CONTROL SYSTEM STUDIO AND THE SNS RELATIONAL DATABASE

J. David Purcell, ORNL, Oak Ridge, TN 37830, U.S.A.
Kay Kasemir, ORNL, Oak Ridge, TN 37830, U.S.A.

Abstract

As the Spallation Neutron Source (SNS) Project moves towards its goal of high reliability, better tools are required to enable operators and users quick and reliable access to relevant data. SNS is taking advantage of its single relational database (RDB) and incorporating it into different plug-ins for use with Control System Studio (CSS). This paper describes some of the existing database related plug-ins along with the plans for future growth.

OVERVIEW OF CSS AND SNS RELATIONAL DATABASE

SNS RDB

SNS has deployed an ORACLE-based RDB. The RDB was developed to support many different aspects of the SNS project. This includes but is not limited to data structures that support project administration, equipment installation, SNS operations, project documentation and the SNS control system. Because the RDB spans many areas of the SNS project, it has become the central storage area for a vast amount of data and it is considered the main source for information and support data.

The control system "area" within the SNS RDB is the most developed. Many different types of data have been captured. The RDB contains beam line equipment support data, networking data, installation data, calibration data, machine setup data, machine protection system (MPS) data and input output controller (IOC) data. SNS also systematically parses the configuration files of the IOCs to capture the latest implementations. This gives users the ability to see current signal names and the associated support data. Using the power of the RDB, we can provide a data summary pertinent to anything related to the control system.

CSS

CSS[1] is based on the Eclipse Rich Client Platform and is the result of a continuing collaborative effort. CSS is an environment that allows the implementation of applications for use with control systems. Because these applications are implemented in one environment, they present a common look and feel to the user. CSS is designed to give facilities the ability to customize the functionality contained within the environment by choosing the various applications or plug-ins that are included. CSS is developed using the Eclipse development environment and is written in Java.

CURRENT PLUG-INS

At SNS, two significant plug-ins have been developed that use the RDB. These plug-ins rely on the implementation of CSS core utility plug-ins and specifically use the RDB utility plug-in for connecting to the SNS database. Because of the integration with CSS core, these plug-ins adapt to the look, feel, and the functionality of the entire environment.

PV Utility

The SNS RDB is set up to track the relationship between deployed equipment and the signals that are associated with them. The process variable (PV) utility is designed to show that relationship to the user.

It has been designed to give users multiple avenues to access this information. Filtering can be performed to find a specific device and then see what signals are produced by that device.

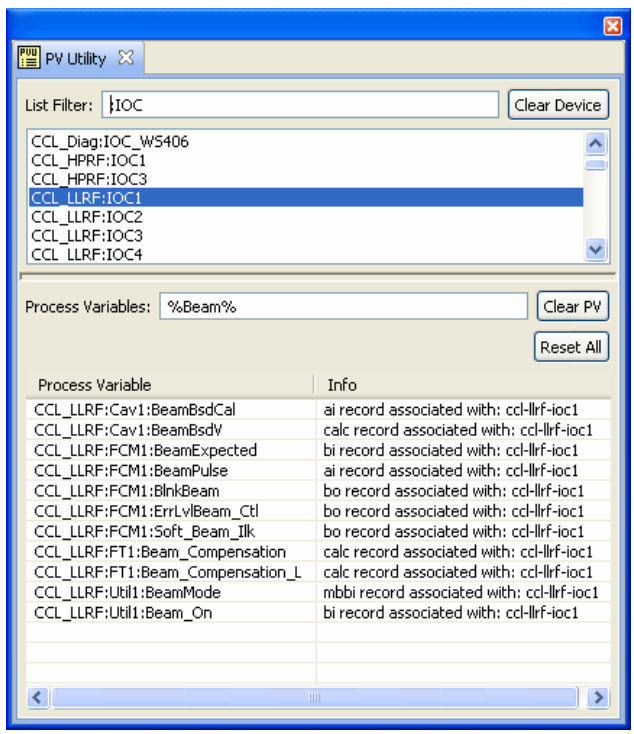

Figure 1: PV Utility

The utility can also take a signal and determine what device is controlling it. This is useful when this information is not intuitive and only can be found using a tracking system like RDB.

