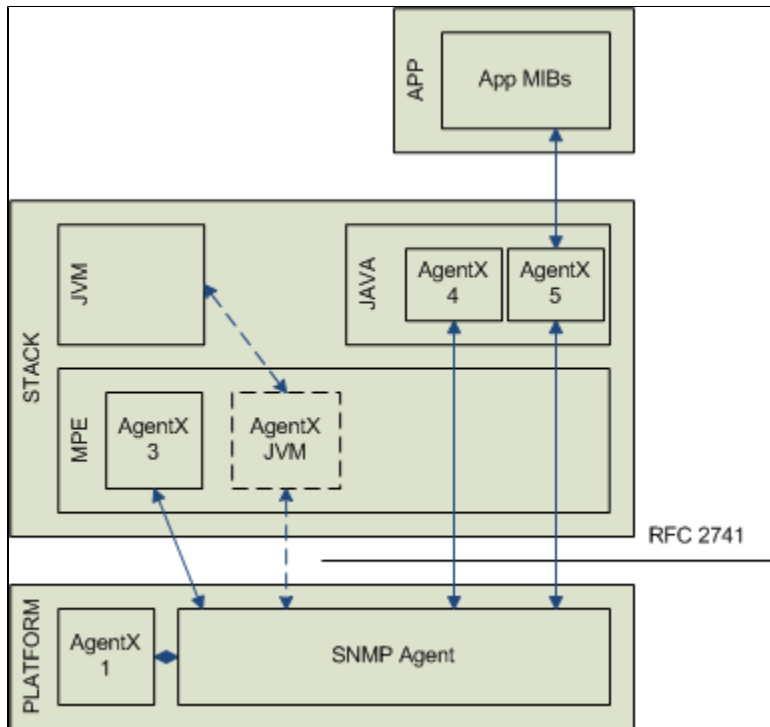


SNMP

SNMP Design Decisions

Through a series of meetings spanning September through December of 2010, it was decided that the master SNMP agent would reside in non-portable C code (MPEOS / platform code). For the first phase, this would specifically be in the platform and not MPEOS. A subsequent phase (beyond release 1.2-B) will have the Master Agent in MPEOS. Furthermore, it was decided that the individual code modules that supply the actual management data would register and report their information through the AgentX protocol by registering sub-agents with the main SNMP agent. The transport would be implemented over TCP/IP for AgentX to/from all sub-agents and UDP/IP for version 2 SNMP access to the Master agent.

<Master Agent in Platform>



SNMP Integration

Since NDS needed to work in parallel with CableLabs to obtain SNMP support in a timely fashion, they decided to implement a stand-alone Master agent (i.e. a separate executable that would require instantiation along side the OCAP RI). This was to be a temporary solution as CableLabs has found multiple executable scenarios problematic in the past (e.g. VLC, HDHomerun, HE4RI, etc.).

[OCAP RI Branch 1.1.4-H and I releases](#)

The 1.1.4 branch does not have the integrated Net-SNMP Master agent. It was decided that its inclusion would cause more work than time allowed. Therefore, the 1.1.4 branch still uses NDS' stand-alone Master agent executable (since this work has not been contributed to the repository it is therefore not tested at CableLabs.) As NDS finds issues with the SNMP contribution they are supplying them back via JIRA where the issue is tested against the 1.2 trunk and the fix (if supplied) then integrated.

