DTAG - TMCZ – Log Splitting Script

Version 1.0 - February 25, 2012

table of contents

Contents

[**1.** **INTRODUCTION** 1](#_Toc318730388)

[**2.** **InstallatioN** 2](#_Toc318730389)

[*2.1* *Deploying the script on the server* 2](#_Toc318730390)

[*2.2* *Installing the script as a LINUX service* 2](#_Toc318730391)

[*2.3* *Installing the log-rotation scripts for the interface logs* 3](#_Toc318730392)

[3. OPERATION 5](#_Toc318730393)

[Appendix A 7](#_Toc318730394)

Document Release Notes

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Comments** |
| 1.0 | Steven Sciriha | Jan 02 , 2012 | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **INTRODUCTION**

This document is intended to outline the installation and operation process for the Log Splitting script, LogSplitter.pl. The script is designed to be deployed on a PCRF server and its scope is to allow the user to break out the main PCRF log into separate log files, each representing a specific PCRF interface. The requirements specified a total of 5 different interfaces (JMS, Gx, PC Provisioning, SDB Provisioning and SOAP) each having its own specific and separate log file (if configured). The way this is achieved is through the use of a configuration file that specifies (using a regular expressions pattern) what logs are to be matched and to which interface log the matched log needs to be appended to.

The script generates its own administrative log file that keeps track of whether the script has been started/stopped and if any issues/errors have been detected. On startup the script validates that the configuration file is in fact valid, if not it will indicate what the problem is. All aspects of Linux processes and log maintenance are covered in this guide including proper installation of the script as a Linux process and log-rotation configuration for the new interface logs introduced.

There are a number of considerations and limitations related to this log splitting project. Please refer to Appendix A (Section 3) for further details.

# **InstallatioN**

## *Deploying the script on the server*

* Copy the script ***LogSplitter.pl***, included in the project package (LogSplit.zip) into a suitable directory such as /stage.
* The variable *$logfile\_dir* located in line 12 of the script needs to be updated to reflect the desired location for all log files generated by the script. This includes the 5 interface logs and the administrative log file, LogSplitter.log generated by log splitting script. The default setting is the Linux system log directory “/var/log”.
* Apply the correct ownership and permissions to the script

# **chown root:root /stage/LogSplitter.pl**

# **chmod 755 /stage/LogSplitter.pl**

## *Installing the script as a LINUX service*

* The script needs to be installed as a Linux service to allow for proper

startup when the system is rebooted.

* The script will be added to the runtime levels 2, 3, 4 and 5 with a startup priority of 90 and shutdown priority of 20. If any of these settings need to be updated please contact AMDOCS support.
* Place the script ***logsplitterd*** included in the feature package into the directory /etc/init.d on the Linux system.

As the **root** user:

# **chmod 755 /etc/init.d/logsplitterd**

* The script contains 3 important variables that need to be updated according to the intended script operation for the system.

1. *path\_to\_script*: This variable represents the absolute directory location of the log splitting script.
2. *logfile\_to\_read*: This variable represents the PCRF logfile that the script with be reading from.
3. *config*: This variable represents the absolute directory location of the configuration file. Please see “Operations” and “Appendix A(Section 2)” for further information on how to set up the configuration file

Example:

*path\_to\_script="/WideSpan/mon/LogSplitter.pl"*

*logfile\_to\_read="/var/log/messages"*

*config="/WideSpan/mon/SplitConfig.cfg"*

* Install the log splitting startup script as a Linux process

# **chkconfig --add logsplitterd**

* Verify that the process has been installed successfully

# **chkconfig --list | grep -i logsplitterd**

*logsplitterd 0:off 1:off 2:on 3:on 4:on 5:on 6:off*

* Start and stop the process to verify basic operation

# **/etc/init.d/logsplitterd start**

*Starting LogSplitter:*

# **ps –ef |grep –v grep | grep LogSplitter**

root 15796 1 0 21:26 pts/1 00:00:00 /usr/local/bin/perl /stage/LogSplit\_v1.0/LogSplitter.pl /var/log/messages /stage/LogSplit\_v1.0/SplitConfig.cfg

