

## EPIC(S) DEVELOPMENTS OF VISUAL DCT

G. Jansa\*, I. List, G. Pajor, R. Sabjan, M. Sekoranja, I. Verstovsek, Cosylab, Slovenia

### Abstract

Visual Database Configuration Tool (Visual DCT) became the most advanced and popular EPICS database configuration tool. Together with the recently implemented and planned features it offers a complete tool to create, edit and debug EPICS databases. In this article we will introduce some important features, one of which is the spread sheet view for templates. This view is very useful for designing databases with a large number of channels with little or no links between them. An important addition is also substitution file editor. Nowadays the substitution files can be very hard to maintain due to their lack of transparency as text files. This editor will introduce some nice features which will make editing the substitution files an easy rather than a painful job. In the article we are also pointing out some best practice designs that became possible with the introduction of the hierarchy mechanism which mimics object orienting programming.

### OVERVIEW

VisualDCT is designed to create and maintain EPICS record instance database (.db) files. In order for VisualDCT to execute properly, a database definition (.dbd) file has to be provided which contains the specifications for the various record and device types that they intend to reference in any record instance database (.db) file to be created by VisualDCT. Once a database definition (.dbd) file has been specified, records can be created, copied, renamed, etc. using the various tools provided by the VisualDCT. As the user interacts with the various VisualDCT windows, selections, and data entry fields, the results of these interactions are displayed on the screen. Revisions and data entry updates of record instance data displayed on the screen do not replace previously stored record instance data until the user saves currently modified record instance database (.db) file.

In order to run VisualDCT, Java Runtime Environment 2 [4] is needed. VisualDCT is distributed as a Java ARchive package (.jar file), so there is only one file in the distribution.

VisualDCT became the most popular database configuration tool among EPICS community. The ever increasing number of users has also a very positive effect on the VisualDCT itself. Extensive use provides bug information and new feature requests. With new features under development, VisualDCT goes on a voyage beyond EPICS databases. It will offer tools to create some very important parts of EPICS control system such as Alarm handler and EPICS Channels Archiver.

### SPREADSHEET VIEW AND EDITOR

Many EPICS databases contain lots of records and have very little logic in it. It can be very unpleasant to upkeep and edit such a database with an ordinary VisualDCT window not to mention ordinary text files. If one field needs to be modified in every single record user must go through all records and click on them to get property inspector open and actually modify the field.

Therefore we will introduce new view of records – spreadsheet style. Rather graphically representing records, it will display them on a table, one record per row. User will be able to configure which fields should be seen in a table.

Spreadsheet view will consist of tabbed pane, one tab for each record type. All cells in the table will be editable using property inspector functionality. It will even be possible to select all records on a single table showing only the fields that are common to all records or for example select only all output records. This will enable user to quickly find errors. Ability to compare the database directly to the signal list will also be available.

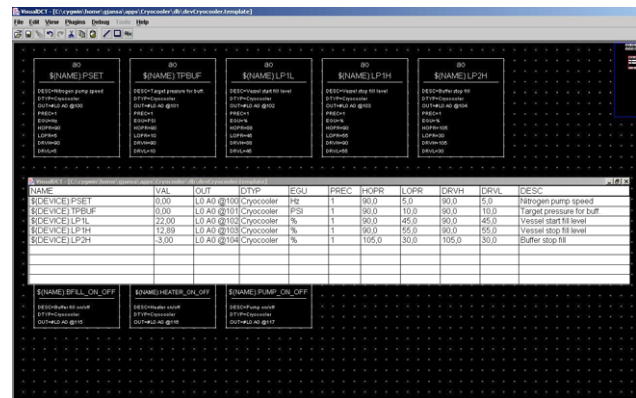


Figure 1: Spreadsheet view and editor.

### ALARM HANDLER CONFIGURATIONS PLUG-IN

EPICS Alarm handler requires text configuration files to run. These files contain EPICS channels structured in a hierarchical manner.

