What is the difference between SNMP4J-SMI and SNMP4J-SMI-PRO?

The differences between the two APIs are

1. the license model
2. the features

License Model

The SNMP4J-SMI license is an user license intended for personal use for individuals (not for companies). Although the 1.x license covers also the usage for comany employees, this will be changed with the 2.0 release.

The SNMP4J-SMI-PRO license is a development site license for all employess at a specific company site (= postal address).

For both licenses, there are no runtime fees.

Features

SNMP4J-SMI is designed for easy integration into SNMP4J and provides basic OID to object name and variable value to human readable string conversion based on (runtime) MIB information.

SNMP4J-SMI-PRO includes all features of SNMP4J-SMI plus full access to all MIB information, including description, object syntax, default values, etc.

Functions (of the SmiManager class)	Description	SNMP4J-SMI	SNMP4J-SMI-PRO

`compile(File textOrZipFile)`
 	
`compile(NamedInputStream mibFile)`
 	
`compile(NamedInputStream[] files, CompilationMonitor compilationMonitor, SmiCompiler.TargetMode compilerTargetMode, SmiCompiler.OverwriteMode overwriteMode, SmiCompiler.Strictness strictness)`
	Compile one or more MIB specification files into the internal MIB repository (memory or persistent cache).	✅	✅

`deleteModule(String moduleName, boolean forceDeletion)`	Remove a MIB module from the internal MIB repository (cache)	✅	✅
`listModules()`	List all available MIB modules in the internal MIB repository (cache)	✅	✅

`loadModule(String moduleName)`
 	
`unloadModule(String moduleName)`
	Load/unload a MIB module from the MIB repository into the SmiManager memory (i.e., activate a MIB module for usage by the API)	✅	✅

`format(int[] value)`
 	
`formatForRoundTrip(int[] value)`
 	
`format(OID instanceOID, Variable variable, boolean withOID)`
	Format an numeric OID value to a human readable object identifier with object name prefix.
	
	Format a SNMP value to a human readable string using MIB information associated with the given OID.
		✅	✅

`parse(String text)`
 	
`parse(int smiSyntax, String text)`
 	
`parse(OID classOrInstanceOID, String text)`
 	
`parseVariableBinding(String text)`
	Parse an OID string or variable string to an OID value and VariableBinding respectively.	✅	✅

`getIndexVariables(OID table, OID index)`	Gets the index Variable array for the sub-index values of a SNMP table from the given index OID (suffix).	✅	✅

`findSmiObject(OID oid)`
 	
`findSmiObject(String moduleName, String objectName)`
	Find and return the SmiObject with the given object name in the given MIB module or by the given OID.		✅

`findSmiModule(String moduleName)`	Find and return the SmiModule with the specified name.		✅
`findSmiObject(String moduleName, SmiObjectFilter<S> filter)`	Find a SmiObject by the given filter.		✅