TDCT- A CONFIGURATION TOOL FOR EPICS RUNTIME DATABASES

R.Keitel , TRIUMF, 4004 Wesbrook Mall, Vancouver, BC, V6T 2A3, Canada

Abstract

The EPICS runtime data bases for the ISAC Radioactive Beam Facility at TRIUMF are developed using Capfast[1]. Capfast is a tool for drawing electronics schematics. EPICS runtime functionality is configured using an EPICS function block symbol library. A translation tool, which was developed in the EPICS collaboration generates the runtime databases from the Capfast schematics. Capfast provides excellent hierarchy support, but has a cumbersome user interface, major limitations and is not fully EPICS aware. As TRIUMF has a big investment in its schematic hierarchy, a compatible tool TDCT was developed, which overcomes the limitations of Capfast. TDCT was developed in Java, is fully EPICS aware and supports the whole development process from schematic drawing to runtime database generation.

EPICS DATABASES

In an EPICS [2] based control system, the core functionality is provided by software running in Input-Output-Computers (IOCs). The control system designer does not create this software by conventional programming techniques. Instead, EPICS provides an extensible set of function blocks (EPICS records), techniques to link them into execution units, and a framework to execute these units based on timed or interrupt-driven schedules. The control system programmer's task is to instantiate the EPICS records with properties (record fields) and define paths for data and control flow between records (input and output links). The resulting record collections (EPICS data bases) are loaded on IOC startup and translated by the EPICS IOC core software framework into data tables in IOC memory. Therefore, in an EPICS system, the traditional control system programming tasks are replaced by designing and configuring EPICS databases.

EPICS databases are ASCII files and can in principle be created and maintained with a plain text editor. Database configuration is, however, made much easier with configuration tools, which are "EPICS aware", i.e. generate the record syntax and provide selection menus for record properties. Additional benefits are gained from database configuration tools which provide a graphical representation of the controls functionality.

EPICS DATABASES AT TRIUMF

Work on the control system for the ISAC radioactive beam facility at TRIUMF was started in 1996. At that time, three database configuration tools were available:

- DCT – fully EPICS aware, not graphical
- GDCT – fully EPICS aware, graphical, but no support for hierarchy
- Capfast™ – partially EPICS aware, graphical, excellent hierarchy support.

As we wanted to implement the ISAC control system based on an object-like device model with reusable components, hierarchy support was a required feature and we chose Capfast™ as our database configuration tool.

Generating EPICS Databases with Capfast™

Capfast™ is an electronics design package. EPICS databases are configured by drawing the equivalent of an electronics schematic using a "parts" library for EPICs records. Capfast™ provides unlimited hierarchy support by allowing the user to design custom symbols, which represent the next lower level schematic. Individual custom symbols can be designed to have a meaningful appearance. This provides visual clues for the interpretation of a schematic's functionality.

For configuring EPICS databases, three Capfast™ components are used:

1. the schematics editor
2. the symbol editor
3. the EDIF* translator

After a top-level schematic and all required component symbols and schematics are developed with the editors, the translator converts these files into standard Electronics Design Interchange Format. The EPICS collaboration provides an EDIF-to-EPICS database translation tool e2db for the final conversion step.

ISAC Control System

The ISAC [3] control system group has used Capfast™ to successfully implement the object-like device model [4]. Top-level Capfast™ schematics are instantiated from a relational device database. Presently the system controls ~4500 devices. This is achieved using ~1250 schematic files, 350 of which are re-usable components. The IOCs are loaded with 640 EPICS databases containing 130000 EPICS records.

TDCT

Why a New Tool?

Capfast™ originated as a UNIX-workstation tool. Its user interface is unusual, differs between Windows and other platforms and does not match the expectations of developers accustomed to the modern desktop environment. The lack of full EPICS awareness makes the database configuration process more error-prone than necessary. In addition, some licensing and support issues

* Electronics Design Interchange Format

were encountered, which led to uneasiness about our exclusive dependency on this product.

