EOS OOPS
Object Oriented PDUs for SNMP

Wes Hardaker
<hardaker@tislabs.com>

draft-hardaker-eos-oops-02pre.txt

2002.Nov.20

Status

- EOS Moving forward -> rewriting:
  - ASN.1
  - formal text

- Operations supported in -02pre:
  - Get-Object-PDU   -> Get-Object-Response-PDU
  - Write-Object-PDU -> Write-Object-Response-PDU

- Feedback needed on Write support

Changes from Yokohama:

- SMIv3 almost supported
  - last minute changes, ...
- cursor field added to GO(R)Ps
- ASN.1 started.
- search-criteria now supports AND/OR/NOT
- errors are now SEQUENCE OFs such that:
  - multiple errors can be returned for a given request
  - no errors is only a 2 byte empty SEQUENCE OF tag

To do:

- Finish SMIv3 support?
- Fix outstanding issues
- Finish writing the textual descriptions
- Notification support?
- Integration with SNMPv3 architecture

Help needed from:

- SNMPv3 architecture experts
- ASN.1 experts
- XML experts
- Implementors!

Why you should look at write support:

- Better grouping transaction models:
  - doAll, tryAll, doAtLeastOne, tryAtLeastOne
- Support for operations:
  - Create, Modify, Delete
- Modification via search-parameters
  - Modify all rows where ...
- Ordered vs Unorderd opertations.
  - "execute in any order" vs
  - "execute in this order"

Questions for the WG:

- Support SMIv3?
  - SMIv3 lacks editors and may be shut down
  - Which should be the priority v2 or v3?
  - Support complex depth operations?
  - Imposes some complexity
    - (can be minimized, of course)

Questions for the WG:

- Define notifications too? (NOP/NORP)

Questions for the WG:

- Return search-criteria field?
  - Is there *any* manager that doesn’t save state for requests???
  - Might be useful for sniffing

Search criteria issues

- Desire indicated for complex expressions:
  - GOP Table where columnX > columnY + columnZ
- Counter value searching.
  - GOP ifTable where diff(ifOutOctets) > 10000?
  - Implementation notes:
    - where := { DIFF, OLDVALUE, 10000 }

Augmentation Retrieving

- Augmentation table implementations must be either:
  - Implemented in conjunction with the indexing data
    - 99% of the time?
  - Implemented differently
    - 1% of the time?

Augmentation Retrieving

- Issue is that:
  - Joins are hard at the agent, if they have to.
  - If data is already aligned, then the responses will already be aligned anyway.
-

Augmentation Retrieving

- Specifically, 1 operation ok?:
  - GOP ifTable and ifXTable
- Or two (but same PDU):
  - GOP ifTable
  - GOP ifXTable

Augmentation Searching

- Which of the following should be possible:
  - (note, this pretty much requires a join)

```
GOP ifTable.ifSpeed
        where ifXTable.ifName = something

GOP ifXTable.ifHCOutoctets
        where ifTable.ifType = 6
```