VisualDCT User's Manual

KGB-DOC-01/N, Version 1.0, October 21-th, 2001

Matej Šekoranja, e-mail: matej.sekoranja@ijs.si, KGB Team, Jozef Stefan Institute, WWW: http://kgb.ijs.si

Ljubljana, Slovenia

History

Version	Date	Author	Description
1.0	2001-10-21	Matej Šekoranja	Created

Revision

Version	Date	Author	Description

Table of Contents

HISTORY 1
REVISION 1
TABLE OF CONTENTS 1
1 INTRODUCTION 2
2 REFERENCES 2
3 BASIC PRINCIPLES 2
3.1 RUNNING VISUALDCT 3
4 FEATURES 3
4.1 RAPID DATABASE DEVELOPMENT (RDD) 3
4.2 DATABASE FILE PARSER, INPUT/OUTPUT FILE 3
4.3 GROUP, RECORD, LINK REPRESENTATION 4
4.3.1 LINKING 5
4.4 GROUPING - HIERARCHY SUPPORT 5
5 USER INTERFACE 5
5.1 GRAPHICAL USER INTERFACE 5
5.1.1 MAIN WINDOW 5
5.1.2 INSPECTOR WINDOW 6
5.1.3 CONSOLE WINDOW 6
5.1.4 SETTINGS WINDOW 6
5.2 COMMAND REFERENCE 6
5.2.1 MENU COMMAND REFERENCE 7
5.2.1.1 File menu 7

5.2.1.2 Edit menu 7
5.2.1.3 View menu 7
5.2.1.4 Help menu 8
5.2.2 MOUSE COMMAND REFERENCE 8
5.2.3 KEYBOARD COMMAND REFERENCE 8
6 FUTURE PLANS 9
7 CONCLUSION 9
8 APPENDIX 10

1 Introduction

Visual Database Configuration Tool (VisualDCT) [1] is an EPICS database configuration tool completely written in Java and therefore supported in various systems. It was developed to provide features missing in existing configuration tools as Capfast [2] and Graphical Database Configuration Tool (GDCT) [3]. Visually VisualDCT resembles GDCT; records can be created, moved and linked, fields and links can be easily modified. But VisualDCT offers more: using groups, records can be grouped together in a logical block, which allows a hierarchal design. Additionally indication of data flow direction using arrows makes the design easier to understand. VisualDCT has a powerful DB parser, which allows importing existing DB and DBD files. Output file is also DB file, all comments and record order is preserved and visual data saved as comment, which allows DBs to be edited in other tools or manually.
This manual describes the VisualDCT tool version 2.0, build 1207.

2 References

[1] http://www.sls.psi.ch/controls/software/VisualDCT/visual_dct_demo.html – VisualDCT
[2] http://www.phase3.com/epics.html - CapFast with EPICS
[3] http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/ReleaseNotes/GDCT313.html - GDCT
[4] http://java.sun.com/j2se/ - The Java(TM) 2 Platform, Standard Edition
[5] http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/ReleaseNotes/DCT313.html - DCT

3 Basic principles

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 facilities 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. As VisualDCT executes, it attempts to trap and display the most common situations that might lead to diminishing the integrity of the user supplied information.

3.1 Running VisualDCT

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. The name of the file depends on the build of VisualDCT (for version 2.0, build 1207, the name of the file is `visualdct1.207.jar`). This file has to be added to the java classpath variable (search path for application classes and resources) to help JVM find `kgb.vdct.VisualDCT` class, which is the main class of the VisualDCT.

Usage of VisualDCT:

```
o) Usage: java kgb.vdct.VisualDCT [<DBD>] [<DB>]
```

VisualDCT accepts two parameters which are not obligatory: database definition (.dbd) file and record instance database (.db) file. If there is no definition database specified an *Open dialog* will appear allowing you to specify database definition (.dbd) file. If even then there is no valid definition database specified VisualDCT will exit with the following output:

```
o) No DBD loaded! Exiting...
```

An example of running VisualDCT, using `sls.dbd` definition database and `test.db` instance database file:

```
java –cp visualdct1.207.jar -DVDCT_DIR=~/VisualDCT kgb.vdct.VisualDCT sls.dbd test.db
```

`VDCT_DIR` environment variable is used to define the default working directory.
VisualDCT also maintains its configuration file - `VisualDCT.set`.

