

Overview

Nagios Enterprises tested Western Telematic Inc's WTI-MPC-16H managed power controller to ascertain the ease and availability of integration with Nagios. This document summarizes the steps we took to successfully integrate the MPC unit with Nagios.

The MPC unit was successfully monitored with Nagios using SNMP. We were able to monitor the following metrics:

- Unit Temperature
- Bus A Current
- Bus A Power
- Bus A Voltage
- Bus B Current
- Bus B Power
- Bus B Voltage

Components Used In Testing

WTI Managed Power Controller

Model: WTI-MPC-16H
Firmware: 1.4.0

Monitoring Server

Operating System: Fedora Core 8
Nagios Version: 3.0.1
Nagios Plugins Ver.: 1.4.11
NET-SNMP Version: 5.4.1

Initial Setup

MPC Unit:

1. The WTI-MPC-16H was installed on our network and given an IP address of 192.168.1.13
2. SNMP was enabled and configured on the device as follows:

SNMP Version: V1/V2 Only
SNMP Community: public
3. Various electronic components were plugged into Bus A of the unit and powered on. Bus B was left unused in our testing.



Monitoring Server:

1. NET-SNMP version 5.4.1 was installed on the monitoring server with the following commands:

```
yum install net-snmp  
yum install net-snmp-utils
```

2. Nagios 3.0.1 was compiled and installed on the monitoring server
3. Version 1.4.11 of the Nagios plugins were installed on the monitoring server.

NOTE: Make sure to compile the plugins after you install NET-SNMP, as this will ensure that the check_snmp plugin gets built and installed.

4. The WTI-MPC MIBS were installed on the system by copying the WTI-MPC-MIB.txt file from the CD-ROM (supplied with the MPC unit) to the /usr/share/snmp/mibs directory of the monitoring server.

Monitoring Configuration

1. We modified our Nagios configuration files to include a new host definition for the MPC unit:

```
define host{
    use                generic-host
    host_name          mpc
    alias              MPC-16-H
    address            192.168.1.13
    parents            linksys-srw224p
}
```

The *generic-host* host template in our Nagios configuration looked like this:

```
define host{
    name                generic-host
    notifications_enabled 1
    event_handler_enabled 1
    flap_detection_enabled 1
    failure_prediction_enabled 1
    process_perf_data 1
    retain_status_information 1
    retain_nonstatus_information 1
    notification_period 24x7
    register            0
    max_check_attempts 3
    check_interval      5
    retry_interval      1
    check_command        check-host-alive
    contact_groups      admins
    notification_interval 120
    notification_options d,u,r,f,s
}
```

2. Next we modified our Nagios configuration and added service definitions for monitoring the MPC unit:

```

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Temperature
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:80 -c 0:100 -l
    "Temperature"-u "Deg F" -o WTI-MPC-
    MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitTemperature.1
    check_interval     5
}

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Bus A Current
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:5 -c 0:10 -l "Current"
    -u "Amps" -o WTI-MPC-
    MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitCurrentA.1
    check_interval     5
}

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Bus B Current
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:5 -c 0:10 -l "Current"
    -u "Amps" -o WTI-MPC-
    MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitCurrentB.1
    check_interval     5
}

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Bus A Voltage
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:125 -c 0:130 -l
    "Voltage" -u "Volts" -o WTI-MPC-
    MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitVoltageA.1
    check_interval     5
}

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Bus B Voltage
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:125 -c 0:130 -l
    "Voltage" -u "Volts" -o WTI-MPC-
    -MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitVoltageB.1
    check_interval     5
}

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Bus A Power
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:500 -l "Power" -u
    "watts" -o WTI-MPC-MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitPowerA.1
    check_interval     5
}

define service{
    use                generic-service
    host_name          mpc
    service_description Unit 1 Bus B Power
    check_command      check_snmp!-P 2c -C public -m WTI-MPC-MIB -w 0:500 -l "Power" -u
    "watts" -o WTI-MPC-MIB::environmentTables.environmentUnitTable.environmentUnitEntry.environmentUnitPowerB.1
    check_interval     5
}

```

The check commands in the service definitions we added used the *check_snmp* plugin to query the MPC unit using SNMP. The *check_snmp* command definition in our Nagios configuration looked like this:

```

define command{
    command_name      check_snmp
    command_line      $USER1$/check_snmp -H $HOSTADDRESS$ $ARG1$
}

```

The *generic-service* service template in our Nagios configuration looked like this:

```
define service{
    name generic-service
    active_checks_enabled 1
    passive_checks_enabled 1
    parallelize_check 1
    obsess_over_service 1
    check_freshness 0
    notifications_enabled 1
    event_handler_enabled 1
    flap_detection_enabled 1
    failure_prediction_enabled 1
    process_perf_data 1
    retain_status_information 1
    retain_nonstatus_information 1
    is_volatile 0
    check_period 24x7
    max_check_attempts 3
    normal_check_interval 5
    retry_check_interval 2
    contact_groups admins
    notification_options w,u,c,r,f,s
    notification_interval 60
    notification_period 24x7
    register 0
}
```

3. After saving our configuration files, we verified our configuration files and started Nagios:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

4. Once Nagios was restarted, we could view the status of the MPC unit in the Nagios web interface:

Service Status Details For Host 'mpc'