Figure 1 shows the utility along with some user input. The user has requested a list of devices that contain ':IOC'. They have selected 'CCL_LLRF:IOC1' and asked to see the process variables associated with this IOC that contain 'Beam'. The percent character is included as wild cards specifically needed for a search of the SNS RDB. The use of percent characters in the process variable text box and not for the List Filter is done as functionality specific to SNS. This is explained further in the section on Data Source Independence.

Rack View Utility

The RDB is also set up to manage the equipment housed within the SNS rack enclosures. The Rack View utility is a plug-in that allows a user to find a rack and display its contents. The utility gives a standard table list of the contents and also displays a real-time image of the layout.

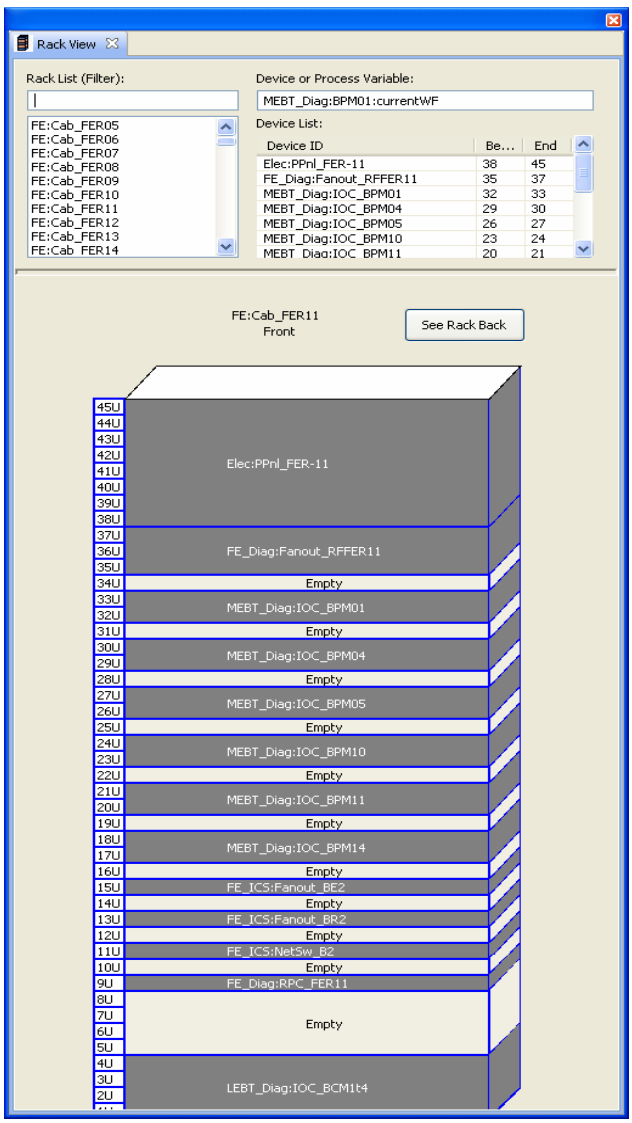

Figure 2: Rack View Utility

Users can look directly for a rack by scrolling through the complete list, or a filter is available to reduce the number of racks contained in the rack list. Also, with the relationships in the RDB between process variables and equipment, users can use a device name or a signal name to directly produce the rack profile. The utility also allows users to look at equipment stored in the rear of the rack.

In Figure 2, the user has supplied the signal "MEBT_Diag:BPM01:currentWF" to the utility. The utility has found the IOC that supplies that signal and produced the rack profile for the rack, "FE:Cab_FER11," which holds the IOC.

Object Contributions

Both the PV and the rack utilities have been designed as extensions that can be used with other CSS plug-ins. CSS uses the PV as an object. Like many other CSS utilities, the PV and rack view utilities also contribute to the PV object. This object contribution enables the utilities to appear in context menus only when the PV object is selected. And the context menu of any PV in any CSS tool lists all other PV-aware tools. If the object selected is not a PV type, these utilities are not available for selection. To the end user this means that after locating a PV in the PV utility, it's very easy to then start a tool to view the live data of that PV. Conversely it is as easy to start these tools from other plug-ins.

The utilities are also designed to accept the PV via a drag and drop.

