```
print("Discovering...");

importlog = System.getenv("IMPORTLOG");
exportmodel = System.getenv("EXPORTMODEL");

//-----------------------------------------------------
print("<import file=\"" + importlog + "\">");

// Use the name of the plugin (in lowercase) to open a log
print("    <pluginfo>");
org.deckfour.xes.model.XLog log = open_xes_log_file(importlog);
print("    </pluginfo>");

logname = org.deckfour.xes.extension.std.XConceptExtension.instance().extractName(log);
print("    <name>" + logname + "</name>");
print("    <traces>" + log.size() + "</traces>");
events = 0;
for (trace : log) {
   events += trace.size();
}
print("    <events>" + events + "</events>");
print("    <classifiers>" + log.getClassifiers().size() + "</classifiers>");
print("</import>");

//-----------------------------------------------------
print("<discover miner=\"HybridILPMiner\">");

clss = (log.getClassifiers().isEmpty() ? new XEventAndClassifier(new XEventNameClassifier(), new
XEventLifeTransClassifier()) : log.getClassifiers().get(0));

print("    <pluginfo>");
res = ilp_based_process_discovery(log, clss);
print("    </pluginfo>");

model = res[0];
marking = res[1];

print("    <name>" + model.getLabel() + "</name>");
print("    <transitions size=\"" + model.getTransitions().size() + "\">");
for (transition : model.getTransitions()) {
   print("        <transition>" +
transition.getAttributeMap().get(org.processmining.models.graphbased.AttributeMap.LABEL) + "</transition>");
}
print("    </transitions>");
print("    <places size=\"" + model.getPlaces().size() + "\">");
for (place : model.getPlaces()) {
   print("        <place>" + place.getAttributeMap().get(org.processmining.models.graphbased.AttributeMap.LABEL) + "
</place>");
}
print("    </places>");
print("    <marking>" + marking + "</marking>");
print("</discover>");
//-----------------------------------------------------
```

```
print("<export file=\"" + exportmodel + "\">");

exportfile = new java.io.File(exportmodel);

print("    <pluginfo>");
pnml_export_petri_net_(model, exportfile);
print("    </pluginfo>");

print("    <bytes>" + exportfile.length() + "</bytes>");

print("</export>");

exit();
```