UPGRADE OF THE ISAC DEVICE DATABASE FROM PARADOX TO POSTGRESQL

R. Keitel, J. Richards, and E. Tikhomolov
TRIUMF, Vancouver, B.C., V6T 2A3, Canada

Abstract

The relational device database for the EPICS control system of the ISAC Radioactive Beam Facility at TRIUMF is at the centre of generating EPICS data bases, operator display panels, and verifying PLC interlocks against specifications. For historical reasons, this database is implemented using Paradox on PCs with shared access to a file server. Use of the database is thus limited to PC nodes with Paradox installed. In addition, the network turns into a bottleneck for database transactions. Therefore a project was started to replace Paradox by PostgreSQL running on a LINUX database server with a web-based user interface. For replacing the current client-side Paradox ObjectPAL code, server-side implementations using the Perl DBI, PHP, and Java servlets were evaluated. The results of these evaluations, the current state of the project, and the effort involved are reviewed.

INTRODUCTION

The EPICS based control system for the ISAC Radioactive Beam Facility at TRIUMF uses a relational data base (RDB) for storing information on all controlled devices. This database evolved from an earlier project [1] where it was mainly used to store device data in support a quality assurance (QA) program and generate QA documents as reports from the database. At ISAC, the scope for the database gradually broadened. Nearly all of the EPICS IOC runtime databases (IOC = input/output controller) are presently constructed from this RDB by generating intermediate schematic files in Capfast format, which are in turn automatically converted to the IOC format [2]. At present it is planned to increase the scope of the database further and provide control system information to a wider range of TRIUMF staff.

THE CURRENT SYSTEM

For historical reasons, the ISAC device database was implemented using the Paradox [3] RDB on PCs with shared access to a file server. The database engine and the form-based user interface runs on the user's PC and the database tables, forms, and application code libraries are stored on the file server. This situation is rather limiting for two reasons: First, any database query involves repeated network accesses and turns the network into a performance bottleneck. Second, each access-point to the database needs a copy of Paradox installed, which would make database access very expensive and maintenance-intensive for a larger user community.

THE NEW SYSTEM

Upgrade Goals

It was therefore decided to upgrade the existing system with the following goals

- Use an SQL compliant database (the majority of the Paradox system was implemented before Paradox supported SQL)
- Use an Open Source database, without precluding a later adoption of a commercial RDB such as Oracle
- Provide a web-based user interface to allow database access from any web-browser

Database

A short comparative evaluation of the Open Source database systems mySQL and PostgreSQL was performed early in the project. Limitations of mySQL in features such as transactions, roll-back, foreign keys, and database triggers led to the adoption of PostgreSQL for the ISAC device database.

The existing Paradox tables were exported to ASCII files and imported into PostgreSQL without any problems. As a next step, the existing database design was reviewed and upgraded. Extensive use was made of views, which were not available in the Paradox system. This allowed a better normalisation of the database tables. The use of database triggers led to some simplification of the application code.

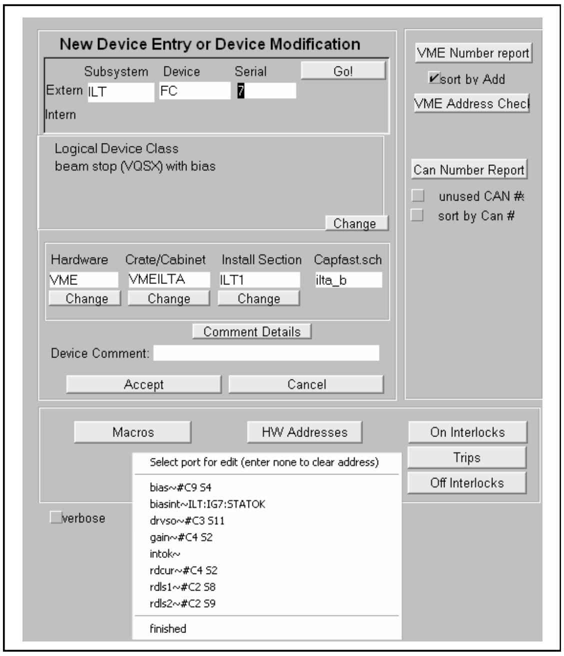
Web Interface

For the web-based user interface, implementations in PHP, Perl and Java (servlets and Java Server Pages) were explored in small sample applications. It was decided up-front not to use client-side scripting.

The following TRIUMF/ISAC context was considered in choosing between these three languages and designing the web interface:

- All of the database development, application code development, and web page generation is done by the same small group of control system developers. Therefore the separation of web page development and content generation was not considered to be important.

- The ISAC device database is quite small, consisting of 14 tables, which are common to all EPICS controls projects and an additional 11 tables that are project specific. The total number of records is less than 20000. The expected transaction volume is very low.
- The web pages needed to present data from the device data base and interact with the data base are very simple form based pages, which use text input fields, drop-down menus, and table displays.
- The ISAC web-site is LINUX hosted and uses the Apache web-server.



