



NerveCenter 6.1 New Functions at a Glance

White Paper

February 2014

NCRN61-EP-01

Copyright

Portions Copyright ©1989-2014 LogMatrix, Inc. / OpenService, Inc. All rights reserved.

Disclaimers

LogMatrix, Inc. ("LogMatrix") makes no representations or warranties, either expressed or implied, by or with respect to anything in this manual, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

These applications are available through separate, individual licenses. Not every feature or application described herein is licensed to every customer. Please contact LogMatrix if you have licensing questions.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of LogMatrix. While every precaution has been taken in the preparation of this book, LogMatrix assumes no responsibility for errors or omissions. This publication and the features described herein are subject to change without notice.

The program and information contained herein are licensed only pursuant to a license agreement that contains use, reverse engineering, disclosure and other restrictions.

Trademarks

LogMatrix is registered in the U.S. Patent and Trademark Office. NerveCenter and the LogMatrix Logo are trademarks of LogMatrix, Inc.

All other products or services mentioned in this manual may be covered by the trademarks, service marks, or product names as designated by the companies who market those products.

Contacting LogMatrix

LogMatrix, Inc.
2 Mount Royal Ave, Suite 250
Marlborough, MA 01752

Phone 508-597-5300
Fax 774-348-4953

contact email: info@logmatrix.com

Website: www.logmatrix.com

Forum: <http://community.logmatrix.com/LogMatrix>

Blog: www.logmatrix.com/blog

NerveCenter 6.1 Enhanced Polling

Polling Multiple Objects "On-the-fly" in a Single Poll

Simple Polling Doesn't Give the Whole Story. Polling for single objects and then applying a simple threshold or basic if/then logic to the resulting data is something many network monitoring tools can do. However, this doesn't provide the complete picture of the true condition (or "state") of the devices or services being managed. For network operators to more effectively manage their own unique network environment, data from multiple SNMP objects (or other management protocols) is required for the complete story. In addition, the data required from multiple sources needs to be collected in real time and kept in a "state-ful" way to be useful.

NerveCenter's "Cross Table Polling" Can Help.

NerveCenter's unique finite-state architecture can help, especially with the new enhanced Cross Table Polling feature that allows for multiple objects to be queried in a single poll. As always, NerveCenter makes it easy to do custom designed polls and logic built to your specific business needs.

As a simple example:

- Via a poll or trap, an operator receives a notice that a link is down.
- To manage the network effectively, the operator must next quickly determine the severity of the condition (Is this an active link? Is it important to a customer?)
- With NerveCenter, an operator can poll basic interface data to determine if a link is down and then in that same poll look at the extended link table to get a description of the use or customer associated with that link.
- Leveraging NerveCenter's unique finite-state architecture, this can all be done "on-the-fly" - i.e. not via a database query after the fact.

Cross table polling info

Polls are now definable to allow more than one Base Object. NC6.1 is the first release to implement the multi-base object polling. NC6.1 provides multi-table-polling, passing "joins" of separate Base Object polls together as *logical rows* into the user's OnResponse function.

- Ex1: Scalar groups, such as "system" and "interfaces" can be used in the same poll. The user's OnResponse poll function is called *once* when the two queries receive back the data.
 - my \$name = system.sysName;
 - my \$uptime = system.sysUptime;
 - my \$numInterfaces = interfaces.ifNumber;
 - FireTrigger("BasicSysInfo");

- Ex2: Data can be gathered from multiple tables wherein the indexing for each table is identical. For example: ifTable and ifXTable. The OnResponse function is called once as each logical row is retrieved. If a device has, say, 30 rows in ifTable/ifXTable, then OnResponse is called once for each (30 times total).
 - my \$index = ifEntry.ifIndex;
 - my \$name = ifXEntry.ifName;
 - my \$inOctets = ifEntry.ifInOctets;
 - my \$outOctets = ifEntry.ifOutOctets;
 - my \$inFastOctets = ifXEntry.ifHCInOctets;
 - my \$outFastOctets = ifXEntry.ifHCOctets;
 - FireTrigger("interface-row");