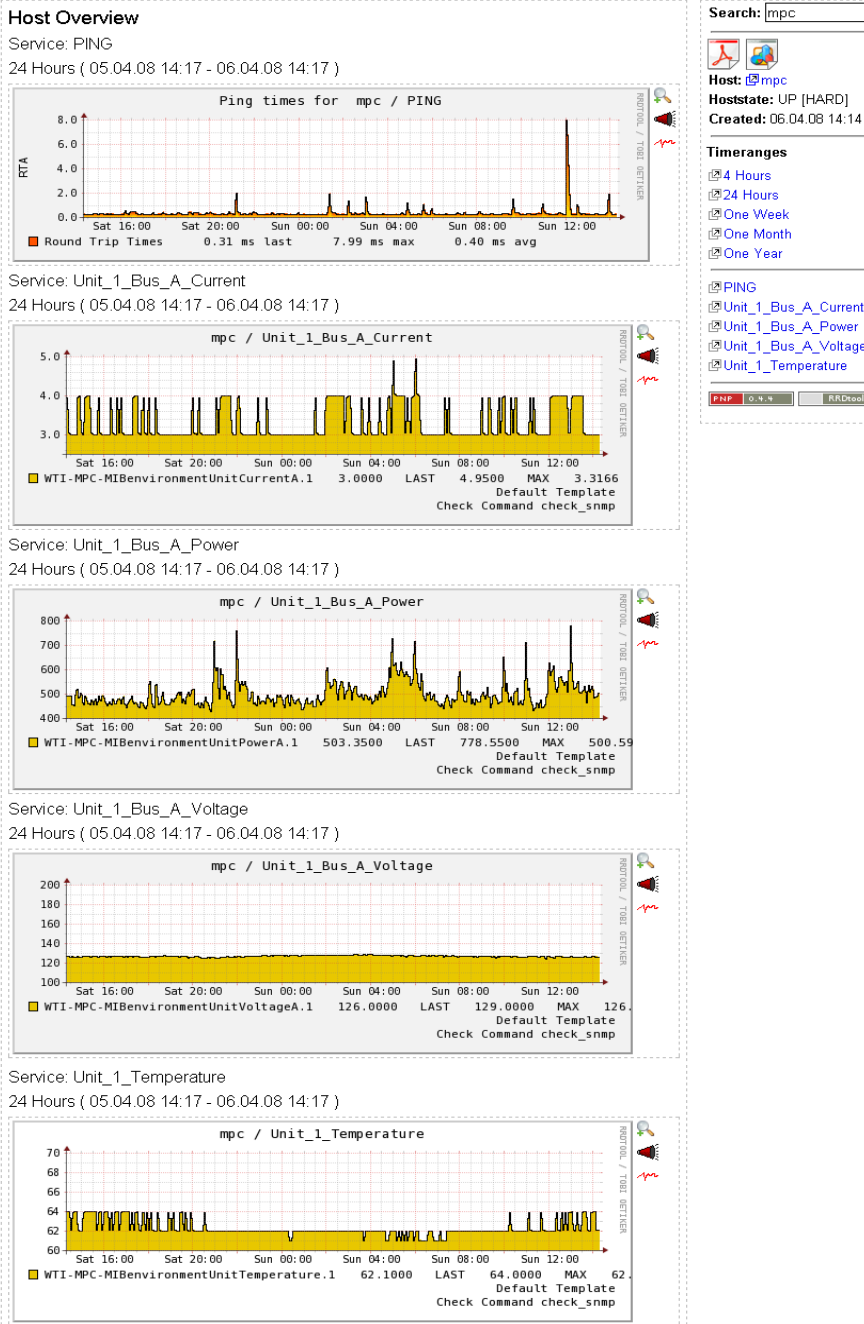
Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
mpc	PING	OK	04-29-2008 11:35:13	26d 19h 1m 47s	1/3	OK - 192.168.1.13: rta 1.284ms, lost 0%
	Unit 1 Bus A Current	OK	04-29-2008 11:36:29	24d 2h 23m 25s	1/3	Current OK - 2 Amps
	Unit 1 Bus A Power	OK	04-29-2008 11:32:46	9d 20h 42m 9s	1/3	Power OK - 281 Watts
	Unit 1 Bus A Voltage	WARNING	04-29-2008 11:33:02	0d 2h 13m 15s	3/3	Voltage WARNING - *128* Volts
	Unit 1 Bus B Current	OK	04-29-2008 11:35:18	24d 2h 19m 37s	1/3	Current OK - 0 Amps
	Unit 1 Bus B Power	OK	04-29-2008 11:36:34	24d 2h 18m 21s	1/3	Power OK - 0 Watts
	Unit 1 Bus B Voltage	OK	04-29-2008 11:32:50	24d 2h 17m 5s	1/3	Voltage OK - 0 Volts
	Unit 1 Temperature	OK	04-29-2008 11:34:06	24d 2h 15m 49s	1/3	Temperature OK - 62 Deg F

We had to wait 5 minutes for Nagios to check all the services on the MPC before complete status information was available in the Nagios web interface.

Graphing

The following graphs were produced by PNP(<http://www.pnp4nagios.org/>) - a graphing addon tool for Nagios. They show historical values for unit temperature, as well as current, voltage, and power readings on Bus A of the MPC unit. Graph data was obtained directly from the performance data returned from the `check_snmp` plugin used to monitor metrics on the MPC unit.

No additional Nagios configuration was required to graph metrics from the MPC unit using PNP. However, PNP must be installed and configured separately from Nagios. Instructions for installing PNP are out of the scope of this document.



Company Contact Information

For more information on monitoring Western Telematics' MPC units with Nagios, contact the following companies:

Nagios Enterprises, LLC

P.O. Box 8154
Saint Paul, MN 55108

Email: inquiries@nagios.com
Web: www.nagios.com
US: (888) 624-4671
Int'l: +1 (651) 204-9102

Western Telematic Inc.

5 Sterling
Irvine, CA 92618-2517

Web: www.wti.com
US: (800) 854-7226