Syslog NodeBrain Module

Version 0.9.03

Syslog NodeBrain Module
December 2014
NodeBrain Open Source Project

Version 0.9.03

Author: Ed Trettevik

Copyright © 2014 Ed Trettevik <eat@nodebrain.org>

Permission is granted to copy, distribute and/or modify this document under the terms of either the MIT License (Expat) or the NodeBrain License.

MIT License

Copyright © 2014 Ed Trettevik <eat@nodebrain.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

NodeBrain License

Copyright © 2014 Ed Trettevik <eat@nodebrain.org>

Permission to use and redistribute with or without fee, in source and binary forms, with or without modification, is granted free of charge to any person obtaining a copy of this software and included documentation, provided that the above copyright notice, this permission notice, and the following disclaimer are retained with source files and reproduced in documention included with source and binary distributions.

Unless required by applicable law or agreed to in writing, this software is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

History

2012-06-04 Title: *Syslog NodeBrain Module*
Author: Ed Trettevik <eat@nodebrain.org>
Publisher: NodeBrain Open Soure 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 **1**

2 Tutorial **3**

3 Commands **5**

3.1 Define 5
3.2 Assert 5
3.3 Disable 5
3.4 Enable 5
3.5 Node Commands 6
3.6 Module Commands 6

4 Triggers **7**

Index **9**

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** å *term* å **node** [å *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

A

assert command 5

C

commands 5
Concepts 1

D

define command 5
disable command 5

E

enable command 5

N

node commands 6

T

triggers 7
tutorial 3