- Ex3: Scalar groups ("system", "interfaces", etc.) can be mixed with a table lookup. The OnResponse function is called once as each logical row is retrieved. If the device has, say, 30 rows in ifTable/ifXTable, then OnResponse is called once for each (30 times total). The scalar value (ex: "system.sysUpTime") are repeated for each such call to the OnResponse function.
 - my \$name = system.sysName;
 - my \$uptime = system.sysUptime;
 - my \$numInterfaces = interfaces.ifNumber;
 - my \$index = ifEntry.ifIndex;
 - my \$name = ifXEntry.ifName;
 - my \$inOctets = ifEntry.ifInOctets;
 - my \$outOctets = ifEntry.ifOutOctets;
 - my \$inFastOctets = ifXEntry.ifHCInOctets;
 - my \$outFastOctets = ifXEntry.ifHCOctets;
 - FireTrigger("interface-row");

General rules are:

If a cross table poll defines only scalars, then we collect as many as the device supports and then call the user's poll function once. If none of the scalars can be gathered, then we do not call the user's poll function.

If a cross table poll is for a table or set of commonly indexed tables, then we call the user's poll function for each combined row. If there are no rows, then of course we don't call the poll function at all.

If a cross table poll is for a table or set of commonly indexed table, plus one or more scalars, then we call the user's poll function for each combined row, adding in the scalars. And if there are no rows to be found, then even though some of the scalars were gathered, we do not call the poll function at all.



For all of the above, we always call the OnStart function before we begin the polling and we always call the OnEnd function once the polling is completed.

IPv6 Changes

The SNMP RFCs are woefully absent on how Trap Forwarding is to be handled in the IPv6 world. The RFCs very well spell out the procedures for IPv4 but nothing for IPv6 - and there are several places in the rules for Trap Forwarding where there simply is -at present- no answer. (Yes, there are RFCs for how Polling is to be forwarded between IPv4 and IPv6, but nothing for Trap Forwarding.)

Despite this we have the strong story that when IPv6 does start to occur in someone's environment, we are uniquely situated to provide forwarding for them. Since our product is entirely software based and under our control, we can work with any customer to decide upon a strategy for providing IPv6 support as part of their Trap Forwarding environment. An agreement on how to proceed can allow us/them to update their mtrapid probe rules file to receive IPv6 forwarding information. NerveCenter can be readily modified to provide the agreed IPv6 forwarding. The time needed for us to modify NerveCenter for this is under a week - meaning we could support it at any point when it become of interest to them. Hardware-based products will not be able to provide such flexibility or timeliness.

IPv6 Improvements and corrections:

- Correction/Enhancement of IPv6 address handling. Repairs made to 'ncget', 'ncping' and 'ncserver'/ 'ncsnmpoller'. Pings as well as SNMP polls were failing when sent to any IPv6 address.
- Repairs made to NCServer's support of the 'Query' and 'Ping' features found in the 'Client' application.
- Addition of several built-in triggers. These represent the error conditions that Polls might incur when performing ICMP 'ping' or SNMP query operations. These additions are sub-codes of the "Destination Unreachable" response that ICMP can be used to report.
 - ICMPV6_NO_ROUTE
[RFC4443](#), Section 3.1 "Destination Unreachable", Code 0: "No route to destination".
 - ICMPV6_ADMIN_PROHIBITED
[RFC4443](#), Section 3.1 "Destination Unreachable", Code 1: "Communication with destination administratively prohibited".
 - ICMPV6_NOT_A_NEIGHBOR
[RFC4443](#), Section 3.1 "Destination Unreachable", Code 2: "Beyond scope of source address".



- ICMPV6_ADDR_UNREACHABLE
[RFC4443](#), Section 3.1 "Destination Unreachable", Code 3: "Address unreachable."
- ICMPV6_PORT_UNREACHABLE
[RFC4443](#), Section 3.1 "Destination Unreachable", Code 4: "Port unreachable."

