

How-to integrate generated code for SNMP4J-Agent into the agent's main program

This page describes how-to integrate AgenPro generated code for SNMP4J-Agent into an agent's main program.



Not Applicable for Maven-Task-Plugin Usage

Although this description uses AgenPro Maven-Plugin files, it steps described herein are not useful when actually also using the AgenPro-Maven-Plugin. The plugin facilitates already many of the below steps. Please read the [AgenPro Manual](#) for details.

Prerequisites

1. The code has been generated by AgenPro into a folder that can be used to further develop the agent software. Otherwise, modifications to the generated code cannot be (easily) integrated into regenerated code, for example if there is a MIB specification update. For this tutorial that **gen-folder** is C:\Users\myuser\Documents\agenpro4\generated\snmp4j\src\org\snmp4j\agent\tutorial. You can replace the name **gen-folder** in the following by this folder.
2. *Optionally:* Install the latest SNMP4J.jar, SNMP4J-Agent.jar, and optionally SNMP4J-AgentX.jar in a **lib** directory. For this tutorial we use C:\Users\myuser\Documents\agenpro4\lib for that. You can download latest release JARs from the AGENT++ Maven repository at: <https://oo.snmp.net/dist/release>.
3. Install Java JDK (the JRE is not sufficient, because you need the Java compiler).
4. *Optionally:* Install Maven 3.x. Note: If you do not install Maven, you need to perform step 2 and manually compile the sources (not explained here).
5. If you have already a main program for your SNMP4J-Agent ready, then continue with step ?? of the Step-by-step Guide below.

Step-by-step Guide

1. *Optionally:* **Only if you are using your own main agent program implementation**, you need first to fix the import section as described in the box below and then add the following function `registerMIBs()` to that class:

Import Statements for Integration

```
// Alternatively you can also import the complete package you specified in AgenPro project wizard step 1:
// import <package>.*;
import org.snmp4j.agent.tutorial.Modules;
import org.snmp4j.agent.tutorial.Snmp4jAgentTutorialMib;
```

Code to Register Generated MIBs

```
/**
 * Register your own MIB modules in the specified context of the agent.
 * The {@link MOFactory} provided to the <code>Modules</code> constructor
 * is returned by {@link #getFactory()}.
 */
protected void registerMIBs()
{
    if (modules == null) {
        modules = new Modules(getFactory());
    }
    try {
        modules.registerMOs(server, null);
    }
    catch (DuplicateRegistrationException drex) {
        logger.error("Duplicate registration: "+drex.getMessage()+" "+
            "MIB object registration may be incomplete!", drex);
    }
}
```

Then call the above function in your agent initialization code before the agent is actually started with `agent.run()`:

Call Module Registration

```
public void run() {  
    // initialize agent before registering our own modules  
    agent.initialize();  
    // this requires sysUpTime to be available.  
    registerMIBs(); // <---- here the MIB module registration is called  
    // add proxy forwarder  
    agent.setupProxyForwarder();  
    // now continue agent setup and launch it.  
    agent.run();  
}
```

2. Compile the sources with Maven:

Agent Compilation

```
cd C:  
cd \Users\myuser\Documents\agenpro4\generated\snmp4j\  
mvn clean install
```

3. Run the agent:

```
java -jar target\tutorial-1.0.0-SNAPSHOT-jar-with-dependencies.jar udp:0.0.0.0/161
```

or alternatively without Maven support:

Run the Agent

```
cd ..\..\..\..  
java -cp .;..\lib\* org.snmp4j.agent.tutorial.Agent udp:0.0.0.0/161
```



Related articles

- [How-to create your own SNMP agent with AGENTPP tools?](#)
- [How-to integrate generated code for SNMP4J-Agent into the agent's main program](#)