

# INFRASTRUCTURE MONITORING SYSTEM FOR THE ADVANCED PHOTON SOURCE CONTROL SYSTEM\*

D. Quock<sup>#</sup>, N. Arnold, A. Johnson, ANL, Argonne, IL 60439, U.S.A.

## Abstract

An Infrastructure Monitoring System (IMS) has been developed at Argonne National Laboratory's Advanced Photon Source (APS) to provide immediate notification to the Controls Group on-call staff in the event of failure of critical controls hardware components or software processes. Approximately 5,000 operation-critical controls components and processes have been identified, many of which can be readily monitored through EPICS records. Nagios [1], an open-source host, service, and network monitoring program, has been interfaced to EPICS Channel Access monitoring tools to provide historical tracking of controls infrastructure events, email, and pager notifications to on-call staff, as well as Internet-accessible status displays. Two methods have been evaluated for checking EPICS alarm status data: a modified version of Mauro Giacchini's NAL [2] plug-in for Nagios, and a Perl script using the Channel Access Perl library recently developed by Andrew Johnson. Personal Device Assistant-viewable Web pages are generated using data retrieved from the Nagios relational database NDOUtils, along with data from the APS controls database IRMIS [3]. Performance studies have validated the reliability of Nagios and the Perl Channel Access software.

## GOALS OF APS CONTROLS IMS

A modern control system for a large accelerator consists of *thousands* of components, processes, and applications that must all be continuously functioning for optimum performance. A failure of one of these components may not be immediately noticeable by the machine operators and may continue for some time, possibly discovered several days later. The Infrastructure Monitoring System is intended to *exhaustively* monitor all parts of the control system and provide immediate notification to the on-call controls staff of an exception, in many times even before the machine operators notice the impact on machine performance. To meet these goals, the following project development steps were taken:

- Identify APS Controls IMS categories that will aid on-call staff in promptly deciphering cascading failures;
- Build a high data throughput, EPICS-based IMS software application; and
- Create PDA-friendly user displays of condensed IMS status reports.

## APS CONTROLS IMS CATEGORIES

Ten major control system IMS categories were identified for creating a highly organized list of APS controls EPICS processes, controls hardware components, and other miscellaneous controls processes. These major IMS categories are:

- AOI (Applications Organizing Index)
- CCMS (Component Communication Monitoring System)
- Controls Servers
- Event Receivers
- IOCs
- Machine Status Links
- Nagios Software Components
- PV Gateway
- Timing
- VME/VXI Power Supplies

The major categories were then broken down into sub-categories to provide a three-level controls IMS grouping hierarchy with the lowest level being the actual EPICS process variable, server, or other type of computer process whose operating status is actually being checked and reported upon. For example, the major IMS category IOC has been divided into the following nine sub-categories:

- CALinks
- CPULoad
- LogServer
- Memory
- SaveRestore
- SoftHeartbeat
- TCPConnections
- TaskStatus

The subcategory CALinks contains 783 EPICS process variable (PV) names where each PV name corresponds to a specific IOC's channel access connection status.

All IMS categories (with the exception of the IOC subcategory SaveRestore and the major category AOI) have been populated with either EPICS PVs, computer server names, or computer process names. Scripts written in PHP and Perl languages parse information out of the APS controls relational database IRMIS to generate lists of unique names for IMS.

## APS CONTROLS IMS SOFTWARE ARCHITECTURE

Nagios software can monitor network services (e.g., SMTP, POP3, HTTP, PING) and host resources (e.g., processor load, disk usage). It also provides a simple plug-in design that allows users to develop their own service checks for unique situations. Starting with Mauro

\*Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11367.

<sup>#</sup>quock@aps.anl.gov

Giacchini's work at INFN on adapting EPICS channel access PV status updates to Nagios software as a plug-in, an in-depth investigation was undertaken to determine whether a full-scale, reliable controls IMS for APS could be achieved with Nagios software as the foundation.

The Nagios definition of the states that a service or host can be in matches well with the EPICS PV alarm severity states shown in Table 1.

Table 1: EPICS-to-Nagios Translation

EPICS Alarm Severity	Nagios State
(none)	OK
Minor	Warning
Major	Critical
PV Readback Invalid	Critical
PV Name Not Found	Unknown

The Nagios feature to define network host hierarchy using host groups, hosts, and services was utilized when defining the APS Controls IMS categories in Nagios configuration files as shown here:

- Host Group
  - Host
    - Service
    - Service
    - ...
- IOC
  - CALinks
    - iocacis:ascaDisco
    - iocacis:dbcaDisco
    - ...
  - CPUload
    - iocacis:load
    - iocbbpm1:load
    - ...

Figure 1 shows APS Controls IMS categories as translated and displayed in the Nagios "Hostgroup Summary."