COMPARISONS TO OTHER SNS TOOLS

SNS has implemented many different tools that report and manage data contained in the SNS RDB. These tools have been independently developed and there isn't a commonality among them.

SNS implemented a Java EPICS RDB Interface (JERI) as a tool that gives users a front end to the relational database. Included in JERI is some functionality needed by SNS for operations and machine protection. JERI does not, however, have a number of functions currently available in CSS. Channel access connectivity is an important function not found in JERI. Currently there is no longer support for the JERI tool. Although it is written in Java, maintenance and development of future tools within JERI has been hindered. The collaborative effort and available plug-ins make CSS a better overall tool for SNS and the JERI functions that are required will be moved in to the CSS environment.

A second tool allowing users access to the SNS RDB has been web browsers. Web pages have been developed for a multitude of functions. These pages are primarily used for reporting but have also been created to allow data manipulation. Recently SNS has adopted the use of the Web Channel Access Plug-in (WebCA) [2]. This plug-in gives the web browser the ability to be a channel access client. Combining the channel access client and the interface with the SNS RDB in a web page makes the web

browser a powerful tool. Many of the plug-in functions that are available in CSS can also be done using a web page. This is especially true of tools using the RDB. The use of style sheets and the new Channel Access Markup Language (CAML) [3] will make collaboration easier and give users a common look and feel. However, adapting code to be independent of the facility has not been implemented. Also, many aspects of the WebCA plug-in need testing and use is limited in terms of compatible browsers. CSS is platform independent and takes advantage of object contributions. Web page functionality like that found in the context menus of CSS through object contribution hasn't been developed yet. And, the use of CSS, allows site independence or the ability to package the plug-ins of CSS as appropriate for any facility.

DATA SOURCE INDEPENDENCE

The PV and rack view utility plug-ins developed for use at SNS have been created so that they are not dependant on the source of the data that is supplied. These plug-ins are interfaces that allow site specific support plug-ins to supply the data. These interfaces expect data to be supplied in a defined format but do not define the data source. This means that sites can implement the functionality of the plug-ins and supply data acquired from the source that is best for them. To enable independence, the use of wildcard characters was eliminated from within the utilities code and wildcards must be supplied by the user as part of their input.

The PV Utility depends on an array of strings that represent the control system objects that are related to process variables. The second requirement on an implementation is to supply a structure that contains the process variable as a string and a second string containing a description. This structure has to be returned to the interface as an array when passed either the control system object or the process variable filter.

The Rack View Utility requires an array of rack names. When supplied with a rack name or a string representing a process signal or rack device, the implementation must supply the interface with both the rack name and the rack height. The interface also expects a structure defined as a string indicating device, integer showing beginning position in rack, integer showing ending rack position, string indicating device type, and a string indicating front or back of rack.

CONCLUSION AND FUTURE PLANS

The CSS environment is an ideal place for plug-ins that take advantage of a control system's RDB. CSS has sufficient tools currently available and will only be made better with additional plug-in options.

At SNS the control system area of the RDB is the basis for many tools that exist to help different users. These current tools can easily be incorporated into CSS. Moving needed applications will eliminate the need for upkeep of the unused portion of different tools and the different programming languages used with them.

The SNS RDB contains much of the same data that is captured within data management systems of other facilities. Other sites can also use the tools. And SNS can take advantage of the tools developed by other facilities.

CSS is currently centered on the process variable. With the data found in the SNS RDB and data stores of other facilities, the CSS toolset can be expanded to also include a higher, broader view. A future step at SNS will be to take advantage of the data related to the hardware. With this data, tools will be developed to relate a variety of information including device configuration, cabling, and positioning. The RDB can also allow CSS to link users to documentation associated with PVs and hardware.

REFERENCES

[1] Matthias Clausen, "CSS Intro," EPICS Collaboration, Knoxville, TN, USA, October 2007, http://neutrons.ornl.gov/workshops/epics2007/index.shtml.

[2] Matej Sekoranja, "Java CA Client Library – Update," EPICS Collaboration, Hamburg Germany, April 2007, http://epics.desy.de/content/e2/e127/e138/index_eng.html.

[3] Tom Pelaia, "CAML Web Based Synoptic Displays," EPICS Collaboration, Padua Italy, October 2008, http://pcapac-workshop.org/index.php.