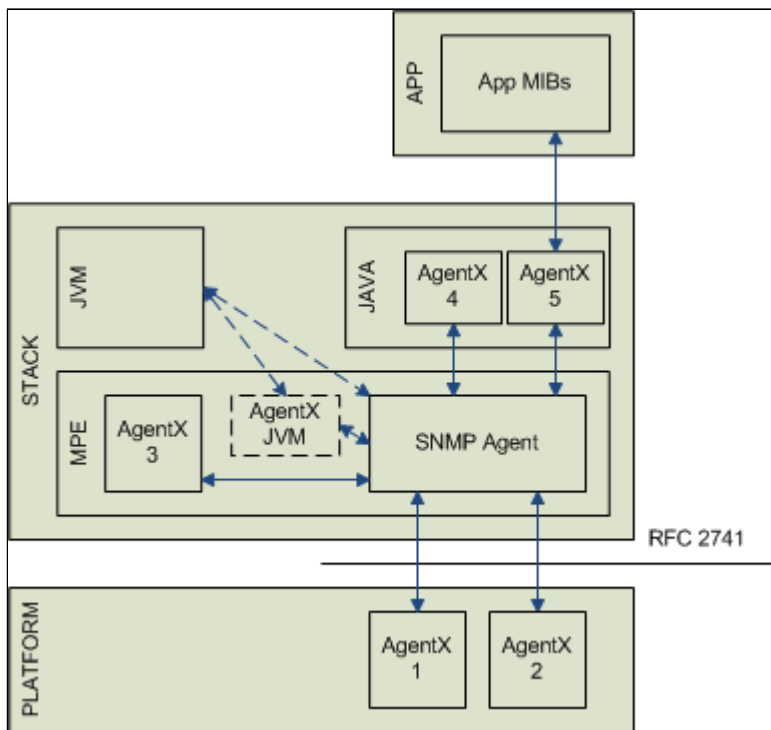
[OCAP RI Trunk 1.2-A and 1.2-B releases](#)

As of February 2011 in a pre-release of 1.2A, the Net-SNMP library and a CableLabs version of a stand-alone agent (configurable as a Master or Sub) was available to be used. In the official release of 1.2A the Master agent is running in the platform code and the NDS contributed Java AgentX stack and sub-agent are running in the OCAP RI stack. This implementation includes default handlers for the required MIBs and many of these get overridden once the stack initializes and its sub-agent(s) register with the master. It should be noted that these handlers are essentially stubs that return hard-coded values and may be replaced over time with actual handlers that obtain the correct object ID value.

[OCAP RI Trunk 1.2-C release and beyond](#)

Moving forward the Master SNMP Agent is intended to reside in the MPE platform independent C code layer, modified to access network facilities using the mpeos_socket porting API.

<Master Agent in MPEOS>



SNMP Testing

As described below, CableLabs uses iReasoning as a Network Manager to access the Master agent in the OCAP RI. Specific MIB object IDs are selected to exercise code paths to the Master agent and also to at least one of its sub-agents. Moving forward specific CTP tests will be developed to test these kinds of scenarios and exploit the different data types reported as well as the boundaries those data types support.

SNMP Library and Agent

The Net-SNMP library has been added to the RI_Platform build. In addition, an agent has been added (source code located in \$PLATFORMROOT/src/snmp).

The Agent can be built in standalone mode or as part of the RI.

In standalone mode the agent includes a main.c and links to the supportLibs build of the Net-SNMP libraries. To build the Agent in standalone mode, go to the \$PLATFORMROOT/src/snmp/ directory and do a 'make standalone'. The executable (platformAgent.exe) lands in \$PLATFORMROOT/install/Win32/debug/bin. When executed, the platform's configuration file must be specified on the command line:

```
> platformAgent
SNMP Interface Launcher. Usage: platformAgent <path to configuration file>
>
> platformAgent platform.cfg
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_InterfaceInit(udp, 10.36.16.157, 10161, 10705, false);
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread((nil)); for 10.36.16.157:10161
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_DEBUG handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_INFO handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_NOTICE handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_WARNING handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_ERR handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_CRIT handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_ALERT handler
20110413 16:22:22.005 INFO RI.NET.SNMP.Agent- snmp_AgentThread - installed the LOG_EMERG handler
....
```

The configuration file must include SNMP items such as:

```
# The IP address of the interface on THIS machine which the RIPlatform uses
RI.Platform.IpAddr = 10.36.16.157
# Boolean flag denoting Platform SNMP Agent type (true = sub false = master)
RI.Platform.SnmpIsSubAgent = FALSE
# The IP protocol on THIS machine which the SNMP Master Agent will use
RI.Platform.SnmpProto = udp
# The IP port on THIS machine which the SNMP Master Agent will use
RI.Platform.SnmpPort = 161
# The IP port on THIS machine which the SNMP AgentX protocol will use
RI.Platform.SnmpAgentXPort = 705
```

SNMP MIB Support

The platform should support the MIBs as described in the OC-STB-HOST-MIB-INN-YYMMDD.txt file.