# **/etc/init.d/logsplitterd stop**

*Stopping LogSplitter:*

*[ OK ]*

## *Installing the log-rotation scripts for the interface logs*

* To ensure proper operation during log-rotation the log-splitting script has been designed to check for logs that have rotated on both the input and output log streams.
* The log-rotation Linux configuration script ***logsplit*** has been included in the project package, for the interface logs being written to by the log-splitting script.
* The log-rotation configuration script includes a number of configurable parameters. Items highlighted in green can be adjusted according to typical system settings, but items highlighted in red **MUST** remain as defined in the script provided to ensure proper operation. Please note this configuration script assumes that all logs being written to by the log splitting script are located in “/var/log”, any variation from this needs to be reflected in the first line of the configuration script:

**/var/log/JMSlog /var/log/SOAPlog /var/log/PCPROVlog /var/log/SDBPROVlog /var/log/GXlog{**

**daily**

**size 1k**

**rotate 3**

**postrotate**

**/bin/kill -HUP `cat /tmp/LogSplitter.pid 2> /dev/null` 2> /dev/null || true**

**endscript**

**}**

* Copy the updated log-rotation script **logsplit** into the directory */etc/logrotate.d* and apply the correct ownership and permissions:

# **chown root:root /etc/logrotate.d/logsplit**

# **chmod 644 /etc/logrotate/logsplit**

* The final installation step is the addition of the log-splitting script, /var/log/LogSplitter.log to the system’s log-rotation schedule using typical system settings. Please see “Operations” for further information.

OPERATION

* The log-splitting script takes 2 command-line arguments; The first argument is the main PCRF operational log that the script will be reading in; The second argument is the configuration file that defines what specific logs to match and to which corresponding interface log file the matched log should be appended to.
* The configuration file **MUST** match the following pattern:

/<REGULAR EXPRESSION TO MATCH>/<INTERFACE LOG>/

/<REGULAR EXPRESSION TO MATCH>/<INTERFACE LOG>/

/<REGULAR EXPRESSION TO MATCH>/<INTERFACE LOG>/  
 .

.

.

*<REGULAR EXPRESSION TO MATCH> is* the regex pattern that

the log splitting script will using to determine which logs to match.

*<INTERFACE LOG>* indicates which specific interface log needs to be

appended with any logs matching the previous regex pattern.

* <INTERFACE LOG> **MUST** be match one of the following interface logs:

*GXlog*, *PCPROVlog*, *SDBPROVlog*, *SOAPlog* and *JMSlog*. Each of these logs map to a specific interface as defined in the PSA requirements (see Appendix A, Section 1)

* By default, each of the interface logs listed above exist under the directory */var/log* or whatever the variable $logfile\_dir (line 12 of LogSplitter.pl) has been set to.
* A sample configuration file has been included in Appendix A (Section 2)
* The log-splitting script is designed to run continuously as a background process and should startup at boottime if all the steps listed above are executed correctly
* If the script needs to be started or stopped it can be done using the following start/stop commands:

# **/etc/init.d/logsplitterd start**

# **/etc/init.d/logsplitterd stop**

* The script produces a log file that which logs the scripts startup/shutdown process and if any issues/errors are encountered by the script. The log ***LogSplitter.log*** is located under /var/log, by default. This location can be changed by updating the variable *$logfile\_dir* in the LogSplitter script as per the installation instructions above.
* Typical logs for a successful startup/shutdown are as follows:

Mon Mar 5 16:07:52 GMT 2012: Starting up logsplitting script ...

Loading configuration file /WideSpan/mon/SplitConfig.cfg...

The logsplitter process has started up successfully and is now running...

Mon Mar 5 16:09:54 GMT 2012: LogSplitter has been stopped...

* An typical implementation example is as follows:

*Assuming the following set of logs are incoming into the main PCRF log:*

Feb 21 13:04:23 bonpcf01a\_pcrf: INFO BPCOP(303) policy pull on START for 4915159701279@telekom.de SUCCESSFUL

Feb 21 13:04:56 bonpcf01a\_pcrf: ERR BPCOP(201) getPolicy for 5555555555@telekom failed:

Feb 21 13:11:46 bonpcf01a\_pcrf: INFO BPCOP(318) policy pull on STOP for 4915159701279@telekom.de SUCCESSFUL

Feb 21 13:16:45 bonpcf01a\_pcrf: INFO BPCOP(1) account change for 4915159701279@telekom.de completed

