

# INTEGRATING THE EPICS IOC LOG INTO THE CSS MESSAGE LOG\*

K.U. Kasemir, E. Danilova, ORNL, Oak Ridge, TN 37831, U.S.A.

## Abstract

The Experimental Physics and Industrial Control System (EPICS) includes the “IOLogServer”, a tool that logs error messages from front-end computers (Input/Output Controllers, IOCs) into a set of text files. Control System Studio (CSS) includes a distributed message logging system with relational database persistence and various log analysis tools. We implemented a log server that forwards IOC messages to the CSS log database, allowing several ways of monitoring and analyzing the IOC error messages.

## EPICS IOC LOG

The Experimental Physics and Industrial Control System (EPICS) front-end computers, commonly referred to as Input/Output Controllers (IOCs), supports a simple TCP-based logging system. IOCs connect to the “IOLogServer”, by default on TCP port 6500, and send basic text messages. A new line character terminates each text message. The IOLogServer annotates each message with the current time/date and the IP address of the IOC, and then writes them to a rotating set of text files [1].

This logging mechanism has proven to be very robust, but it lacks tools to analyze the log messages. Use of the IOC log at the Spallation Neutron Source (SNS) was limited to a post-mortem search for clues in case of IOC problems, based on UNIX command-line text tools like “grep” or directly reading the log files in a text editor.

## CSS LOG

CSS applications utilize a Java Message Server (JMS) to issue log messages [2]. The same mechanism is also used for network traffic between components of the CSS alarm system [3]. All CSS JMS messages can be routed to a relational database (RDB) for later analysis.

Fundamentally, each JMS message can consist of arbitrary property/value pairs, but the following standard properties should be included to allow analysis of messages from different sources:

- TYPE: A value of ‘log’ indicates a log message.
- DATUM: Date and time of the message.
- TEXT: For log messages, this is the actual message text.
- SEVERITY: An indicator for the message severity, for example “INFO”
- HOST: The name or IP address of the computer that sent the message.
- USER: Name of the user who submitted the message.
- APPLICATION-ID: Name of the application that submitted the message.

\* SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy

Fig. 1 shows the generic CSS Message Log Viewer that can display selected properties of messages that have been logged to the RDB. This viewer also allows filtering, for example to restrict the display to messages of a certain TYPE, messages that originate from a specific HOST, or messages with TEXT that contains a certain word. This viewer allows interactive review of messages.

Since the messages are stored in a relational database, the information is accessible to a wide variety of tools. It is for example relatively easy to create reports that are targeted to the analysis of a specific problem, as we will describe in the section called JSP Message Reports.

ID	TIME	DELTA	TYPE	TEXT	NAME	STATUS	SEVERITY	CURRE.	VA.	CREATETIME	USER	HOST	APP.	CLASS	FIDEL.
59316	2008.11.02.095519.769		alarm	STATE	FE_MP...	LOLO	OK				Shu	pc714	Ala...		
59315	2008.11.02.095519.574		000000	alarm	STATE	FE_MP...	LOLO	OK			Shu	pc714	Ala...		
59314	2008.11.02.095519.123		000006	log	* Control System...	INFO	OK			2008.11.02...	Shu	pc714	CSS	msg...	Appl...
59313	2008.11.02.095440.665		000104	alarm	STATE	KCS_T...	OK	OK	OK		ky@	ky@...	Ala...		
59312	2008.11.02.095440.645		000000	alarm	STATE	CF_ST...	OK	OK	OK		ky@	ky@...	Ala...		
59311	2008.11.02.095440.619		000000	alarm	STATE	FE_MP...	OK	OK	OK		ky@	ky@...	Ala...		
59310	2008.11.02.095440.610		000000	alarm	ACK	KCS_T...					ky@	ky@...	CSS		
59309	2008.11.02.095440.593		000000	alarm	ACK	CF_ST...					ky@	ky@...	CSS		
59308	2008.11.02.095440.585		000000	alarm	ACK	FE_MP...					ky@	ky@...	CSS		
59307	2008.11.02.095356.279		000012	log	* Control System...	INFO	OK			2008.11.02...	Shu	pc714	CSS	msg...	Appl...
59306	2008.11.02.095356.260		000012	alarm	STATE	FE_MP...	LOLO	OK	2		ky@	ky@...	Ala...		
59305	2008.11.02.095356.268		000000	alarm	STATE	FE_MP...	LOLO	OK			ky@	ky@...	Ala...		
59304	2008.11.02.095356.256		000000	alarm	STATE	FE_MP...	LOLO	OK			ky@	ky@...	Ala...		
59303	2008.11.02.095356.240		000000	alarm	STATE	FE_MP...	LOLO	OK			ky@	ky@...	Ala...		
59302	2008.11.02.095340.071		000023	log	* Control System...	INFO	OK			2008.11.02...	Shu	pc714	CSS	msg...	Appl...
59301	2008.11.02.095334.387		000005	alarm	STATE	CF_ST...	LOLO	OK			Shu	pc714	Ala...		
59300	2008.11.02.095334.376		000000	alarm	STATE	CF_ST...	LOLO	OK	3.3...		ky@	ky@...	Ala...		
59299	2008.11.02.095334.384		000001	alarm	STATE	CF_ST...	LOLO	OK	3.3...		ky@	ky@...	Ala...		
59298	2008.11.02.095334.370		000000	alarm	STATE	CF_ST...	LOLO	OK	3.3...		ky@	ky@...	Ala...		
59297	2008.11.02.095334.385		000018	log	* Control System...	INFO	OK			2008.11.02...	Shu	pc714	CSS	msg...	Appl...
59296	2008.11.02.095334.401		000014	TALK	MAJOR alarm...						Shu	pc714	Ala...		
59295	2008.11.02.095334.381		000000	alarm	STATE	CF_ST...	LOLO	OK	3.3...		ky@	ky@...	Ala...		
59294	2008.11.02.095334.370		000000	alarm	STATE	CF_ST...	LOLO	OK	3.3...		ky@	ky@...	Ala...		
59293	2008.11.02.095334.391		000002	alarm	STATE	CF_ST...	LOW	MINOR	MINOR	4.8...	ky@	ky@...	Ala...		
59292	2008.11.02.095334.380		000000	alarm	STATE	CF_ST...	LOW	MINOR	MINOR		Shu	pc714	Ala...		
59291	2008.11.02.095334.371		000000	TALK	MAJOR alarm...						Shu	pc714	Ala...		
59290	2008.11.02.095341.298		000014	alarm	STATE	KCS_T...	STAT	MINOR	OK	SN...	ky@	ky@...	Ala...		
59289	2008.11.02.095341.284		000000	alarm	STATE	KCS_T...	STAT	OK			Shu	pc714	Ala...		
59288	2008.11.02.095340.175		000001	alarm	STATE	FE_MP...	LOLO	OK	2...		ky@	ky@...	Ala...		
59287	2008.11.02.095340.164		000000	alarm	STATE	FE_MP...	LOLO	OK			Shu	pc714	Ala...		
59286	2008.11.02.095336.273		000013	alarm	STATE	FE_MP...	LOLO	OK			Shu	pc714	Ala...		
59285	2008.11.02.095336.364		000000	alarm	STATE	FE_MP...	LOLO	OK	2		ky@	ky@...	Ala...		

