Syslog NodeBrain Module

Release 0.8.17

Syslog NodeBrain Module
August 2014
NodeBrain Open Source Project

Release 0.8.17

Author: Ed Trettevik

History

2012-06-04	Title: *Syslog NodeBrain Module* Author: Ed Trettevik <eat@nodebrain.org> Publisher: NodeBrain Open Source Project
2012-06-04	Release 0.8.10 • This is a first edition.

Preface

This manual is intended for users of the Syslog NodeBrain Module, a plug-in for sending and receiving syslog messages. The reader is expected to be familiar with the basic concepts of NodeBrain. See www.nodebrain.org for general information on NodeBrain.

Documents

NodeBrain Guide - Information on using `nb`
NodeBrain Tutorial - A gentle introduction to `nb` and the rule language
NodeBrain Language - Rule language syntax and semantics
NodeBrain Library - C API

Document Conventions

Sample code and input/output examples are displayed in a monospace font, indented in HTML and Info, and enclosed in a box in PDF or printed copies. Bold text is used to bring the reader's attention to specific portions of an example. In the following example, the first and last line are associated with the host shell and the lines in between are input or output unique to NodeBrain. The `define` command is highlighted, indicating it is the focus of the example. Lines ending with a backslash \ indicate when a command is continued on the next displayed line. This is supported by the language within source files, but not for other methods of command input. If you copy an example of a command displayed over multiple lines, you must enter it as a single line when used outside the context of a source file.

```
$ nb
> define myFirstRule on(a=1 and b=2) mood="happy";
> assert mood="sad";
> show mood
mood = "sad"
> assert a=1,b=2,c=3,d="This is an example of a long single line that",\
    e="we depict on multiple lines to fit on the documnet page";
2008/06/05 12:09:08 NB000I Rule myFirstRule fired(mood="happy")
> show mood
mood = "happy"
> quit
$
```

Table of Contents

1 Concepts

The Syslog module implements nodes that send or receive syslog messages.

2 Tutorial

I decided that it was not wisdom that enabled [poets] to write their poetry, but a kind of instinct or inspiration, such as you find in seers and prophets who deliver all their sublime messages without knowing in the least what they mean.
—Socrates (469 BC–399 BC), in "Apology," sct. 21, by Plato

If your sublime messages are delivered via the syslog protocol, you may prefer to use a Syslog node instead of an Audit node. This enables NodeBrain to respond immediately to arriving syslog UDP packets without waiting to poll a log file.

You have a couple options. You can configure your Syslog node to listen on UDP port 514 on a server that doesn't already have a syslog daemon. However, if you need (or want) more flexibility, I recommend that you use NodeBrain in combination with `syslog-ng`. In that case, you configure NodeBrain to listen on a different port and configure `syslog-ng` to forward all or selected syslog entries to NodeBrain.

The example below is configured to listen on UDP port 1514, assuming `syslog-ng` is used to forward syslog to NodeBrain.

```
#!/usr/local/bin/nb -d
# File: tutorial/Syslog/syslog.nb
-rm syslog.log
set log="syslog.log",out=".";
define syslog node cache(~(h(8))):(~(1h):route,appl,group,node,object,severity,text(1));
syslog. define alarm if(text._hitState):$ -|mail.form ...
... source=tutorial route="${route}" appl="${appl}" group="${group}" ...
... node="${node}" severity="${severity}" text="${text}" >> mail.log
syslog. define audit node syslog("syslog.nbx",1514);
```

You should reference the documentation for `syslog-ng` to see how to configure it to forward to NodeBrain. Here's an example to get you started.

```
destination nodebrain { udp("localhost" port(1514)); };
filter f_nodebrain { host("(humpty|dumpty).mydomain.com|franklin.otherdomain.com"); };
log { source(src);  filter(f_nodebrain); destination(nodebrain); };
```

Refer to the *Audit Node* tutorial above for a sample `syslog.nbx` file. Refer to the *Translator Node* tutorial for more information on coding a translator.

3 Commands

This section describes commands used with a Syslog node.

3.1 Define

The `define` command is used to create a Syslog node.

Syntax

syslogDefineCmd	::= **define** ș *term* ș **node** [ș *syslogDef*] •
syslogDef	::= **syslog(**"*translator*"**,***socket***);**
translator	::= name of translator file (*.nbx)
socket	::= *port* | "**udp:**//*interface***:***port*" | "**udp:**//*socketfile*"

The *translator* arguments may be any cell expression that resolves to a string value containing a filename. The value at definition time is used. A syslog node will not response to or recognize changes to the value of expressions for these parameters.

The *translator* by convention has a ".nbx" suffix. See "Translators" under the `DEFINE` command in the *NodeBrain Language Reference* for instructions on coding translator files.

```
define syslog node syslog("messages.nbx",1514);
define syslog node syslog("messages.nbx","udp://0.0.0.0:514");
define syslog node syslog("messages.nbx","udp://socket/syslog-foobar");
```

3.2 Assert

Assertions are not supported by this module.

3.3 Disable

The `disable` commands may be used to stop the node from listening for syslog messages on the specified port.

```
disable node
```

3.4 Enable

The `enable` command may be used to start listening for syslog messages on the specified port.

```
enable node
```

A Syslog node is automatically enabled when an agent goes into background mode (daemonizes). So the enable command is only required if you want to enable it when running in a different mode, or to re-enable the node after it has been disabled.

3.5 Node Commands

A trace mode can be toggled on or off to assist in debugging translator rules. When trace is on, lines from the file are displayed when processed.

```
node:trace
node:notrace
```

3.6 Module Commands

The Syslog module currently implements no module commands.

4 Triggers

All the triggers of a Syslog server node are implemented by the specified translator. A Syslog client node has no triggers.

Index