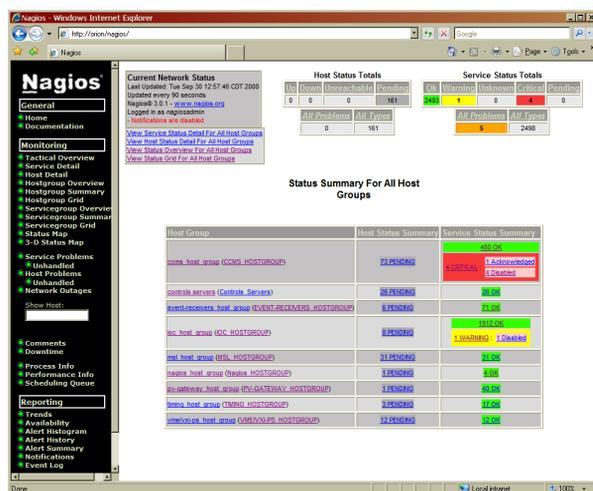


Figure 1: APS controls Nagios display.

## EPICS Perl Channel Access Monitor

Currently, a mixture of approximately 2,500 EPICS PVs, computer servers, and computer processes are being monitored by Nagios software as Nagios "services." All but 35 of these services are EPICS PVs. It is estimated that the number of EPICS PVs that will be included in the APS Controls IMS will ultimately grow to more than 5,000.

The EPICS plug-in for Nagios developed at INFN uses channel access "gets" and can be configured in an active polling mode within the Nagios software on an automated schedule. An example of such a Nagios configuration could be: once every three minutes, Nagios executes the EPICS plug-in for each EPICS PV that is listed as a "service" in Nagios. This type of active polling for such a large number of services in Nagios creates a burden on all IOCs visible to the plug-in, and Nagios software performance degradation is also noticeable. Consequently, an alternative to using the EPICS plug-in for checking on the status of an EPICS PV was developed to reduce both network traffic and processing of unnecessary information within Nagios.

Andrew Johnson has recently developed a Perl interface to the EPICS channel access library. A script was written specifically for the APS Controls IMS that utilizes channel access monitor callbacks. This script runs indefinitely and only pushes "service status updates" to Nagios when a change in an EPICS PV alarm severity state is detected. Nagios has been configured at APS to accept external passive checks through a UNIX pipe for all of the Nagios services that are EPICS PVs. The server CPU usage of this Perl monitoring script for roughly 2,500 quiescent EPICS PVs is negligible (less than 1%).

## PDA-Friendly Web Displays

The Nagios software comes with many fine tools for such things as displaying alert history, producing availability reports, disabling and enabling configuration parameters. However, a phone call at two in the morning from an APS main control room operator to a soundly sleeping controls engineer can be startling. The engineer requires a simplified view of the controls problem at hand for prompt recovery action. This simplified view was created in PHP Web displays (Figure 2) that retrieve summary information from a Nagios NDOUtils relational database using MySQL database software, and controls software and hardware "as-built" information from the APS controls relational database IRMIS. Three levels of IMS PDA-friendly displays are provided, which allow a user to drill down to more levels of detail as needed.

Figure 3 shows how all of the various software components of the APS Controls IMS work together.

## FUTURE ENHANCEMENTS

The APS Controls IMS has been rigorously and successfully tested. The only IMS failure encountered thus far was due to a power supply bump from the electric utility supplier to APS. This loss of electrical power

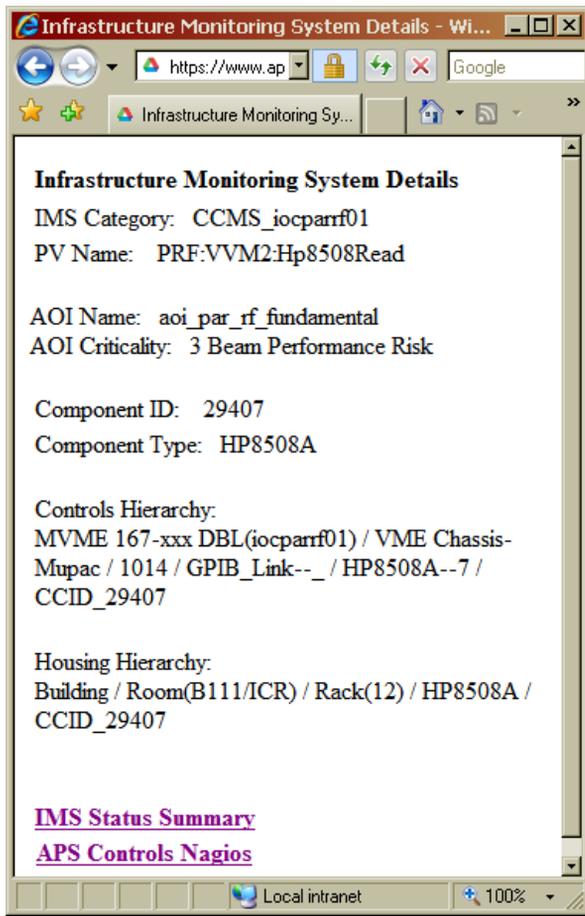


Figure 2: PDA-friendly display.

resulted in lost network communications, and the Perl EPICS channel access monitor script stopped running due to a bus error. This fault was immediately detected by Nagios.