Figure 1: CSS Message History Browser.

## IOC LOG SERVER REPLACEMENT

To benefit from the existing infrastructure for CSS message logging, we implemented a new IOC Log Server in Java that operates similar to the original EPICS IOLogServer. It also listens on TCP port 6500 for text messages, but each received line of text is written to the CSS message RDB with the following properties:

- TYPE: A value ‘iocLog’ identifies IOC messages.
- CREATETIME: Data and time when message was received.
- TEXT: Original message text.
- HOST: IP address of the network client, i.e. the IOC.
- SEVERITY: Defaults to “INFO”
- APPLICATION-ID: “IOLogServer”

To EPICS IOCs, the replacement of the original log server for text files with the new implementation is transparent. We simply start either the old or the new log server, and IOCs will use the current instance as soon as they attempt to log a message.

## Filtering

We noticed early on that the IOC log messages were dominated by repeated messages. Certain IOCs would

often send the exact same message every few seconds. To reduce the amount of space used by such messages in the RDB, we added a simple filter:

If the same network client (i.e. IOC) sends the same message text, this is not logged as a separate message. Instead, a repeat count is updated in the REPEATED property of the original message.

The filtering is based on the last message received from each network client, only comparing a new message with the last message from that client. If a client keeps sending a message with text “A”, this will be filtered by the repeat count mechanism. If a client should repeat two messages in a pattern “A”, “B”, “A”, “B”, ..., this will not be filtered out. While we do observe this pattern, it has at this time not been frequent enough to warrant checking all recent messages of a client for repetitions.

In addition to filtering repeated messages, there is a pattern-based filter. Based on regular expressions for the message content, it can be configured to ignore messages that match a pattern. This can for example be used to suppress known benign status messages that we prefer to exclude from the log.

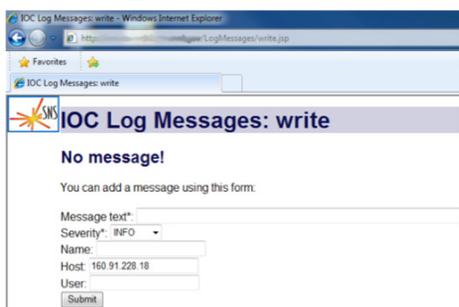


Figure 2: Web form for submitting a message.

## JSP LOG WRITER

The IOC Log Server replacement is convenient for EPICS IOCs, but less accessible to control system front-end computers that are implemented by other means.

By wrapping the essential Java code for logging messages to the RDB as a Java Server page (JSP) for Tomcat [4], we created a web service that can for example be used by programs implemented in National Instruments LabVIEW to submit log messages.

