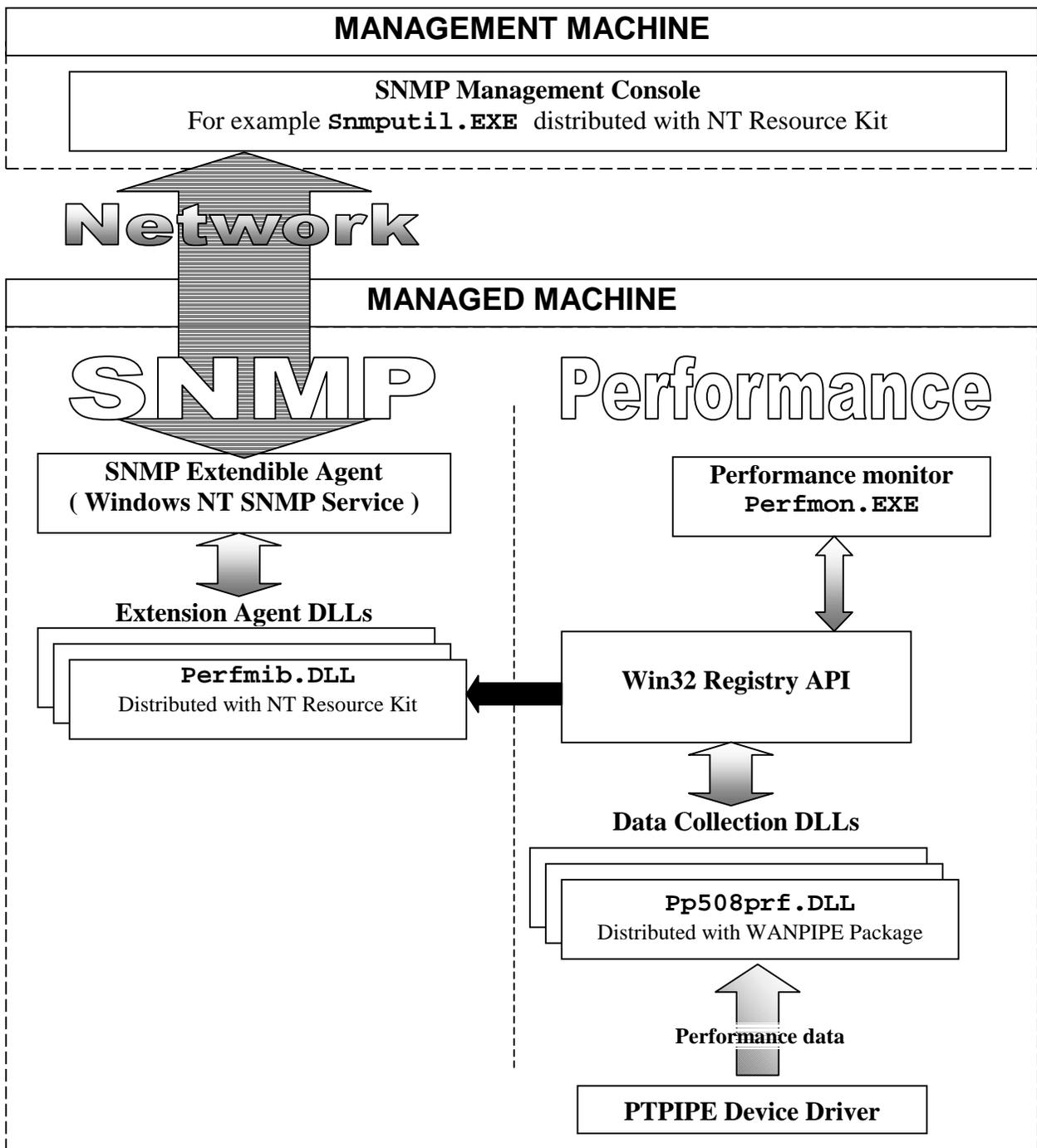


Monitoring PPIPE performance counters with SNMP

The `/snmp` directory of PPIPE distribution contains several files that can be used to monitor PPIPE performance counters via SNMP.