This file is derived from the OpenCable HostDevice 2.X MIB specification (can be downloaded from the CableLabs doczone) and includes several other MIB dependencies. The \$PLATFORMROOT/src/snmp/ directory includes the latest MIB specification .txt file and an ./orig/ directory that includes the generated code for support of the required MIBs. The default make of this directory will take the generated MIB support code, patch it to build with the RI, and finally build and create a library (\$PLATFORMROOT/install/Win32/debug/bin/libplatformAgent.dll) for the RI to link against.

When a new MIB specification is released (or support for a different one is desired), the following instructions should be followed:

1. copy the new MIB .txt file into the \$PLATFORMROOT/src/snmp directory and remove the old one. (note the Makefile assumes the file begins with "OC-STB-HOST-MIB...") Sometimes the MIB changes go into one of the *.my files as necessary. make mibsourceupdate in step2 below will create one MIB txt file called temp-mib.txt under \$PLATFORMROOT/src/snmp which will hold ALL of the MIB objects which are held in different *.txt and *.my files.
2. make mibsourceupdate (if the output printed complains about not finding mib2c, go to the "Net-SNMP Applications" section, otherwise the ./orig/ directory has been updated and you can perform svn diff on that directory to observe the changes. A subsequent svn ci will be required to commit those changes).
3. make clean purge build (if there are no errors, you can test your SNMP support with a manager of your choice, otherwise goto "MIB(s) Added or Changed" section).
4. make mibtest (if the output printed matches your expectation of MIBs to support - you are done, otherwise go to "Changing Existing MIB Support" section).

MIB(s) Added or Changed

If there are one or more new or different MIBs you may need to add or edit a patch file for the handler of the respective MIB. If this is the case, the build yielded an error when compiling your handler file. You should edit the respective handler file, correct the error (including adding code to support the specific object(s) within the MIB). This edit would be followed by generating a patch file for the handler and placing it in the ./orig/ directory. An example command:

```
diff -Naur ./orig/myHandler.c myHandler.c > ./orig/myHandler.patch
```

Always have one patch per file (1 patch per .c or .h file).

This will allow future builds to use your new patch and apply it to the original (or freshly generated) MIB handler code.

Once this process is complete for all new or changed MIB(s) you should check-in the changes made. This may include edits or additions, so don't forget to perform "svn add" for each of your new files and patches. Always test with a "make clean purge build" afterwards to check that all is working.

Changing Existing MIB Support

There is a list of MIBS to ignore in the Makefile:

```
IGNORE_MIBS = \
    cableLabs \
    clabCommonMibs \
    clabFunction \
    clabProjCableHome \
    clabProjDocsis \
    clabProject \
    ...
```

This list can be modified to allow a specific MIB to be recognized and supported or for one to be removed. When this has been done the instructions above in section "MIB(s) Added or Changed" must be followed after the Makefile modification.

Net-SNMP Applications

To use the Net-SNMP applications with the OC-STB-HOST-MIB spec, the following MIBs must be in reach of the respective app:

CLAB-DEF-MIB.my
DOCS-IETF-BPI2-MIB.my
DOCS-IF-MIB.my
HOST-RESOURCES-MIB.my
IANAifType-MIB.my
IF-MIB.my
INET-ADDRESS-MIB.my
SNMP-FRAMEWORK-MIB.my
SNMPv2-CONF.my
SNMPv2-MIB.my
SNMPv2-SMI.my
SNMPv2-TC.my

To update the supported MIBs generated source code you must have the Net-SNMP applications "snmptranslate" and "mib2c".

If you don't have these applications, executing certain make targets like "mibsourceupdate" and "mibtest" will show the following results:

```
> make mibsourceupdate
*****
Could not find mib2c - can't compile mibs!
*****

> make mibtest
*****
Supported MIBs (from hard-coded list in Makefile!):
*****
mib
hrDeviceTable
hrProcessorTable
hrStorage
hrStorageTable
...
```

Note the "Could not find..." message - this means your system is not capable of compiling MIBs as mib2c is missing. Also note the "(from hard-coded..." message - this means your system is not capable of generating the list of mibs as snmptranslate is missing.

