PRACTICAL EXERCISE 1 (W1_PE1) : SNMPWORKSHOP BASICS

This practical exercice gives an overview of the SnmpWorkshop toolset and provides instructions to deploy it.

SnmpWorkshop is a Java-based toolset allowing to develop efficiently SNMP V1-based management prototypes for studying purposes. In summary, SnmpWorkshop is a Java library/API that provides an object-oriented point of view on the SNMP V1 management protocol, as well as tools to facilitate the implementation of agents and managers. It consists of :

- A framework that handles management processes and provides SNMP V1 stacks; it is implemented by libraries that are linked to the application. The framework has a simple, object-oriented API;

- A compiler called `smic` (SMI Compiler) that plays an essential role : it generates attribute tables for the manager application and the agent, as well as Java squeletons of accessors to the managed objects.

The toolset is complemented by a generic graphical SNMP browser playing the SNMP manager role.

The development process using SnmpWorkshop distinguishes two phases : design and execution.

- During the design phase, the MIB specification is compiled by the SMIC tool to generate Java implementation squeletons (called "Java accessors") of managed objects and attribute tables (Java model of SMI attributes) that will later be linked to libraries (framework) to build a working Agent. On the other hand, the SNMP Manager application would make profit of this offline phase to read the MIB specification as well.

- When the design phase is finished, a build command generates a full working Agent bearing previous generated parts, as well as the software infrastructure: the SnmpWorshop framework.

- During the execution phase, the manager sends its requests to the Agent ; the underlying SnmpWorkshop framework will forward the request to (either generated or handcoded) implementation code that will execute the request by manipulating equipment resources in an application-specific way.
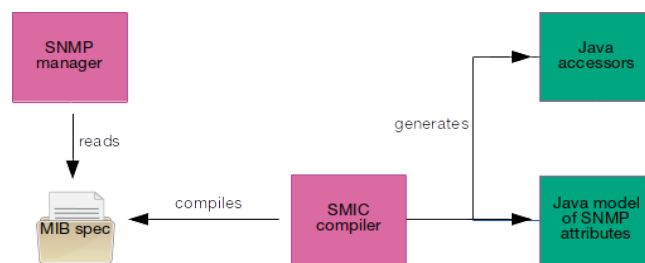
SnmpWorkshop during design phase
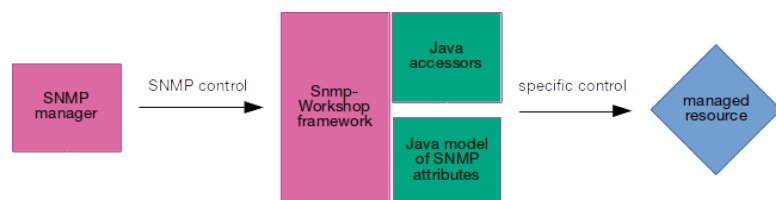
SnmpWorkshop during execution phase

1. SnmpWorkshop deployment

All tools, software and libraries are contained in a single archive. Just open a terminal, travel to the right spot and unarchive it to deploy instantly. Simply type:

```
$ cd /home/user/snmp-lab
$ tar zxf SnmpWorkShop-nojava.tgz
```

2. SnmpWorkshop development environment

The first thing to do at this stage is to go to the directory of the freshly deployed environment:

```
$ cd SnmpWorkShop-nojava
```

> **Note:** Each time that you start a new SnmpWorkshop work session with a fresh terminal, you **MUST** configure the shell environment variables by typing:
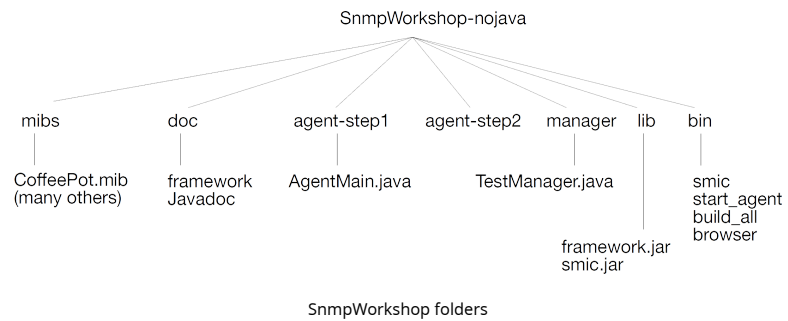>
> ```
> source set_envt.sh
> ```
>
> Otherwise the different tools of the lab will not work!

libraries, commands, sample MIBs etc.

```
$ ls
```

Please do not hesitate to have a look and to get a feeling of the different areas, such as depicted by the picture below:

SnmpWorkshop folders

The different folders of SnmpWorkshop correspond to:

- **mibs:** this folder contains some usual MIB specification files such as RMON, MIB-II and others; and of course, you'll find there the MIB specification used in the lab: `CoffeePot.mib`.

- **agent-step1/agent-step2:** those are the folders you'll work in most of the time. You will find there some code for a minimal SNMP Agent: `AgentMain.java`.

- **manager:** this is the folder supporting a representative but simple SNMP Manager application you'll work with as well.

- **lib:** this folder contains the major support code for the SNMP framework and the SMIC compiler. This code is wrapped into Java libraries (.jar files).

- **bin:** this folder contains all lab-specific commands such as `start_agent`, `smic` etc. The `browser` command starts a generic graphical SNMP Manager tool that will be used to perform some tests.

- **doc:** this folder bears a complete Javadoc documentation for the SNMP framework. Students familiar with Java should not hesitate to consult it!

QUESTION W1.PE1.1 : SMI (1/1 point)

SMI is an acronym that has been used in the above course material.

What does SMI mean? (NA=1)

- ○ Simple Management Interface

- ○ Snmp Meta Interoperation

- ◉ Structure of Management Information ✔

- ○ Structure of Middleware Information

Correct: SMI defines the data

Vous avez utilisé 1 essais sur 3

QUESTION W1.PE1.2: DEPLOYMENT (1/1 point)

The lab needs basic Linux shell manipulation skills. Did you succeed in deploying the archive ? In positioning the environment variables ? This can be validated by typing the "smic" command at the prompt, with no arguments.

What is the result of the smic command? (NA=1)

- ○ "Command not found"

- ◉ A "Usage" explanation ✔

Correct: Bravo! Everything worked well!

QUESTION W1.PE1.3: SMIC (1/1 point)

What is the purpose of the smic command? (NA=1)

○ To check code validity

○ To read SNMP Protocol Data Units

◉ To compile the MIB file for generating Java code reflecting the SNMP attributes of the MIB ✔

○ It does not check the syntax of the MIB definitions

Vous avez utilisé 1 essais sur 3

QUESTION W1.PE1.4: FRAMEWORK (1/1 point)

The SNMPWorkshop "framework" takes in charge: (NA=1)

○ SNMP alarm display

○ SNMP attribute generation

◉ Generic SNMP Agent functionality ✔

Vous avez utilisé 1 essais sur 3