New Device Entry or Device Modification

	Subsystem	Device	Serial
Extern	ILT	FC	7
Intern			

Logical Device Class
beam stop (VQSX) with bias

Hardware	Crate/Cabinet	Install Section	Capfast.sch
VME	VMEILTA	ILT1	ilta_b

Device Comment:

☑ sort by Add
□ unused CAN #
□ sort by Can #
□ verbose

Select port for edit (enter none to clear address)

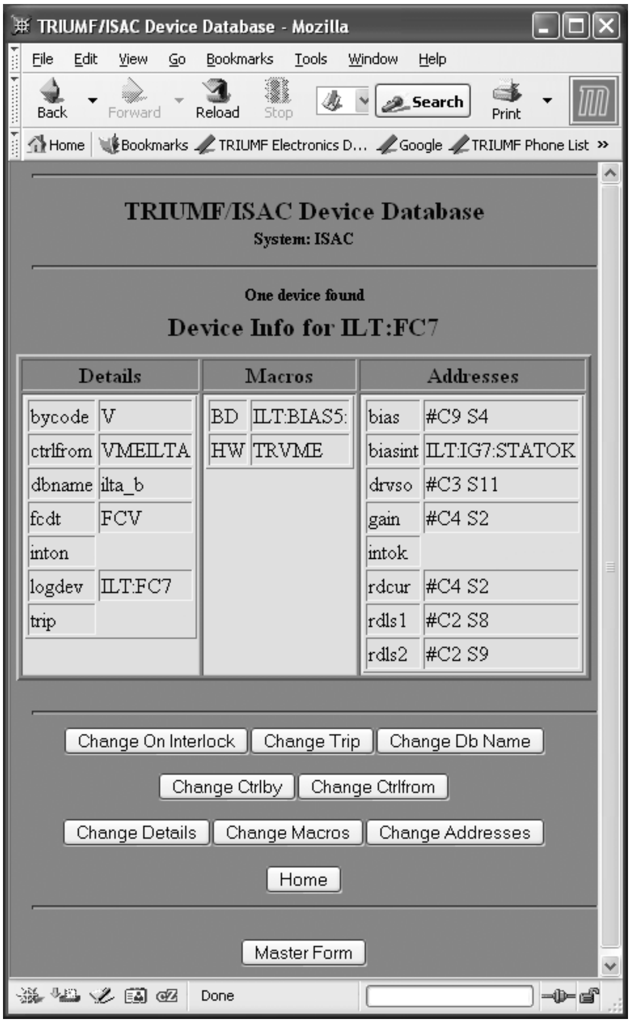
bias~#C9 S4
biasint~ILT:IG7:STATOK
drvso~#C3 S11
gain~#C4 S2
intok~
rdcur~#C4 S2
rdls1~#C2 S8
rdls2~#C2 S9

finished

Figure 1: Device entry form in the Paradox GUI

All ISAC programmers are familiar with the Perl language, which is the standard scripting language in the ISAC control system. From the example applications it was concluded that neither introducing PHP nor Java would provide any advantages in the above-mentioned context. In fact, using the Perl CGI module for generating the web pages produced the most easily readable code. Therefore Perl was chosen to build the new database application, the Perl CGI module is used to generate the web-interface, and the Perl DBI module provides the interface to the data base.

In developing the Perl scripts, programming techniques were used that allow the application scripts to run both as CGI scripts and also compiled under the Modperl module of the Apache webserver. This simplifies debugging of the web-interface in the CGI environment, but allows for a fast production environment.



TRIUMF/ISAC Device Database - Mozilla

TRIUMF/ISAC Device Database
System: ISAC

One device found
Device Info for ILT:FC7

Details		Macros		Addresses	
bycode	V	BD	ILT:BIAS5:	bias	#C9 S4
ctrlfrom	VMEILTA	HW	TRVME	biasint	ILT:IG7:STATOK
dbname	ilta_b			drvso	#C3 S11
fcdt	FCV			gain	#C4 S2
inton				intok	
logdev	ILT:FC7			rdcur	#C4 S2
trip				rdls1	#C2 S8
				rdls2	#C2 S9

Done

Figure 2: Device entry form in the web-browser GUI

The user interface of the Paradox application guided the user by extensive use of sequential pop-up windows. The new web-application, by contrast, attempts to minimise the number of web transactions by grouping as many form elements on a web page as is sensible in a given situation. Figure 1 shows a device data entry form of the old Paradox application. Figure 2 shows the corresponding web-page generated by the new perl scripts.

STATUS

The database redesign is complete. The device database application code in Paradox consisted of approximately 3500 ObjectPal code lines. At present, the functionality of 80% of this code has been ported to Perl scripts, which contain a total of 3200 lines of code. All user interface functions relating to device entry, device modification, and generation of CAPFAST schematics are implemented.

Testing is underway with an exported copy of the current ISAC device database. Another couple of weeks of implementing the reporting functionality of the old system are anticipated before the Paradox system can be retired and the Web/PostgreSQL based system takes its place.

Finally design work has been started to integrate the device data base with the control system asseting and fault report data bases.

REFERENCES

[1] R. Keitel, D. Dale, and B. Milton, “The TR30 Control System, a Case for Off-the-shelf Software”, Cyc92, Vancouver , July 1992.

[2] R.Keitel, “Generating EPICS IOC Data Bases from a Relational Data Base – A Different Approach” , ICALEPCS01, San Jose, November 2001.

[3] Paradox is a trademark of Corel Corp., Kanata, Canada