

Web interface for online ROOT and DAQ applications

Joern Adamczewski-Musch, Sergei Linev, GSI, Darmstadt, Germany Bertrand Bellenot, CERN, Geneva, Switzerland



DABC is a general-purpose DAQ framework, developed in GSI since 2007 and used in different experiments. Because it is used together with many other components (like slow control systems, other DAQ systems, online analysis), DABC requires a simple and flexible user interface for monitoring and control.

A specialized web server, based on embeddable Civetweb server, was implemented in DABC and ROOT. Such server can deliver data directly from running applications to web browser, where JavaScript with JSRootIO is used for implementing of interactive web graphics. JSRootIO makes it possible to display different ROOT objects in interactive web pages. To increase usability and flexibility of JSRootIO, several important improvements were done. As a result, any ROOT-based application with minimal efforts can be equipped with a web server, providing direct access to all registered objects. Alternatively, with FastCGI application can be easily integrated with standard web servers





The main goal of the JSRootIO project is to display ROOT objects like histograms or graphs in a core functionality of the code is revealed bv the name – JavaScript ROOT Input/Output. This provides the functionality to decode data from binary ROOT and create JavaScript Using interactive JavaScript web graphics, such objects can be displayed in most browsers like modern web Microsoft IE, Mozilla Firefox, Google Chrome, Opera. Advantage of JSRootIO approach - one could use JavaScript and HTML code together with any web server like Apache or IIS without any special requirements to the server. It is sufficient to put JSRootIO scripts and ROOT files on the web server and to provide a correct HTTP address on the main HTML page – the web server is used as simple file server.



GSI

like Apache or lighttpd.

Using a flexible plugin mechanism, DABC is able to handle data from other frameworks and display such data in a web browser. At the moment ROOT, Go4 (GSI ROOT-based analysis framework), MBS (GSI DAQ system), FESA (CERN/GSI accelerator control system), EPICS and DIM are supported. For instance, without modifications any Go4-based analysis now can be monitored and controlled via web interface. Or status information from many MBS nodes can be displayed in a web browser.

Using an agent/master approach, DABC offers a way to monitor and control complex systems. Many DABC agents can gather information from different sources like DAQ, slow-control system, or online analysis. A web server, running on a DABC master node, provides transparent access to data from all these agents. As a result, a unified user interface for distributed heterogeneous systems can be build.







MBS is a well-established DAQ system at GSI that has in many experiments for 20 years. A been used *mbs::Monitor* class has been implemented in DABC, which can acquire and display statistic information from running MBS node, e.g. event rate, data rate, and file storage rate. All these values are published and can be observed in the web browser. With most recent MBS version 6.3 such worker also can acquire logging information from the MBS node and execute arbitrary commands on the MBS node.

Several instances of mbs::Monitor class can run in parallel. Thus full control and MBS monitoring of many nodes simultaneously is possible now via web interface. Moreover, same information can be obtained via special DABC socket channel, using provided DABC interfaces. Simple command-line tool is delivered



protocol and the hierarchical structures created by the workers. Each worker can publish its internal hierarchy, making it available for other DABC components. All elements of such global hierarchy can be accessed 16:16:00 16:16:10 16:16:20 16:16:30 16:16:40 via http::Server and HTTP protocol. JavaScript code has been developed to organize and display hierarchical structures in a "tree view"

Same kind of information can be received via special socket-based command channel and displayed in any other applications – here in Go4 GUI.

Go4 is ROOT-based analysis framework. Its concept The information from TRootSniffer consists in the separation of an analysis process class can be seamlessly integrated executing the Go4 data processing code, from another into the hierarchical DABC structures. process that optionally provides an asynchronous A special DABC plugin makes ROOT graphical user interface (GUI). In addition to the default objects accessible via *http::Server* of DABC. This opens up the possibility Go4 GUI that was implemented as Qt and ROOT to integrate ROOT-based applications graphics application, an alternative webserver-based into a larger system that is combined GUI for inspecting analysis objects has been implemented. Now any existing Go4 analysis without any modification can be monitored via web browser.

HTTP access to different kind of online applications is

The EPICS plug-in can read specified IOC records and publish them in the DABC web server. Obtained records can be optionally packed into binary buffers and delivered to an analysis process together with DAQ data. This significantly simplifies implementation of analysis code, because DAQ and slow-control data will come synchronized from a single DABC data source. Similar approach (reading of preconfigured list of records) was used in plugin for FESA -CERN/FAIR accelerator control system.

The name server of DIM system delivers a complete list of available records therefore corresponding DABC plugin could provide access to all such DIM records. Through DABC command interface one could also access and execute available DIM commands. Information from different slow-control systems can be combined together and provided to the users via unified interfaces.

where information is acquired on different nodes (agents), but monitoring and control can be performed via web interface on a central "master" node. Master node automatically collects hierarchy descriptions from all agents and provides combined global description to the clients (web browsers). Communication between master and agents is done by means of a TCP/IPbased socket protocol. When a browser sends an HTTP request to the webserver on the master node, the request is redirected to the agent, responsible for the specified element.

representation

An arbitrary number of agents can be connected to the master. A flexible master/agent communication protocol allows at any time stopping/braking an agent and starting it again. Also the master application can be restarted at any time without the need to restart agents. This gives flexibility to dynamically increase/decrease number of agents without reinitializing the complete system; a failure on a single node will not cause a system-wide error. Information from the master can be accessed via: http - web browser, wget or curl socket - command-line tool, DABC API, Go4 GUI

For example, one can build distributed system, which consist from DAQ and two analysis kinds, all running on different nodes. Information from all these tasks will be available via the web interface on the master node, which also can be used for control of the remote tasks.

provided.

by means of DABC.

With minimal efforts any existing ROOT application can be equipped with an HTTP server and monitored from any web browser.

JavaScript code for browsing and display of different objects kinds is implemented; JSRootIO is used for ROOT classes.

Any existing Go4 application can use web interface for monitoring.

Heterogeneous distributed systems could be steered via web interface using DABC software.

The code is available on:

ROOTDEV git repository: http://root.cern.ch/git/rootdev.git DABC subversion repository: https://subversion.gsi.de/dabc/trunk

