Java SNMP Oplet

Tal Lavian
Goals

• Portable across a range of devices
• Extensible
• Simple and convenient for client use
• Consistent with SNMP model
• Hide unnecessary SNMP details
• Permit optimized access
• Re-use MIB documentation
API Generation

• Most of the Java code is generated automatically
• ASN.1 MIB definitions are converted into Java classes
• Documentation and commentary in the MIB definitions is placed as Javadoc formal comments
• HTML documentation generated from Javadoc
MIB Objects

• The MIB data model is structured as a tree
• API represents MIB groups with Java classes
• MIB variables are represented with accessor methods
• Conceptual tables are represented with iterators
• API converts SNMP data values into standard Java types
Variable access

- API uses a MIB Map to dispatch requests to variable access routines
- Different parts of the MIB tree can be serviced by different mechanisms
- Two main schemes:
  - An ad hoc interface to the SNMP instrumentation layer
  - A generic SNMP loopback
Advantages of MIB map

• Allows immediate generic implementation of the entire MIB via the loopback scheme
• Enables optimized native implementation of key MIB variables for maximum efficiency
• Permits definition of pseudo-MIB variables for extending MIB dynamically
• Provides site for centralized access management
Proxy mode

- Uses SNMP loopback mechanism to target a remote network element
- API can be used to control devices that don’t have an embedded JVM

Diagram:
- Java Server
- Java "Optlet"
- JVM on PC
- SNMP
- Cisco Router with No JVM