

# DATABASE APPLICATIONS DEVELOPMENT OF THE TPS CONTROL SYSTEM

Y. S. Cheng, Jenny Chen, C. Y. Liao, C. H. Huang, P. C. Chiu, Y. T. Chang, K. T. Hsu  
National Synchrotron Radiation Research Center, Hsinchu 30076, Taiwan

## Abstract

The control system had been established for the new 3 GeV synchrotron light source (Taiwan Photon Source, TPS) which was successful to commission at December 2014. Various control system platforms with the EPICS framework had been implemented and commissioned. The relational database (RDB) has been set up for some of the TPS control system applications used. The EPICS data archive systems are necessary to be built to record various machine parameters and status information into the RDB for long time logging. The specific applications have been developed to analyze the archived data which retrieved from the RDB. One EPICS alarm system is necessary to be set up to monitor sub-system status and record detail information into the RDB if the problem happened. Some Web-based applications with RDB have been gradually created to show the TPS machine status related information. The efforts are described at this paper.

## INTRODUCTION

The TPS is a new high brightness synchrotron light source which constructed at the National Synchrotron Radiation Research Center (NSRRC) in Taiwan. It consists of a 150 MeV electron linac, a booster synchrotron, a 3 GeV storage ring, and experimental beam lines. Civil construction had started from February 2010. The construction works had been finished in half of 2013. Accelerator system installation and integration had been started in later 2013. The control system environment was ready in half of 2014 to support final subsystem integration test and commissioning without beam. Commissioning with beam was successful at December 2014.

The EPICS (Experimental Physics and Industrial Control System) [1] is a set of open source software, libraries and applications developed collaboratively and used to create distributed soft real-time control systems for scientific instruments such as the particle accelerators. Many facilities have good practical experiences for the EPICS and adopt it as the accelerator control systems. Many resources and supports are available as well as numerous applications have been developed.

The TPS Control system is also based on the EPICS framework [2]. The EPICS toolkit provides standard tools for display creation, archiving, alarm handling, etc. The big success of EPICS is based on the definition of a standard IOC (Input Output Controller) structure together with an extensive library of driver software for a wide range of I/O cards. The EPICS toolkits which have

various functionalities are employed to monitor and to control accelerator system.

The Java-based EPICS data archive system was set up for the TPS project. The Relational Database (RDB) has been used as the data storage mechanism for recording historic EPICS data [3]. The various operation interfaces for browsing archived data have been developed. Taking the performance and redundancy into considerations, the storage servers and database table structures are tuned up relatively.

The Java-based alarm system has been developed as the alarm handler for the TPS control system. A distributed alarm system monitors the alarms in a control system and helps operators to make right decisions and actions in the shortest time. Moreover the historic alarm messages are logged into the RDB system, and the clients can use Java-based graphical operation interface to query and effectively check which system is failure.

Web-based machine status broadcasting is convenient to use with Web browser. The PHP webpage has been created for the TPS machine status broadcasting. The trend data are retrieved from the archive database, and also show variations on the Web-based machine status broadcasting.

The efforts for implementing are summarized as followings.

## TPS EPICS DATA ARCHIVE SYSTEM

### Software Environment

For the installation and commissioning phases, one EPICS data archive system is necessary to be developed to record various accelerator parameters and machine status information for long time observation. The archive system of CS-Studio [4-5] had been implemented completely to be used as the TPS EPICS data archive system in earlier 2014. The archive engines take PVs data from EPICS IOCs via channel access, and stores them into the data storage.

The CS-Studio based archive system can store data in a relational database (RDB). It applies JDBC libraries [6] in some databases, such as MySQL, Oracle, PostgreSQL, and the archive system includes example database definition (DBD) files to create the required tables for the database dialects. The EnterpriseDB [7] (PostgreSQL) RDB have been used for the EPICS data archive system of TPS project. The PostgreSQL RDB is a good compromise and bigger table sizes. Both the PVs historic data and the Archive Engine configuration are stored in the same relational database. The engine configurations are imported from an XML file format into the database.

The software architecture of TPS EPICS data archive system is shown as Fig. 1. The storage system is separated into two databases which named RDB1 and RDB2. The RDB1 is mainly used to store into the EPICS historic data from EPICS IOCs via PV channel access. The RDB2 is synchronized to be a duplicate of the RDB1. If the RDB1 causes service halt, the RDB2 will do replication mode for restoring to the RDB1. The RDB2 provides higher reading priority to enhance transmission bandwidth for extracting queried historic archived data.

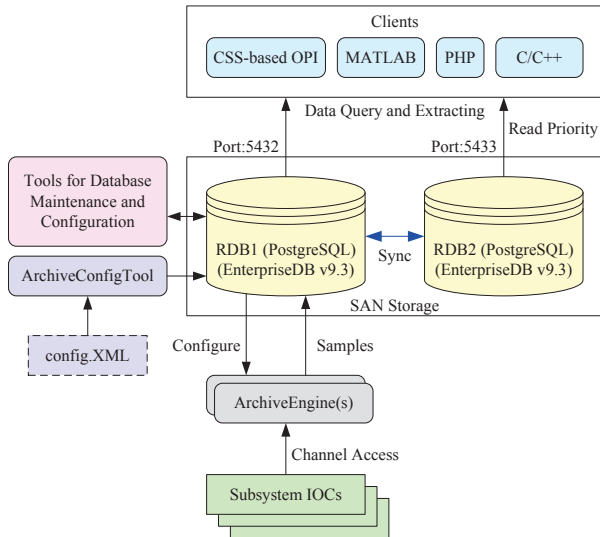


Figure 1: Software architecture of the TPS EPICS data archive system.