Other changes to 6.1

The nccmplistener utility is only run by the ncserver when the system has been configured to 'Process ICMP Messages as Traps' (in NCAdmin.exe)

Log file changes:

- ncserver: Now writes Manager-Configuration.csv and Manager-Performance.csv to /var/opt/NerveCenter/tables/. Manager-Performance.csv is a table view of the most current rows of TraceQueues.log
- nccmplistener: Now writes log messages to /var/opt/NerveCenter/log/nccmplistener.log and no longer to ncserver.log

Enhancement to the tracing of login attempts (ie: to ncserver from client, ncadmin or nccmd)

- ncserver:
 - Continues to support "ncstart -pamdebug", which is the equivalent of "trace server logins on" - as defined in the second bullet item.
 - When enabled, writes traces all subsequent login authentication processing to /var/opt/NerveCenter/log/ncsecurity.log (formerly /opt/OSInc/tmp/ncsecurity.log)
 - Starts a new ncsecurity for each login attempt check.
- nccmd: New option to turn on/off tracing of login processing:

trace server logins (on | off)

When set to on, ncserver writes all subsequent login authentication tracing to /var/opt/NerveCenter/log/ncserver.log

When set to off, ncserver ceases trace updates to /var/opt/NerveCenter/log/ncserver.log

```
% /opt/OSInc/bin/nccmd
```

```
NC> connect -s localhost -u nervectr -p .... -n 32506
```

```
NC> trace server status all
```

```
No tracing enabled now
```

```
NC> trace server logins on
```

```
Tracing logins enabled
```



```
NC> trace server status all
Tracing enabled for
  logins
NC> trace server logins off
Disabled trace for logins
NC> trace server status all
No tracing enabled now
NC> trace server -?
Trace:  traces specified objects in the active server
Usage:
To trace logins
  trace server logins [[on/off]]
  Check file ncsecurity.log
To trace traps
...
NC> quit
```

- nccmd: New option '-t' to force tracing of a login attempt:

```
connect -s serverhost -n port -u username -p password [ -t ]
```

The '-t' is optional. If used, the '-t' forces tracing of the authentication processing for this login attempt. Using this option does not change the permanent setting of the above server logins tracing flag.

When '-t' is used, ncserver activates (if not already enabled) trace logging to `/var/opt/NerveCenter/log/ncserver.log`.

```
% /opt/OSInc/bin/nccmd
```

```
NC> connect -s localhost -u nervectr -p .... -n 32506 -t
```

The benefit of this option is to allow enabling the login tracing at the moment when a difficulty with gaining access is countered and tracing was not already enabled (as with using `ncstart -PAMdebug` or `nccmd's trace server logins on`). As it is already 'too late' to activate these options, the '-t' allows the tracing to `/var/opt/NerveCenter/log/ncsecurity.log` to be forced.

- Relational Operator Choice for Alarm Counter Action

The Alarm Counter Action allows for selection of what relation can be used. Prior to BLD17, the only choice was 'equals'. This has been made selectable with a range of

- equals



- not equals
- less than
- less than or equals
- greater than
- greater than or equals

All Alarms inherited into NC6.1 BLD17+ will auto convert their Alarm Counter Actions to use 'equals' as the default - since that was the only prior ability.



LogMatrix Technical Support

LogMatrix is committed to offering the industry's best technical support to our customers and partners. You can quickly and easily obtain support for our NerveCenter proactive IT management software.

Professional Services

LogMatrix offers professional services, when customization of our software is the best solution for a customer. These services enable us, in collaboration with our partners, to focus on technology, staffing, and business processes as we address a specific need.

Educational Services

LogMatrix is committed to providing ongoing education and training in the use of our products. Through a combined set of resources, we can offer quality classroom style or tailored on-site training.

Contacting the Customer Support Center

Telephone Support

Phone: 1-800-892-3646 or 1-508-597-5300

E-Email support

E-mail: techsupport@logmatrix.com.

Online Access

For additional NerveCenter support information, please go the LogMatrix website www.logmatrix.com for access to the following sections of information.

Software Alerts – latest software alerts relative to NerveCenter.

User Community Access

You can seek as well as share advice and tips with other NerveCenter users at

<http://community.logmatrix.com/LogMatrix/> .

Contact Support@logmatrix.com for:

- Patches and Updates – latest installation files, patches and updates including documentation for NerveCenter.