
  – Change “id” attribute to “message-id”
  – Make “message-id” mandatory
  – Add <rpc-error> example
draft-01 changes

• Section 5
  – <edit-config>: Allow an "operation" attribute to indicate the desired operation (merge, replace, or delete)
  – This attribute makes the "write-option" parameter superfluous, so remove it
  – Add more <edit-config> examples
<edit-config> replacing config

<rpc message-id="107" xmlns="http://ietf.org/xmlconf/1.0/base">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns="http://example.com/schema/1.2/config"
             xmlns:xc="http://ietf.org/xmlconf/1.0/base">
      <interface xc:operation="replace">
        <name>Ethernet0/0</name>
        <mtu>1500</mtu>
        <address>
          <name>1.2.3.4</name>
          <mask>255.0.0.0</mask>
        </address>
      </interface>
    </config>
  </edit-config>
</rpc>
<rpc message-id="107" xmlns="http://ietf.org/xmlconf/1.0/base">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns="http://example.com/schema/1.2/config"
            xmlns:xc="http://ietf.org/xmlconf/1.0/base">
      <interface xc:operation="delete">
        <name>Ethernet0/0</name>
      </interface>
    </config>
  </edit-config>
</rpc>
draft-01 changes

• Section 6
  – Fix lock and unlock examples

• Section 7
  – Drop canonical XML requirement
  – Add “no embedded DTD” requirement
    (was in canonical XML)
draft-01 changes

- Section 9
  - Make “message-id” attribute mandatory in the XSD
- General
  - Fixed a number of namespace issues in the examples
  - Use empty tags where appropriate now that canonical XML is gone