



Synapse

Ganglia Native Windows Node Agent

Steve Flasby

steve.flasby@aprconsulting.ch

Markus Koller

markus.koller@aprconsulting.ch

Version 0.3, 1 September 2006

This document contains an overview of the APR Consulting port of the Ganglia Node Agent (*gmond*) to Microsoft Windows.

Table of Contents

1. Introduction.....	1
1.1 Purpose of document.....	1
1.2 Structure of document.....	1
1.3 Caveat.....	1
1.4 Glossary of terms.....	1
1.5 Copyright.....	1
1.6 History.....	2
2. Installation Guide.....	3
2.1 Supported Operating Systems.....	3
2.2 Additional Software Requirements.....	3
2.3 Installation.....	4
2.3.1 Service Name.....	4
2.3.2 Installing.....	4
2.3.3 Starting the service.....	4
2.3.4 Stopping the Service.....	4
2.3.5 Removing the Service.....	4
2.3.6 User.....	5
3. Native Implementation.....	6
3.1 Libraries.....	6
3.2 Logging.....	6
3.3 Metrics.....	6
4. Changes from the Open Source Version.....	9
4.1 Accessing the Native Windows API.....	9
4.2 Future Changes.....	9

1. Introduction

The Ganglia open source network monitoring tool is focused on Linux and Unix operating systems. There is a need to provide a native Windows version. APR Consulting has implemented the Ganglia node agent as a native Windows service.

1.1 Purpose of document

This document describes the APR Consulting port of the Ganglia node agent to Microsoft Windows. It provides an installation guide and an overview of the port.

1.2 Structure of document

Section 1, Introduction

Section 2, Installation Guide How to install and remove Synapse.

Section 4,

1.3 Caveat

This is a beta release.

Development is still in progress with further releases anticipated. This document hints at future changes. If the reader has any hard requirements, then please let us know and we will consider them for the next release.

1.4 Glossary of terms

<i>Term</i>	<i>Definition</i>
Ganglia	Ganglia is a distributed monitoring system for clusters of computers. See http://ganglia.sourceforge.net/ for more details.
gmond	A daemon which gathers metrics from a single computing node for distribution to 'gmetad'.
gmetad	A daemon which gathers metrics from a cluster of 'gmond' agents and/or other 'gmetad' daemons.

1.5 Copyright

This document is copyright ©, APR Consulting 2006, all rights reserved.

1.6 History

Version	Changes
First draft	
Second draft	Updates Glossary. Added Ganglia introduction.

2. Installation Guide

Synapse is distributed as a simple executable, together with supporting libraries, which installs as a Windows Service.

2.1 Supported Operating Systems

Synapse is intended to run on:

- Windows 2000 Professional
- Windows XP Professional
- Windows 2000 Server
- Windows 2003 Server

It is not intended to run on any of the Windows 9x releases nor on Windows NT.

2.2 Additional Software Requirements

Synapse is shipped with a small number of additional libraries. These are either standard Microsoft C runtime libraries or open source libraries for which the source can be supplied. No other special software installation is required.

Note that Synapse does not require the installation of the '.NET' framework.

2.3 Installation

The service is installed by unpacking the supplied ZIP file and then running the executable *gmondservice-3.0.3.exe*. The ZIP should not be unpacked and run on a windows user desktop directory as this has specific permissions which will prevent the service from starting.

Unpacking into *c:\ganglia-3.0.3* is the default location.

The service may be installed and removed using

```
gmondservice-3.0.3.exe -install <path to config>
gmondservice-3.0.3.exe -remove
```

which installs or removes the service, respectively.

2.3.1 Service Name

The service is installed as '**APR GMOND 3.0.3**'

This name is subject to possible change.

2.3.2 Installing

A single executable is used to install and remove the service. To install run:

```
gmondservice-3.0.3.exe -install c:\ganglia-3.0.3\gmond.conf
```

The *install* command expects a second parameter to tell the service where to find its configuration file. The configuration file '*gmond.conf*' is a standard Ganglia configuration file.

A pair of simple batch files are provided to install and remove the service if it is the standard location.

2.3.3 Starting the service

The service does not start automatically after installation. Either use the services control panel snapin to start the service or the command line command:

```
net start "APR GMOND 3.0.3"
```

can also be used.

Note that the service will start if the machine is restarted.

2.3.4 Stopping the Service

The service can be stopped using either the service control panel snapin or via the command line:

```
net stop "APR GMOND 3.0.3"
```

2.3.5 Removing the Service

```
gmondservice-3.0.3.exe -remove
```

will stop and then remove the service.

2.3.6 User

Like most services, the service runs as the standard Local System account.

3. Native Implementation

The open source Ganglia distribution relies on Cygwin to provide a Unix-like execution environment. The APR port relies only on native windows libraries plus a small number of other open source libraries. These are all provided with the Synapse distribution so no additional installations are required.