The Windows NT Resource Kit 4.0 includes a special SNMP Extension Agent DLL that communicates with Win32 Registry API in order to fetch performance counters values and makes them accessible via SNMP queries.

The following diagram shows the data flow between SNMP Management Console and a machine of interest, i.e. the one that has PPIPE device installed:



The performance counters are made accessible to SNMP by **Perfmib.DLL** supplied with Windows NT Resource Kit 4.0. The counters are located in the following SNMP subtree under **.iso.org.dod.**

```
internet.private.enterprises. :
    311 microsoft
        1 software
            1 systems
                3 os
                    1 winnt
                        1 performance
```

The numeric ID of the MIB branch for PTPIPE device is **514** however this can be any number that does not conflict with other monitored performance counters. The SNMP prefix for the PTPIPE counter names is **pp** however it is not mandatory either. The files located in **/snmp** directory were created with above values – you are free to change them, but please make sure you understand the changes you are willing to make.

Setting up the managed machine

There are two things that you have to do in order to expose performance counters via SNMP:

- Register **Perfmib.DLL** as an SNMP extension agent.
- Show to **Perfmib.DLL** the correspondence between SNMP OIDs and specific performance counters.

The following has to be done on the managed machine in order to accomplish the above tasks:

1. Stop SNMP service by executing **net stop snmp** from command prompt.
2. Copy **Perfmib.DLL** into system directory :
copy %ntreskit%\perfmib.dll %SYSTEMROOT%\System32
3. Register **Perfmib.DLL** with Windows NT SNMP service as SNMP extension agent. To accomplish that create the following registry entries :

```
\Registry\Machine
    Software
        Microsoft
            PerformanceAgent
                CurrentVersion
                    Pathname = REG_EXPAND_SZ %SystemRoot%\System32\perfmib.dll
    System
        CurrentControlSet
            Services
                SNMP
                    Parameters
                        ExtensionAgents
                            < Pick the next ordinal number > = REG_SZ
                            SOFTWARE\Microsoft\PerformanceAgent\CurrentVersion
```

4. Create **Perfmib.ini** file in the system directory. This file shows to **Perfmib.dll** how to map SNMP OIDs to actual performance counters. This file contains information about *all* performance

counters on your system that are monitored via SNMP. The **Perfmib.ini** file located in **/snmp** directory on PTPIPE distribution contains information about PTPIPE counters only. Please take a look on this file – the syntax is straightforward. Note that if you already monitor other performance counters via SNMP you will have to add the contents of **/snmp/Perfmib.ini** to your current **%SYSTEMROOT%\System32\Perfmib.ini** file.

5. Start SNMP service by executing **net start snmp** from command prompt.

Done! From now on you can query PTPIPE performance counters using any SNMP Management tool that is available to you. For example here is how you can use **Snmputil.exe** from NT Resource Kit in order to get PTPIPE ‘Up Time’ performance counter value:

```
snmputil get YourServerAddressOrName public .iso.org.dod. <continued>
internet.private.enterprises.311.1.1.3.1.1.514.1.35.0
```

As you have probably already noticed the counter value in this example is accessed using numeric OID. It is important to mention that the resolution of SNMP symbolic OIDs (such as **.iso.org.dod.internet.private.enterprises.microsoft.software.systems.os.winnt.performance**) is always done on the *client*, i.e. manager machine, side. The way you set up your management console depends on what console do you use and should be explained in the console documentation.

However, to get things going this is how you set up the **Snmputil.exe** from NT Resource Kit:

Setting up **Snmputil.exe** on the management machine

The **Snmputil.exe** uses file called **mib.bin** located in the system directory to map SNMP symbolic OIDs to its numeric equivalents. The special MIB compiler **mibcc.exe** from NT Resource Kit is used to create this file from **.mib** files that contain descriptions of various SNMP MIBs.