*Assuming the following configuration file has been loaded:*

/policy pull on ST[ARTOP]{1,3} for [0-9]{10}@[A-Za-z]+.de/GXlog/

/getPolicy for [0-9]{10}@[A-Za-z]+.de/JMSlog/

/account change for [0-9]{10}@[A-Za-z]+.de/SDBPROVlog/

*The contents of each of the configured interface logs will be as follows:*

# **cat /var/log/GXlog**

Feb 21 13:04:23 bonpcf01a\_pcrf: INFO BPCOP(303) policy pull on START for 4915159701279@telekom.de SUCCESSFUL

Feb 21 13:11:46 bonpcf01a\_pcrf: INFO BPCOP(318) policy pull on STOP for 4915159701279@telekom.de SUCCESSFUL

# **cat /var/log/JMSlog**

Feb 21 13:04:56 bonpcf01a\_pcrf: ERR BPCOP(201) getPolicy for 5555555555@telekom failed:

# **cat /var/log/SDBPROVlog**

Feb 21 13:16:45 bonpcf01a\_pcrf: INFO BPCOP(1) account change for 4915159701279@telekom.de completed

* If any issues or concerns arise please contact Steven Sciriha (email: Steven.Sciriha@amdocs.com)

Appendix A

**Section 1:**

Log Splitting Requirements (EXCERPT FROM PSA DOCUMENT)

*BWS will work with TMCZ operations to identify the messages in the log files that need to be placed into separate files. A near real-time script will be developed by BWS to parse the main PCRF log file based on the identified messages. The main PCRF log file will continue to hold all logs. Only a subset of the messages will be filtered by the script. A maximum of five separate log files will be created**covering DT SOAP, Gx (Gx log will be provided by DRA), JMS and provisioning (SDB and PC provisioning) interfaces. System performance may be adverp-sely affected and the impact will depend on the quantity of messages selected and the number of files to be written. This script will only be used as a temporary solution until either the CR for separated log files is developed or TMCZ decide to not place an order for the CR.*

*The script is not considered part of the Amdocs software deliverables and therefore will not be included as part of project acceptance. The scripts will be developed prior to start of acceptance testing while meeting the needs/requirements of TMCZ that will be agreed during the scripts development process. The script will not be supported by Amdocs beyond 12 months of initial deployment.  T-Mobile Czech will have the option to continue using and supporting it on their own after this initial period.*

**Section 2:**

Sample Configuration File:

*#this is a comment*

*#LogSplitter.pl Configuration File*

*/DHCPACK [A-Za-z]+ [0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}/SOAPlog/*

*/DHCPREQUEST[A-Za-z0 ]+ [0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3} port [0-9]{1,6}/JMSlog/*

*/[A-Za-z0 ]+ [0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3} -- [A-Za-z ]+[0-9]{3}/SOAPlog/*

*/account changed[0-9]+?8\*/PCPROVlog/*

*/policy pull for [0-9A-Za-z]\*/GXlog/*

*/User [0-9]{10}@sumdomain.com Deleted/SDBPROVlog/*

*/policy push[0-9]+/GXlog/*

*/PCEF unavailable/GXlog/*

**Section 3**:

Considerations/Limitations pertaining to the log splitting feature:

1. *Disk space could be an issue with at least twice the logs now being generated.*
2. *Performance impact could be significant with at least twice the I/O operations to read from the logfile and write to the new logfiles per interface.*
3. *Retention, Log rotation needs to be managed for all the new interface logs.*
4. *Given the implementation, where the script always defaults to the end of the main logfile to check for new logs, starting and stopping the log splitter at any time will result in lost logs generated in the interim.*
5. *Since this log splitting feature is required to assist troubleshooting a potential substitute would be the Subscriber Trace Project P1832 that allows one to trace a specific subscriber call.*
6. *JMS logs can only be captured if the logging level for those type of logs is set to DEBUG.*
7. *Something that needs to be addressed in the medium to long term is the fact that BPCOP logs are not created for each interface event. This needs to be addressed through a future defect and is a potential loophole.*
8. *The actual interface is not indicated on the individual logs which makes it very difficult to identify which interface a log is representing. Again this needs to be addressed through a future defect and is a potential loophole.*
9. *SDB provisioning logs only includes account modification notifications on the PCRF*
10. *PC provisioning would only cover TEMP TIER provisioning logging…not service/network configuration changes like provisioning a new service for example.*
11. *Proper testing would require obtaining a list of logs that represent each interface to create the regex matching expressions and include those logs in all lab testing (a 9.4.1 lab would be required for final testing)*