### Database Tuning Up for the TPS Archive System

To easily maintain and manage the archived data in the database, the sample data table has been separated into a lot of sub-tables by days. The new EPICS sampled data have been inserted into the specific sub-table which is according to the day in the year. If the older samples need to be deleted, the day-named table can be indicated and deleted. Each created sub-tables have been added the table index to enhance performance for data retrieving. Actually the test is that one week historic data (one channel-sample per second) need about 10~20 seconds to be queried out, extracting and plotting. Otherwise, there is about 1~2 minutes at least for retrieving one week historic data without creating the table indexes. It is effective to create table indexes to enhance performance for retrieval.

### CS-Studio Based Operation Interface

The clients can use the specific toolkits to retrieve the historic archived data from the PostgreSQL RDB. As a result, several operation interfaces have been created for difference purposes.

The Data Browser is a generic CS-Studio toolkit that combines Strip Tool and Archive Viewer functionality. It can display live samples as well as archived data in a plot, or export the data to files. Based on plugins for archive data sources, it can currently interface to the Channel Archiver and the EPICS Archive record. Each PV may

have multiple data sources: for example, the Data Browser can merge samples from an archive record for recent history with those from a midterm and long-term archive. It is convenient that the CS-Studio based OPI can be used in difference operation systems, such as Windows and Linux-based. For the TPS commissioning and operation phases, the graphical interface of machine status is necessary to be created for observing the historic variations for long time, such as beam current of booster and storage ring. The CS-Studio based graphical interface is shown at Fig. 2.

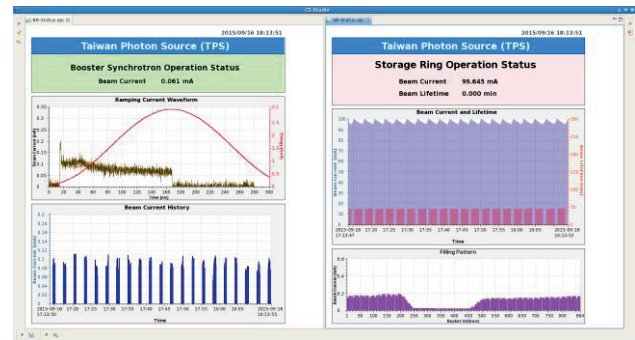


Figure 2: GUI of CSS-based archive data browser.

To easily observe the temperature and vacuum variations during baking process of in-vacuum insertion device, the specific graphical monitor interface was necessary to be created by use of the CS-Studio as shown in Fig. 3. This interface was displayed the latest 30 minutes historic temperature and vacuum data trend which retrieved from the RDB of archive system.

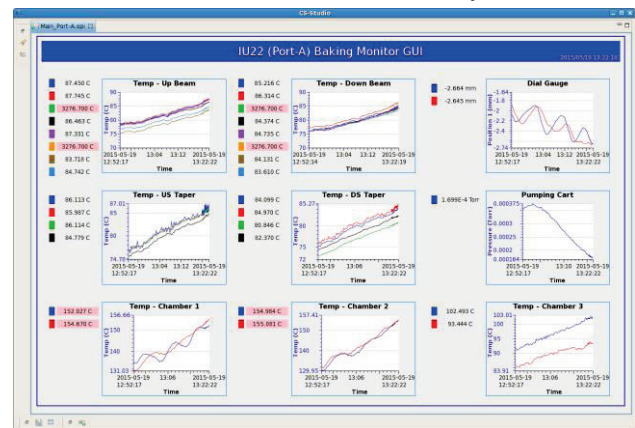


Figure 3: Archive data browsing GUI for ID baking.

### Miscellaneous Operation Interfaces

To special purposes, the specific toolkit, such as MATLAB [8] (with JDBC connection), is used to create the interface to retrieve the EPICS archived data from the RDB.

To investigate the beam loss and its distribution, a RedFET (radiation-sensing field-effect transistor) reader was implemented with an EPICS IOC. All of the RedFET threshold voltage values based on the EPICS PVs channel access can be recorded into the PostgreSQL RDB archive server for further off-line data processing. The archived

data can be retrieved by use of a form of graphical representation of the CS-Studio based data browser to observe the trend. The MATLAB toolkit has been used to analyze the RadFET threshold voltage archived data which retrieved from the RDB archive system directly [9]. The MATLAB toolkit for analyzing the RadFET threshold voltage archived data is shown as Fig. 4.