To install these applications, you may either download the Net-SNMP binary for your specific machine type and install it, or you can build them. If you choose to build them you will have to have the necessary understanding of your system (Linux, MinGW, etc) to use configure properly. The Net-SNMP tools must be installed in /usr/bin. (By default, the Windows binary installer for Net-SNMP places the tools in C:/usr/bin. So if your system doesn't have directories mounted accordingly, it won't work). Once installed, you can test your success with either of the two aforementioned commands and the results should look like this:

```

> make mibsourceupdate

*****
Updating supported MIBs (from OC-STB-HOST-MIB-I11-110204.txt):
*****

writing to hrDeviceTable.h
writing to hrDeviceTable.c
running indent on hrDeviceTable.c
running indent on hrDeviceTable.h
hrDeviceTable
writing to hrProcessorTable.h
writing to hrProcessorTable.c
running indent on hrProcessorTable.c
running indent on hrProcessorTable.h
hrProcessorTable
...

> make mibtest
*****
Supported MIBs (from OC-STB-HOST-MIB-I11-110204.txt):
*****

mib
hrDeviceTable
hrProcessorTable
hrStorage
hrStorageTable
hrSWRunPerfTable
...

```

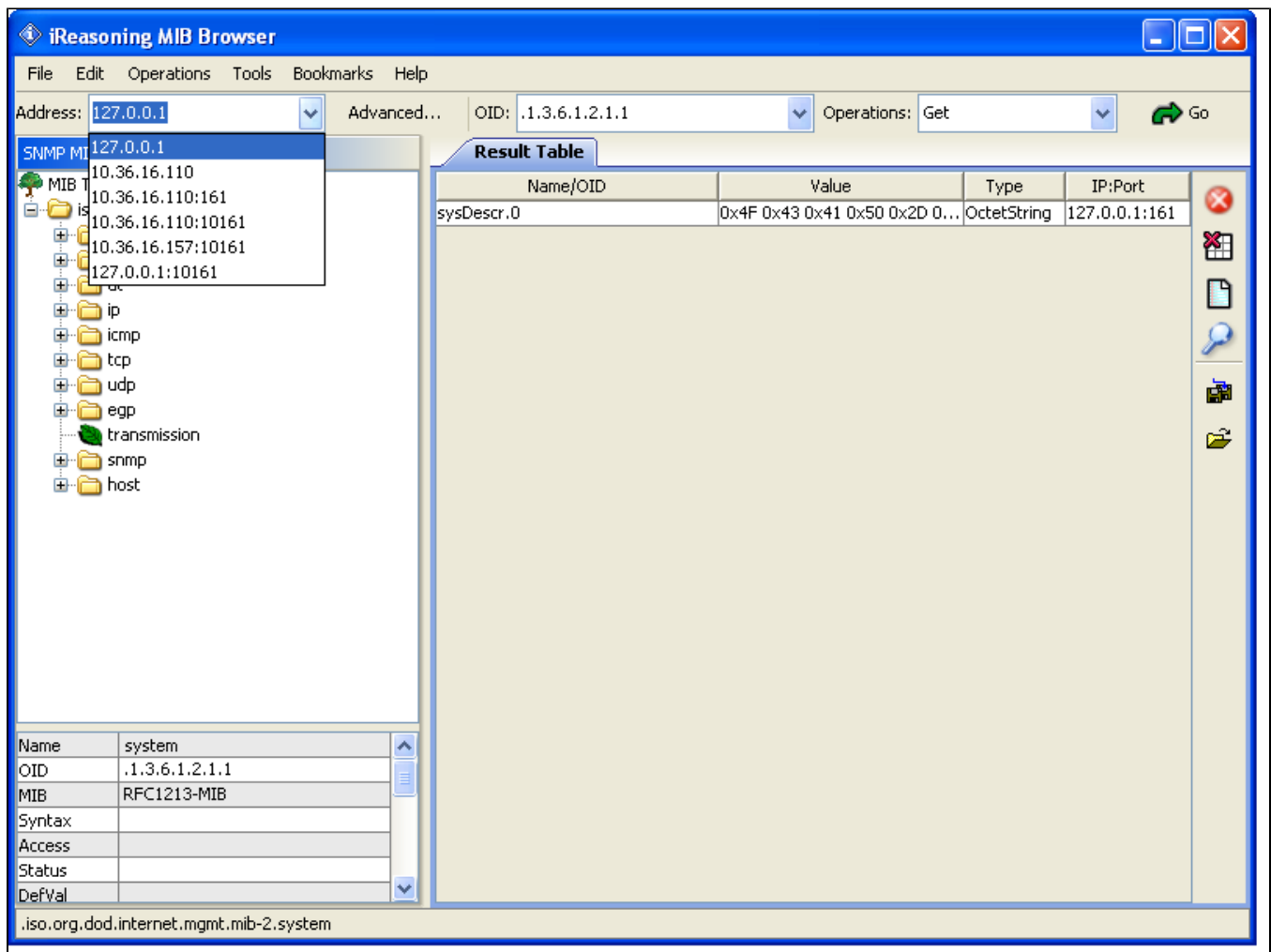
SNMP Manager

You'll need an SNMP manager to test SNMP support.

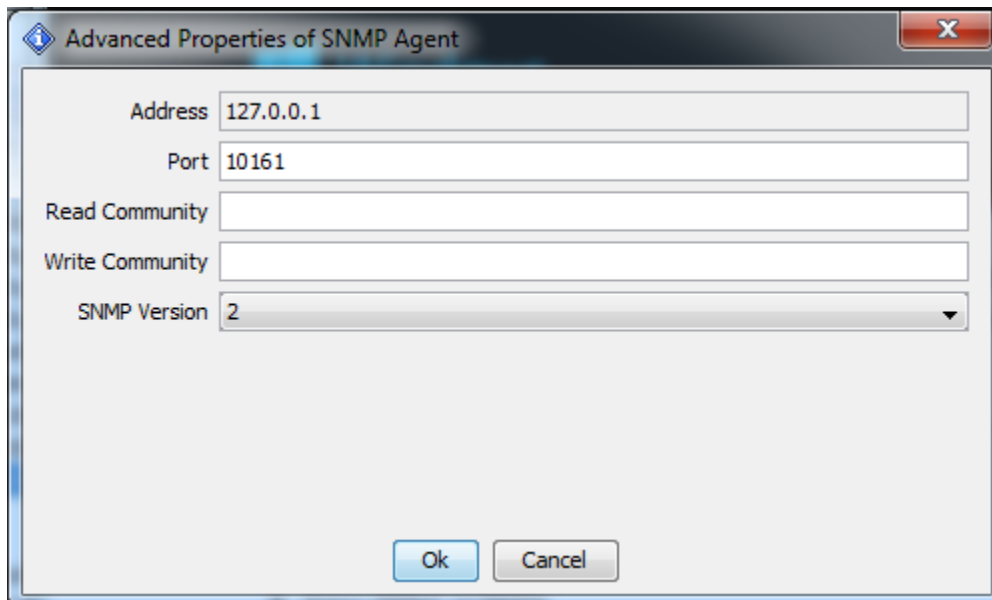
iReasoning (<http://www.ireasoning.com>) is a tool we use.

Once installed and running, you can set the IP of the device to connect to (if you are running standalone, your address may be localhost).

.