The **Ppipe.mib** file located in **/snmp** directory on PTPIPE distribution contains ASN.1 syntax MIBs for PTPIPE device performance counters. This file was automatically generated by **Perf2mib.exe** tool from NT Resource Kit. The following is a command line that generated this and **Perfmib.ini** file (please keep in mind that PTPIPE MIB branch ID – **514**, and the prefix – **pp** are arbitrary):

```
perf2mib Ppipe.mib Perfmib.ini Ptpipe 514 pp
```

The following steps have to be performed on the management machine in order to use **Snmputil.exe** with symbolic OIDs:

1. Assuming that your current directory is **/snmp** at the distribution floppy or whatever directory you’ve copied the files to, the following command line compiles PTPIPE MIBs together with the standard set of other Windows NT MIBs into **mib.bin** file which will be used by **Snmputil.exe** for symbolic OIDs resolution:

```
mibcc -o%ntreskit%\mib.bin -n -t -w2 %ntreskit%\smi.mib <continued>
%ntreskit%\LMMIB2.MIB %ntreskit%\mib_II.mib Ppipe.mib
```

Assuming you run it in Windows NT command prompt you will see detailed trace of compiler actions as well as the summary of created MIB branches. If you are using the same numeric ID and the prefix as in supplied MIB file the following branch will be created for PTPIPE device:

```
311 microsoft
      1 software
```

```

1  systems
  3  os
    1  winnt
      1  performance
        514 ppptpipeTable
          1 ppptpipeEntry
            1 ppptpipeIndex
            2 ppptpipeInstance
            3 ppPPPState
            4 ppDCDState
            5 ppCTSState
            6 ppBytesTransmitted
            7 ppBytesReceived
            8 ppFramesTransmitted
            9 ppFramesReceived
           10 ppRxdFramesBadLength
           11 ppRxOverrunErrors
           12 ppRxCRCErrors
           13 ppAbortFramesReceived
           14 ppReceiverDisabledBuffers
           15 ppAbortFramesTransmitted
           16 ppMissedTxUnderrunInts
           17 ppDCDChanges
           18 ppCTSChanges
           19 ppPacketsDiscardedHeader
           20 ppPacketsDiscardedProtocol
           21 ppLCPpacketsRx
           22 ppLCPpacketsTx
           23 ppIPCPpacketsRx
           24 ppIPCPpacketsTx
           25 ppIPXCPpacketsRx
           26 ppIPXCPpacketsTx
           27 ppPAPpacketsRx
           28 ppPAPpacketsTx
           29 ppCHAPpacketsRx
           30 ppCHAPpacketsTx
           31 ppIPPacketsRx
           32 ppIPPacketsTx
           33 ppIPXpacketsRx
           34 ppIPXpacketsTx
           35 ppUpTime

```

2. Save your current **mib.bin** file :
copy %SYSTEMROOT%\System32\mib.bin %SYSTEMROOT%\System32\mib.old
3. Copy a new **mib.bin** file into system directory :
copy %ntreskit%\mib.bin %SYSTEMROOT%\System32

Done! Now instead of using numbers for SNMP queries you can use the much more meaningful names. For example here is the exact same query as on previous page with symbolic OIDs instead of numeric. It will return the 'Up Time' performance counter for PTPIPE device installed on the managed machine:

```

snmputil get YourServerAddressOrName public .iso.org.dod.      <continued>
internet.private.enterprises.microsoft.software.systems.os.  <continued>
winnt.performance.ppptpipeTable.ppptpipeEntry.ppUpTime.0

```

Note that **Snmputil.exe** was designed as a very basic SNMP management tool that is very good for debugging SNMP problems, since it basically allows you to “talk” SNMP directly. However, if you are seriously thinking of managing your network with SNMP you should probably get some of the nicer SNMP consoles such as IBM NetView® or HP OpenView®.