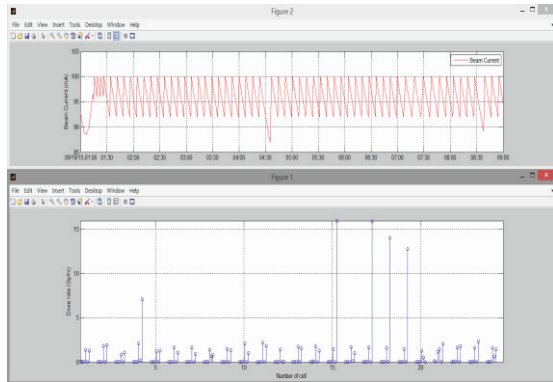


Figure 4: MATLAB interface for analyzing archived data.

### TPS ALARM SYSTEM

The “BEAST” (Best Ever Alarm System Toolkit) of CS-Studio with the MySQL RDB has been adopted as the alarm handler for the TPS as shown in Fig. 5. A distributed alarm system monitors the alarms in a control system and helps operators to make right decisions and actions in the shortest time. In the CS-Studio based alarm system, each alarm is supposed to be meaningful, requiring an operator action. An alarm is no status display that operators may ignore. Each alarm requires an operator to react because the control system cannot automatically resolve an issue.

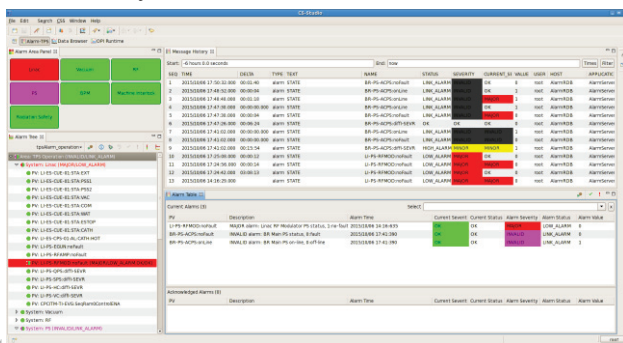


Figure 5: CS-Studio based graphical interface of TPS alarm system.

### WEB-BASED APPLICATIONS

The webpage has been created by using the PHP and AJAX (Asynchronous JavaScript and XML) technique for the TPS machine status broadcasting. The client users can use Web browsers of PCs or smartphones to watch the machine status immediately. The trend graph is one of necessary components to show the historic variations of beam current and beam lifetime on the TPS machine status webpage. The trend data of several hours are retrieved from the PostgreSQL database of TPS archive

system by the PHP program [10]. The TPS Web-based machine status broadcasting with trend graphs is shown as Fig. 6.

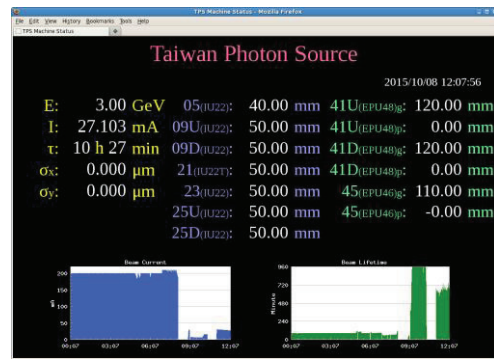


Figure 6: TPS Web-based machine status broadcasting.

The Olog [11] solution has been selected for the TPS electronic logbook. The progress of commissioning and operation information has been recorded into the MySQL RDB via the Olog by the commissioning team and operators. The Olog supports use on the Web browser and CS-Studio, and also supports search function and print function to copy data into the logbook for logging.

### SUMMARY

The RDB has been adopted as the data storage for the TPS archive system. The archived data are retrieved by use of the CS-Studio based interface. The archived data are also retrieved by specific toolkits for special purposes. The RDB alarm system has been built to help operators to make right decisions and actions in the shortest time. More database applications are in developing during the commissioning and operation phases.

### REFERENCES

- [1] EPICS website: <http://www.aps.anl.gov/epics/>
- [2] C. Y. Liao, et al., “Commissioning of the TPS Control System”, FRB3001, these proceedings, ICALEPCS’15, Melbourne, Australia (2015).
- [3] Y. S. Cheng, et al., “Implementation of the EPICS Data Archive System for the TPS Project”, THPEA049, Proceedings of IPAC’13, Shanghai, China (2013).
- [4] K. Kasemir, “Control System Studio Applications,” MOPB03, Proceedings of ICALEPCS’07, Knoxville, Tennessee, USA (2007).
- [5] K. Kasemir and G. Carcassi, Control System Studio Guide, 2012.
- [6] JDBC website: <http://docs.oracle.com/javase/6/docs/technotes/guides/jdbc/>
- [7] EnterpriseDB website: <http://www.enterprisedb.com>
- [8] MATLAB with JDBC connection website: [http://www.mathworks.com/help/database/ug/databas\\_e.html](http://www.mathworks.com/help/database/ug/databas_e.html)

- [9] C. H. Huang, et al., “Beam Loss Study of TLS Using RadFETs”, MOPTY072, Proceedings of the IPAC’15, Richmond, USA (2015).
- [10] PHP with PostgreSQL database website:  
<http://php.net/manual/en/book.pgsql.php>
- [11] Olog website: <https://github.com/Olog/logbook>