Tutorial Example

MIB Design and SNMP Agent Code Generation

Universidade do Minho
v1.0 February 2021

João Pereira, d12267@di.uminho.pt
Bruno Dias, bruno.dias@di.uminho.pt

Context

In this tutorial we want to create a MIB and generate the agent code for that MIB. The MIB will allows us to register/present some information of the GR students. The MIB as the following structure:

- grStudentsMIB
  - .1 – total number of students [Read][Counter]
  - .2 – total number of students from MIETI [Read][Counter]
  - .3 – total number of students from MIEI [Read][Counter]
  - .4 – total number of students from MERSTel [Read][Counter]
  - .5 – total number of students from MEI [Read][Counter]
  - .6 – table
    - .6.1 – table entry
      - .6.1.1 – index [Read][Counter]
      - .6.1.2 – number [Read][Integer]
      - .6.1.3 – course [Read][OctetString]
      - .6.1.4 – name [Read][OctetString]
      - .6.1.5 – email [Read][OctetString]
      - .6.1.6 – tp status [Read/Write][Integer]

0 - no delivery	1 - TP1 delivered
2 - TP2 delivered	4 - TP3 delivered
3 - TP1 + TP2 delivered	5 - TP1 + TP3 delivered
6 - TP2 + TP3 delivered	7 - TP1 + TP2 + TP3 delivered

Software

In order to design our MIB we will use the MIB Designer. We will then use the output of the MIB Designer in the AgenPro so we can generate the Agent Code and the MIB module code.

Name	License	Evaluation Goal	Link
MIB Designer 5.x	Evaluation Version	Educational	evaluation request
AgenPro 5.x	Evaluation Version	Educational	evaluation request

MIB Designer

Start the MIB Designer 5.x by running: `java -jar mds-5.0.3.jar` then enter the License Key that you received via email. After that the software will ask to install some files, choose any directory that serves that purpose.

- step1 - start a new MIB: File -> New -> In current Windows (Ctrl-N)
- step2 - follow the wizard:
  - "Common Object Name Prefix": grMIB
  - "New MIB Module Name": GR-STUDENTS-MIB
  - Next
  - Check "Import enterprises from SNMPv2-SMI": OID 8888 (any number will do)
  - Next
  - Check "Create default top level MIB structure"
  - Check "Create basic object groups"
  - Finnish
- step3 - cleanup
  - Edit the MODULE-IDENTITY (first after enterprises), mouse right-click -> Edit: Rename (grMIBRegMIB -> grMIB) add description
  - Delete grMIBObjects and grMIBEvents, mouse right-click -> Remove
- step4 - add new objects to your MIB:
  - Right-click in the MODULE-IDENTITY (first after enterprises) -> Add -> Object-Type (or Table)
  - Create the objects according to the desired MIB structure.
  - Every object that you add, needs to be also added to the OBJECT-GROUP defined in your MIB. Nested objects are not added automatically (object in the entry of our table) they need to be added manually to the Basic OBJECT-GROUP (you can edit this object to do so) or you can create an OBJECT-GROUP for the table.
- step5 - OID coherence: iterate all the created object and verify if the OIDs are according to your design.
- step6 - save: File -> Save (only pay attention to errors after saving)
- step7 - Export: File -> Export MIBs plain text, select you MIB from Available MIBs, choose the output directory, OK.

AgenPro

Start the AgenPro 5.x by running: `java -jar agp-5.0.0.jar` then enter the License Key that you received via email. After that the software will ask to install some files, choose any directory that serves that purpose.

- step1 - Import MIB file: File -> Import MIB File (Alt - I): choose the output file from MIB Designer (.txt)
- step2 - Open MIB: File -> Open/Close MIBs: select the desired MIB (it should be under Loaded Modules). When asked "Add loaded MIB modules to current code generation project?" select "Yes"
- step3 - edit project: Project -> Edit
- step4 - Project wizard:
  - choose template root dir: this dir should be under the installation directory that you select at the first start of the program. <installation dir> -> templates -> snmp4j-agent_3_1
  - choose the root in/output dir: create a dir that serves this purpose.
  - job 1 - click in Add New — Execution Type: By selection; Generation Template: java_code.vm (from the Templates root path); File Name Template: java_filename.vm (from the Templates root path); Output Directory: create a new dir (inside the in/output root path); Selection Template: select_1module1file.vm (from the Templates root path); OK
  - job 2 - Click in Add New — Execution Type: Once; Generation Template: java_init_code.vm (from the Templates root path); File Name Template: java_init_filename.vm (from the Templates root path); Output Directory: same as job 1.; OK
  - job 3 - Click in Add New — Execution Type: Once; Generation Template: java_agent_main.vm (from the Templates root path); File Name Template: java_agent_main_filename.vm (from the Templates root path); Output Directory: same as job 1.; OK
  - Next
  - Define Properties - Click in Add New and then in Edit: Property Name: package; Property Value: org.snmp4j.agent.<what ever you like>; OK
  - Finish
- step5 - Generate the code: Project -> Generate; the files should now be in the selected output dir.

Disclaimer

This is a very fast approach to the problem at hands. You are here incentive to explore a better MIB structure and a more refined Code generation.

You can explore the PDF manuals present in the installation directory of each Program.