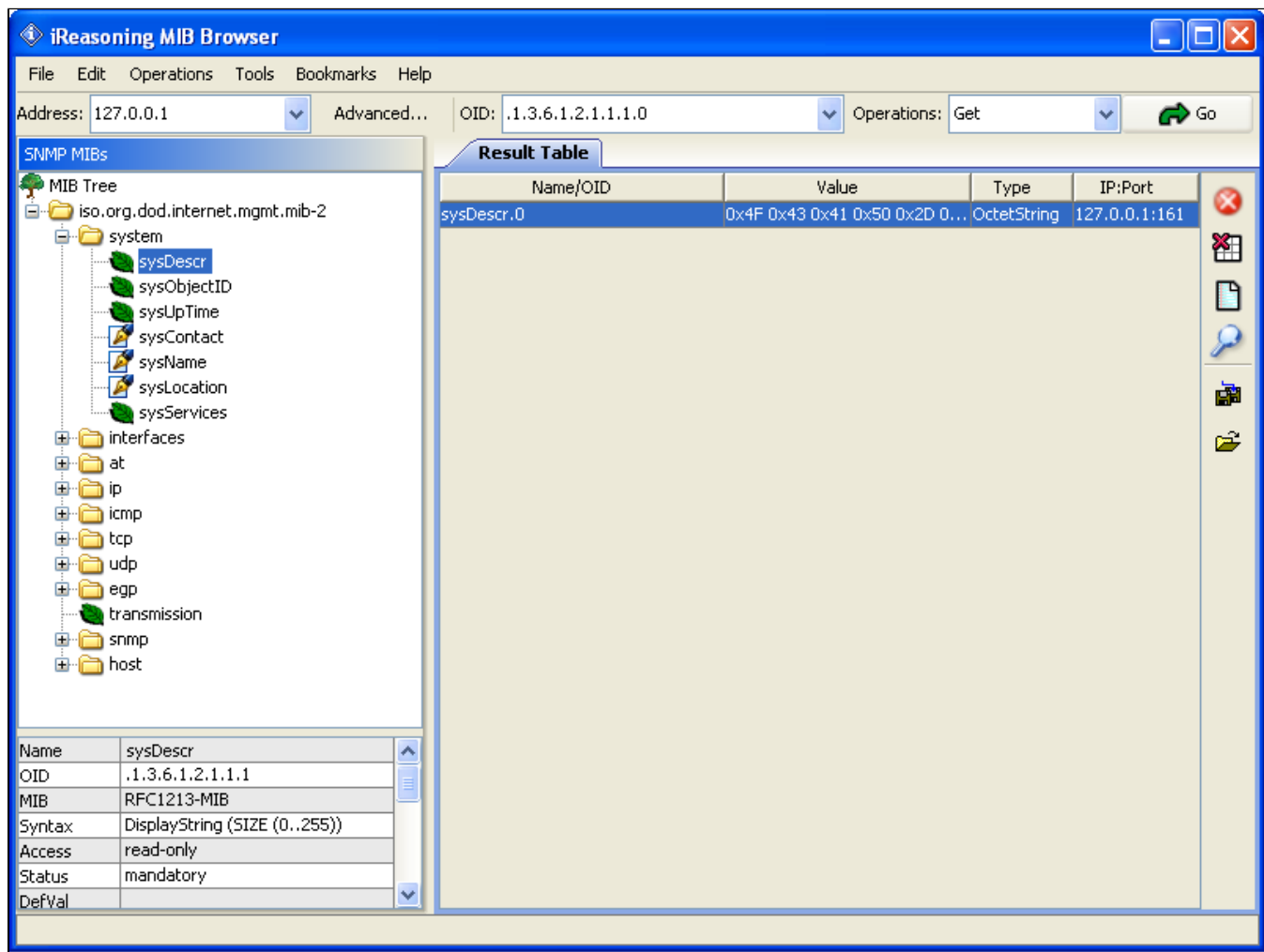
You will need to select "Advanced" and set the Port to 10161 and the SNMP Version to 2 when communicating with the RI (this will have to be done each time you enter a new IP address).



As an example, select the "system" folder under the MIB tree and subsequently select "sysDescr".

Then select the "Get" operation from the drop-down menu near the upper right hand corner.

Finally select "Go" in the upper right hand corner to get the value.



Validating a specific OCAP MIB OID using iReasoning MIB Browser - OCVET 75 example

Copy the MIB files from the RI platform to the iReasoning MIB directory.

go to: \$PLATFORMROOT/install/Win32/debug/share/snmp/mibs. Copy all the .my files.

go to C:/Program Files (x86) /ireasoning/mibbrowser/mibs . Paste in all the OCAP .my files.

go to: \$PLATFORMROOT/src/snmp/. Copy OC-STB-HOST-MIB-INN-YYMMDD.txt.

go to C:/Program Files (x86) /ireasoning/mibbrowser/mibs . Paste OC-STB-HOST-MIB-INN-YYMMDD.txt.

Start the RI.

The RI SNMP agent is started automatically by the runRI.sh script.

Use default values in platform.cfg and mpeenv.ini.

Start the iReasoning MIB browser

Under the "File" drop down menu, load in the MIB named OC-STB-HOST-MIB-114-120531. It was copied into iReasoning's MIB directory in an earlier step.

Find the OID to be tested. For this test it was a table. Traverse to the OID and double click it.