The web service is accessible via a URL similar to the following for submission of a basic text message:

`http://server/LogMessages/write.jsp?TEXT=...`

If desired, additional URL parameters like SEVERITY or USER can be provided to set the associated message properties.

If no message parameters are included in the URL, a web form as shown in Fig. 2 is presented to allow interactive entry of the message content.

## JSP MESSAGE REPORTS

With all IOC log messages available in the relational database, we are now able to create various online reports. Fig. 3 shows a Java Server Page (JSP) that summarizes recent IOC log messages.

A pie chart depicts the “Top 10” IOCs that produced the highest number of messages in the analyzed time period, which defaults to the last hour. Tables list repeated messages as well as all messages of the time period. Both tables contain the message text but also time, severity, host, number of repeats, and optional user name. By default, the tables are sorted by the time when each message was created, but users can sort the tables by any column by clicking on the table column header.

A web form allows users to configure the time range, provide pattern for the IOC name, and choose the number of messages to display.

A detailed list of messages for a particular IOC is accessible by either clicking on the associated sector on the pie chart sector (if this IOC is one of top 10 message producers), or by selecting the IOC name from drop-down lists of IOCs that are sorted by name as well as message count.

These JSP reports are available to any computer with a web browser at our site.

## SUMMARY

Without any change to EPICS IOCs, we were able to transparently route IOC log messages from the original file-based system to the CSS message log that is stored in a relational database. Web reports show for example which IOC produced the most messages in the last few days.

In the past, our IOC log files were only consulted “after the fact” when an IOC had run into severe problems. Now we are able to spot an increase in message traffic, hopefully anticipating a more severe error.

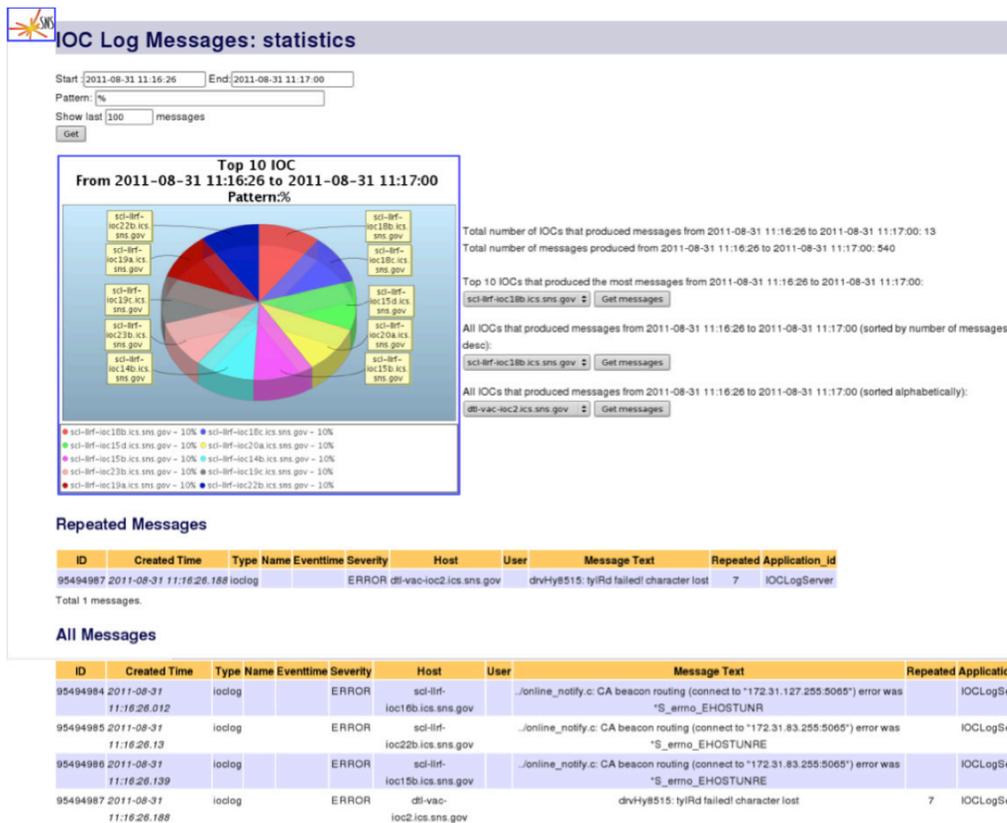


Figure 3: JSP Report of IOC Log Message statistics.

**REFERENCES**

- [1] M. Kraimer, J. Anderson et al, "EPICS Application Developer's Guide, Release 3.14.12", <http://aps.anl.gov/epics/base/R3-14/12-docs/AppDevGuide.pdf>.
- [2] K. Kasemir, G. Carcassi, "Control System Studio Guide", <http://cs-studio.sourceforge.net/docbook>
- [3] K. Kasemir, "The Best Ever Alarm System Toolkit", ICALEPCS 2009, Kobe, Japan
- [4] Tomcat web application server, <http://tomcat.apache.org>