Configuration can be done with only one file holding all the channels user wants to observe or the structure can also be achieved with one root file which includes many other configuration files. These included files can have their own structure with many more included files, thus achieving a very complicated structure. Labs all across the world spend a lot of effort developing systems that generate these files which are very specific to certain institute and are therefore hardly adopted by others.

\*gasper.jansa@cosylab.com

Having in mind simple and easy procedure for generating configuration files and trying to import the goodies from VisualDCT the idea of alarm handler configuration plug-in arose. This plug-in will offer a very simple and straightforward way of configuring these files.



Figure 2: Alarm handler configuration plug-in.

The main screen is divided into an area showing a tree, representing structure of groups and channels already defined in imported configuration file. User can expand each group or channel to define properties (ALIAS, COMMAND, GUIDANCE ...) or in the case of a group display also subgroups. On the right side of the screen is a list of records. These are obtained from the currently opened template file in the main VisualDCT window. Channels and groups can be dragged and dropped and therefore arbitrary configuration can be achieved without typing in the common text files.

In future, support for grouping alarm handler configuration files with INCLUDE command will be supported. It will also be possible to instantiate groups with substitution files and thus minimize the extra work.

## CHANNEL ARCHIVER CONFIGURATION PLUG-IN

EPICS Channel Archiver is used to archive EPICS channels. Similar to alarm handler, archiver also needs configuration file with names of channels to archive. This file comes in a form of xml and is also less complicated than alarm handler configuration file(s) since there cannot be any sub grouping i.e it is flat and it is only one.

Archiver plug-in looks similar to Alarm handler plug-in. On the left side is tree containing xml root attributes which are editable. Groups are displayed similar to Explorer's directories in tree. They can be expanded to show channels they contain and those can be further expanded to list their properties. Records can be easily dragged and dropped from group to group. On the right side of the main window is a list of records that comprises current template file. They can be selected and moved to selected group in tree.

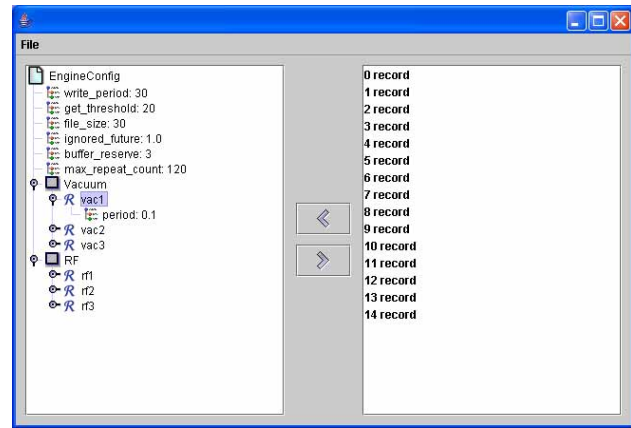


Figure 3: Channel Archiver configuration plug-in.

Further developments should include possibility to use substitutions file upon group of channels. Hence user would obtain many instance of the device that is represented with template file. This tool will simplify the extra work that needs to be done to generate archiver configuration files manually.

## SUBSTITUTION FILE EDITOR

EPICS database consist of template and substitution file. Former contains records with macros and the latter hold the definitions for these macros. Substitution files with huge amount of macros have tendency to become hard to maintain.

Substitution file editor will offer syntax checks and a comparison between number of macros and the number of their substitutes. These features will enable user to focus more on actual values of macro substitutes rather on correctness of substitution file.

