# How to use the extended features SNMP4J-SMI-PRO provides?

The extended features (compared with SNMP4J-SMI) the PRO edition provides are accessed through the `SmiManager.find*` methods:

```
SmiManager smiManager;
// Define the MIB repository
smiManager = new SmiManager(null /* Put your license key here */, new MemRepositoryDriver());
// Load MIB modules from the repository
String modulePatterns = "IF-.*";
loadModules(smiManager, modulePatterns);
// Alternatively compile some new MIB specifications and load them on-the-fly:
File[] files = new File[] { new File("IF-MIB.txt") };
List<CompilationResult> result =
              compiler.compile(files, null /* no monitoring*/,
                               SmiCompiler.TargetMode.storeIntoRepositoryAndLoad,
                               SmiCompiler.OverwriteMode.overwriteIfNewer,
                               SmiCompiler.Strictness.standard);

// Here the formatting with MIB information is finally configured in SNMP4J:
SNMP4JSettings.setOIDTextFormat(smiManager);
SNMP4JSettings.setVariableTextFormat(smiManager);
smiManager.setOidFormat(SmiManager.OIDFormat.ObjectNameAndDecodedIndex4RoundTrip);
smiManager.setOidVariableLengthStringQuote(smiManager.getOidFixedLengthStringQuote());
// Activate your own error messages (if needed), othwerwise comment out the following line:
smiManager.setSmiErrorTextResourceBundle("SampleSmiErrorResourceBundle");

// Use the extended features:

// We can cast here to SmiObjectType because we know that ifAdminStatus is an OBJECT-TYPE.
// To be 100% safe, we would have to check it with insteanceof...
SmiObjectType smiObjectType = (SmiObjectType)smiManager.findSmiObject(new OID("ifAdminStatus"));
System.out.println(smiObjectType.toString());
String description = smiObjectType.getDescription();
SmiSyntax syntax = smiObjectType.getSyntax();
String units = smiObjectType.getUnits();
...
```

Loading modules based on regex patterns can be implemented like this:

```
public static void loadModules(SmiManager smiManager, List<String> modulePatterns) throws IOException {
  String[] availableModules = smiManager.listModules();
  for (String modulePattern : modulePatterns) {
    Pattern p = Pattern.compile(modulePattern);
    boolean matched = false;
    for (String availableModule : availableModules) {
      Matcher m = p.matcher(availableModule);
      if (m.matches()) {
        matched = smiManager.loadModule(availableModule);
        System.out.println("Loaded MIB module '"+availableModule+"'");
      }
    }
    if (!matched) {
      System.err.println("No module loaded for pattern '" + modulePattern + "'");
    }
  }
}
```