Recently more EPICS sites use VDCT, a Java-based visual database configuration tool developed under contract by Cosylab [4]. VDCT surfaced in 2004, is fully EPICS aware and has recently added hierarchy support. However, VDCT has no provisions for customizing the shapes of symbols. This was deemed unacceptable for the use at ISAC. Therfore development was started on the TRIUMF Database Development Tool (TDCT). The working title YADCT "Yet Another …" has been dropped.

Development Goals

The following requirements were specified for TDCT:

1. It should support the full work flow of EPICS database generation, i.e. combine the functionality of both the Capfast™schematics editor, symbol editor and be able to produce the EPICS database file ready for loading into the IOC
2. It should be compatible with the Capfast™ data file format, both on input and output.
3. It should be EPICS aware
4. It should allow simultaneous work on multiple schematics and symbols
5. It should allow copy/paste operations between schematics
6. EPICS databases should be generated in a format that allows comparison between Capfast™ and TDCT builds.
7. It should work under Windows and Linux operating systems.
8. It should provide schematic check functions.

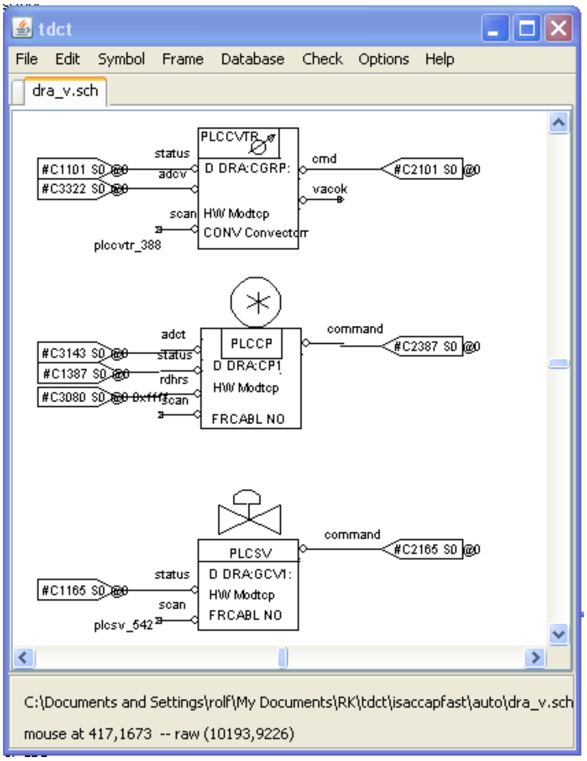
Implementation Status

TDCT was developed as a stand-alone Java application using Plain Old Java Objects (POJOs) and a JFC/Swing graphical user interface. Some Capfast™ features not used at TRIUMF, such as symbol rotation and bus-wires, were not carried over to TDCT. The requirements 1 to 7 stated in the previous section have been implemented. For requirement 8, the following checks are presently implemented:

- Check wire connections for incompatible ports.
- Check for invalid record names

Figure 1 shows part of a top-level TDCT schematic where devices are instantiated. Different device types are clearly discernible by shape. Figure 2 shows a device schematic with components for interlock latching, time-out and on-off control. Figure 3 shows several watchdog devices. All three screen dumps demonstrate the value of discernible shapes which support the pattern recognition abilities of the human brain.

For comprehensive testing of the EPICS database generation, a batch-building function was incorporated into TDCT and all 640 databases used at TRIUMF were



C:\Documents and Settings\rolf\My Documents\RK\tdct\isaccapfast\auto\dra_v.sch

mouse at 417,1673 -- raw (10193,9226)

Figure 1: Top level schematics showing different device types.