## HIERARCHY SUPPORT

Some other database configuration tools also included support for some sort of hierarchical design. A new hierarchy design has been used for VisualDCT, which will be considered at integration of hierarchy support into EPICS base. VisualDCT remains backward compatible as the hierarchy can be flattened into normal EPICS database. The current EPICS template substitution mechanism is very restricted in its capabilities. It requires two input files (.template and .substitutions) that have radically different syntaxes, and it only allows macros to be passed downwards into a template instance. Hierarchical templates as implemented by VisualDCT must allow macro values to be passed into the template instance (giving values for fields within the expanded template), and values to be exported from the template instance to the higher level (usually the destination field name for a link in a record defined in the higher level .vdb file). Macros are defined in an **expand** statement and pass information into a template; ports are a kind of macro defined in a **template** statement that pass information upwards out of a template instance to their calling database.

Templates have special representation in VisualDCT. They are displayed almost like records. Inside the rectangle, template's macro values are shown and the user can access template's ports as any other link fields of a record. The user can also use SHIFT-click technique on it to open the template itself:

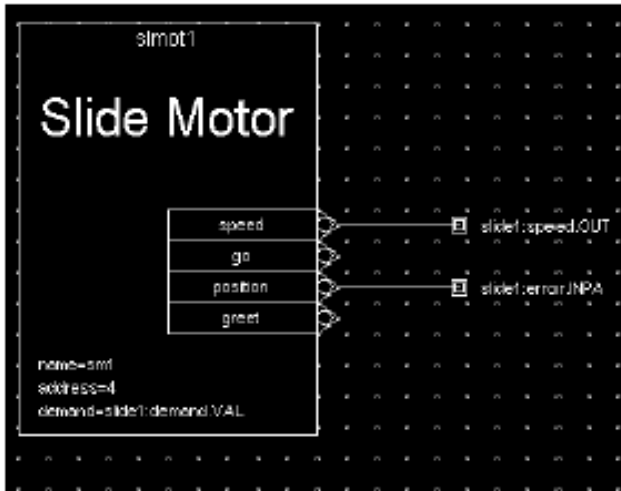


Figure 4: VisualDCT has special graphical representation for template instances

### *Database Flattening*

In order to make database useful for EPICS all the templates have to be instantiated (e.g. the database needs to be flattened). VisualDCT is the first tool that is capable of flattening this new database hierarchy syntax (capFast designs have always had similar hierarchical template capabilities). The flattening process involves expanding all templates and replacing the macro and port macro variables with their strings. If a macro name is found that has no definition within its scope, it will be left exactly as it was found, which allows load-time macros to be used.

## CONCLUSION

Taking into account all aforementioned features, we can justifiably state that EPICS has obtained a very powerful development tool, beating the competition in capability and user friendliness. We must not forget that a large proportion of EPICS community still uses text editors to configure their databases and all sorts of configuration files. With extensive use of shell scripts and manual editing, it gets very hard to maintain a large control system. So, though EPICS is very scalable, it gets very messy over a period of time and drives a lot of human resources to maintain all the databases.

We believe that VisualDCT can help tremendously in this area. Even the costs of initial database and configuration files creating are cut, but the biggest gain may occur after a month or a year, when something needs to be changed. This is proved by the use of VisualDCT around the world.

## ACKNOWLEDGEMENTS

We would like to express out gratitude to all people involved on Visual DCT project, especially the funders of certain packages.

Our thanks go to APS, Diamond and SNS for funding the development of hierarchical features in Visual DCT.

Once again we would like to thank Diamond for supporting the development of Alarm handler and Channel Archiver plug-in. This is truly the most important improvement and a large step forward.

## REFERENCES

- [1] <http://visualdct.cosylab.com>
- [2] <http://www.aps.anl.gov/epics>
- [3] <http://www.aps.anl.gov/epics/extensions/msi/index.php>
- [4] <http://www.aps.anl.gov/epics/extensions/alh/index.php>
- [5] <http://ics-eb1.sns.ornl.gov/~kasemir/archiver/>
- [6] M. Sekoranja et al., "Visual DCT – Visual EPICS Database Configuration Tool", ICALEPCS 2001
- [7] R. Sabjan et al., "Visual DCT – EPICS Database Can Be Fun", PCaPAC 2002