3.1 Libraries

The following libraries are distributed with Synapse. They are simply deployed into the Synapse installation directory, no other installation is required.

libapr The Apache Portable Runtime

libConfuse Parsing library

oncrpc An open source implementation of the ONC Remote Procedure Call API.

pthreadVC2 An open source pthreads implementation.

msvcXX Standard Microsoft runtime libraries.

3.2 Logging

Synapse logs into the Windows Application event log.

The original code provides very little information about internal events. This will be enhanced to provide data about the operational state of the service and any error conditions encountered. Currently, the service writes to the event log when errors are encountered in the metrics collection process and when the service starts up and shuts down.

3.3 Metrics

Synapse collects practically a full set of metrics.

The metrics collection processing is still being developed as this document is being written. If some of the non-supported metrics are required they will be implemented with a higher priority. The intention is to provide all metrics which can be obtained at a reasonable cost in run-time.

The following metrics are provided:

- **boottime**
System boot time stamp.
- **bytes_in**
Number of bytes in, per second.
- **bytes_out**
Number of bytes out, per second.
- **cpu_idle**
Percentage of time spent in idle.
- **cpu_aidle**
Percentage of time since boot spent in idle.

- **cpu_system**
Percentage of time spent in system space.
- **cpu_user**
Percentage of time spent in user space.
- **cpu_num**
Number of CPUs. This is the physical number of CPUs. A dual core processor will list as 2 CPUs. A single hyperthreading CPU will list as one CPU, not two.
- **cpu_speed**
The CPU clock speed as reported by the Windows API.
- **disk_free**
The sum of all free space on all disks on this machine.
- **disk_total**
The sum of all the disk space on this machine.
- **part_max_used**
The free space of the disk partition with highest percentage of used space.
- **os_release**
As returned by the Windows API.
- **proc_run**
Number of running processes.
- **proc_total**
Total number of processes.
- **swap_free**
Free space in the allocated swap file.
- **swap_total**
Size of the swap file.
- **sys_clock**
Current host time.
- **pkts_in**
Number of network packets received.
- **pkts_out**
Number of network packets sent.
- **bytes_in**
Number of bytes received via the network.
- **bytes_out**
Number of bytes sent via the network.

The following metrics are not provided:

- **cpu_wio - Waiting for I/O**
This is not available directly from the windows monitoring API so would require constant polling to maintain. We think the overhead of collecting this data would outweigh any benefit of providing this metric.

The following metrics are still in work but we intend to provide them unless implementation proves unacceptably expensive:

- **cpu_nice**
Number of processes with their process priority below normal.
This metric will be provided in a later release.
- **location**
Supposedly supported for all implementations but not yet verified.

- **load_one**
As none of the metrics are directly available from the Windows APIs, implementing this will require frequent polling and as such will be expensive. This will be implemented at a much later stage and will never be as accurate as the native implementations provided by typical Unix operating systems.
- **load_five**
idem.
- **load_fifteen**
idem.

The following metrics are Solaris specific. At this point they are not supported.

- **bread_sec**
Solaris specific, not supported.
- **bwrite_sec**
Solaris specific, not supported.
- **lread_sec**
Solaris specific, not supported.
- **lwrite_sec**
Solaris specific, not supported.
- **phread_sec**
Solaris specific, not supported.
- **phwrite_sec**
Solaris specific, not supported.
- **rcache**
Solaris specific, not supported.
- **wcache**
Solaris specific, not supported.

The following metrics are Linux specific and are not supported.

- **mem_buffers**
Linux only
- **mem_cached**
Linux only

The following metrics are MS Windows specific.

- **phys_disk_bytes_sec**
Is the rate bytes are transferred to or from the disk during write or read operations.
- **phys_disk_time**
is the percentage of elapsed time that all disks drive were busy servicing read or write requests.
- **sys_cpu_queue_len**
is the number of threads in the processor queue.
- **mem_pages_sec**
is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause system-wide delays.
- **mem_committed_bytes**
is the amount of committed virtual memory, in bytes.

4. Changes from the Open Source Version

To run as a service a number of changes have been made.

- Removal of any dependencies on the CygWin libraries
- Addition of new libraries to replace a small number of external dependencies.
- Modification of source code

The modified source code is isolated and easy to maintain. A small number of changes in code outside the core '*gmetad*' sources have been made because of issues with the Microsoft compilers. These changes will be provided back to the Ganglia source tree where possible.

The main change is the way which the metrics are gathered. This module is completely rewritten to take advantage of the native windows API.

4.1 *Accessing the Native Windows API*

The Microsoft Windows API is a C++ layer. The metrics code is all written in C++ behind a C façade.

4.2 *Future Changes*

We believe that some of the libraries required by the node agent are very sparsely used. These libraries will be identified and, if possible, minimal ports of the required functions implemented. This will reduce the footprint of the agent still further.