Future enhancements to the APS Controls IMS include:

- Add list of approximately 1,000 designated EPICS PVs, which convey the overall health of accelerator control applications [4], and 300 IOC SaveRestore status readback PVs.
- Continue to refine and add to list of CCMS EPICS PVs. It is estimated that there are 87 unique component types and 2,200 installations of these components at APS that can be effectively monitored by IMS for loss of controls I/O. Only 481 CCMS PVs are included in IMS thus far.
- Create Nagios event handlers that automatically respond to loss of critical processes and take corrective action.

### REFERENCES

- [1] E. Galstad, Nagios Enterprises, LLC, <http://www.nagios.org>.
- [2] M. Giacchini, Laboratori Nazionali di Legnaro, Istituto Nazionale di Fisica Nucleare, <http://www.lnl.infn.it/~epics/NAL.html>.
- [3] D. Quock, N. Arnold et al., "IRMIS," <http://www.aps.anl.gov/epics/irmis>.
- [4] D. Quock, N. Arnold, "Administration of Control Systems at the Advanced Photon Source Using Applications Organizing Index," Proc. of ICALEPCS'07, Knoxville, Tennessee, October 2007, RPPB35, p. 686 (2007); <http://www.JACoW.org>.

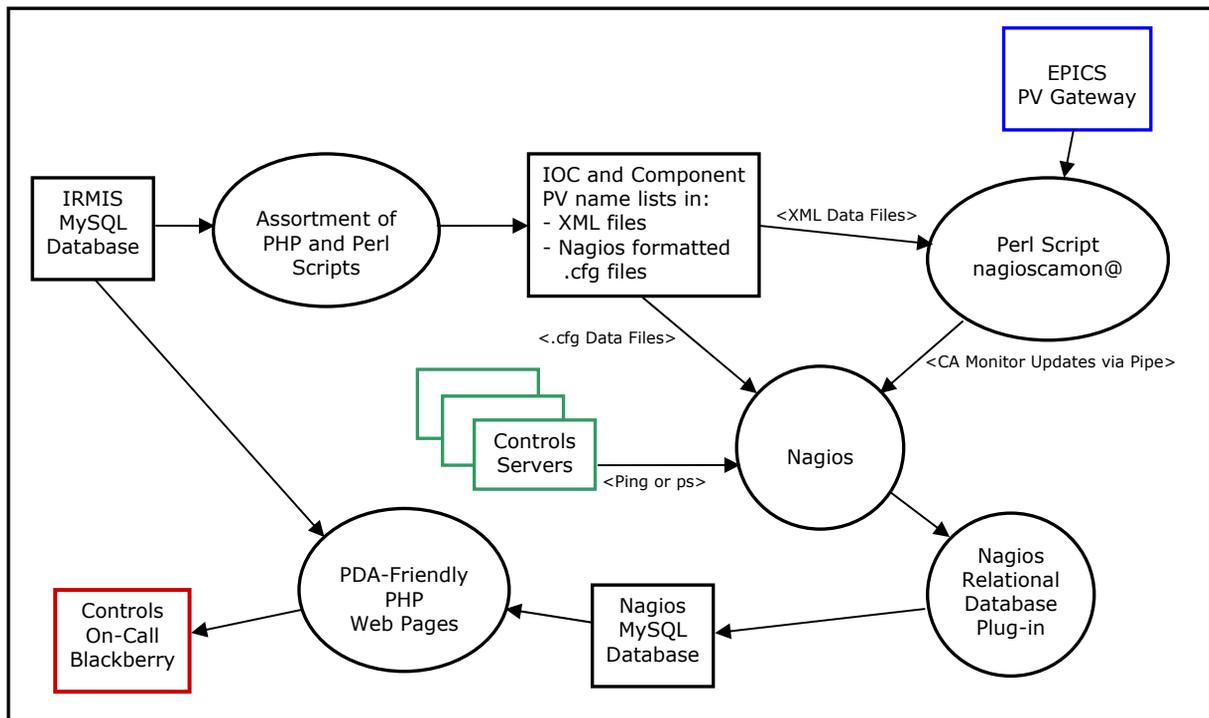


Figure 3: IMS process data flow diagram.