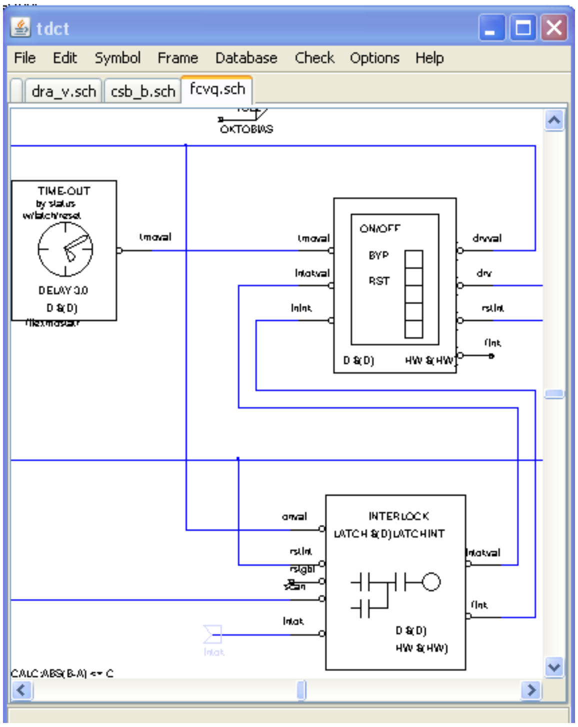
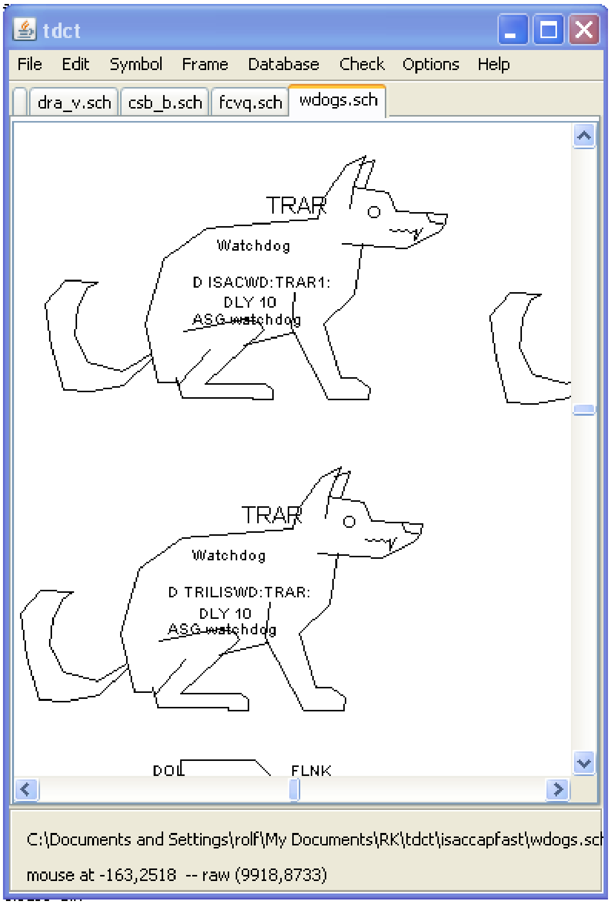


Figure 2: Part of a device schematic showing components for timeout, interlock latching and on-off control.

built using both TDCT and the Capfast™ tool chain. A Perl tool for batch-comparison of databases was developed and used to verify the TDCT builds against the Capfast™ builds.

TDCT has been used on recent projects in parallel with Capfast™ and a gradual migration to TDCT is foreseen in the near future.



```
C:\Documents and Settings\rolf\My Documents\RK\tdct\isaccapfast\wdogs.sc
mouse at -163,2518 -- raw (9918,8733)
```

Figure 3: part of a schematic containing watchdog devices.

SUMMARY AND OUTLOOK

With TDCT, the ISAC controls developers have shed their dependency on Capfast™ and overcome its limitations in the EPICS environment. Future development plans for TDCT are presently on hold due to the resource limits of the group. The author would like to thank R.Nussbaumer for his help with testing the application, providing a database comparison script and many helpful suggestions for improving TDCT.

REFERENCES

[1] Phase Three Logic, Beaverton, Oregon, http://www.phase3.com/.

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

[3] http://www.triumf.ca/about-triumf/triumf-faq/isac-backgrounder.

[4] http://www.cosylab.com/solutions/particle_accelerators/VisualDCT.