4 Features

4.1 Rapid Database Development (RDD)

VisualDCT can be considered as a rapid database development tool - unintuitive database construction using ordinary text editors can be done quickly with a few simple mouse-clicks minimizing all unnecessary keyboard input. Visualization of the record instance database makes databases easier to understand, errors are much easier to find (e.g. broker links are indicated by a red cross) and helps find a better design of the databases, allowing user to user hierarchal design and split databases into logical blocks.

4.2 Database file parser, input/output file

VisualDCT creates and maintains only one file, the record instance database (.db) file, and does not have any additional graphical information file avoiding any possible consistency problems when having multiple files, all necessary visual composition data is stored as a comment at the end of the record instance database (.db) file.
An example of simple record instance (.db) file:

```
#! Generated by VisualDCT for Java v2.0

# this is a record comment
record(ai,ai001) {
  # this is a field comment
  field(INP,"ao001")
}

# another record comment
record(ao,ao001) {
}

#! Further lines contain layout data used by VisualDCT

#! Record(ai001,2241,2345,0,1,"ai001")
#! Field("ai001.INP",16711731,1,"ai001.INP")
#! Link("ai001.INP","ai001/INP")
#! Connector("ai001/INP","ao001.VAL",2505,2495,16711731,"")
#! Record(ao001,2641,2500,0,1,"ao001")
#! Field("ao001.VAL",16711731,0,"ao001.VAL")
```

VisualDCT has powerful parser which has ability to parse already existing record instance database (.db) files, files which have been created or modified with other tool. It also detects syntax errors in databases, including definition databases. Defective visual composition data or its absence are safely handled and do not raise any critical error, VisualDCT simply automatically layouts all objects without any visual data. What is more, VisualDCT preserves comments and record/field order in the record instance database, which offers the ability to edit your databases in other tools or manually without making any harm to the databases and VisualDCT.

4.3 Group, record, link representation

Record is represented as a write square with its type and name written inside. Below the line inside the record there is an area where all fields with non-default value are shown.

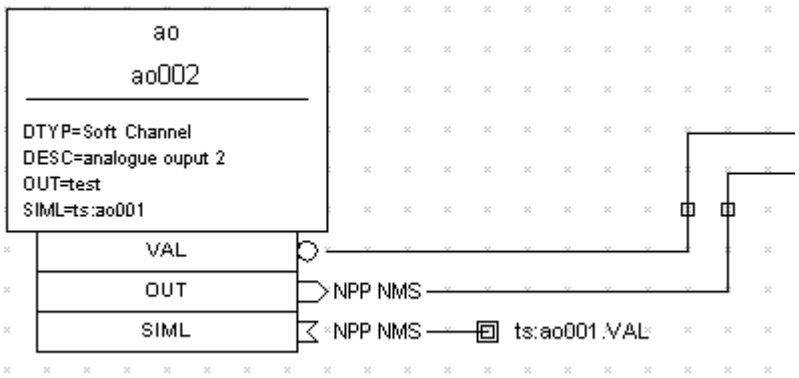

Figure 1 – Representation of the record

There are three types of fields that can appear as part of the record (white squares below the record): VARIABLE (data), INPUT, OUTPUT and FORWARD fields. Variable fields hold a piece of data, such as the VAL or HIHI fields. Since the variable fields can be populated by other record's output fields and read by other record's input fields, a field node will appear below the record. Additionally indication of data flow direction using arrows makes fields easy to distinguish: circle for VARIABLE fields, out-arrow for OUTPUT and FORWARD fields and in-arrow for INPUT fields. A multi-point wire can be drawn between any two linkable fields simply by adding connectors (moveable

small squares on a link line). If a link is an inter-group link (link between two fields which are not in the same group), the link is represented as a line going in the screen with the target link name shown by side.
Group is represented as a white square with its name inside (see *Appendix - Figure 3, object 5*).

4.3.1 Linking

There are two ways of linking:

- value of the INPUT, OUTPUT or FORWARD link field is entered using Inspector tool
- using linking capability of VisualDCT using only mouse:
    1. Right click on the parent record of the INPUT, OUTPUT or FORWARD link field
    2. Pop-up menu will appear, choose the appropriate link field
    3. The parent record will blink until the VARIABLE field or record if FORWARD link is determined; to do this, there are tree options:
        - left click on the record – VAR field is used, or record if FORWARD link
        - left click on the field – link to clicked filed is created
        - right click on the record – pop-up menu will appear allowing you to select the VARIABLE field

4.4 Grouping - hierarchy support

VisualDCT support hierarchal design of the databases. Hierarchy is based on the naming, for instance record with name *grp1:ao001* belongs to group *grp1* and record *grp1:grp2:ao002* belongs to group *grp2* which belongs to *grp1*, so groups can be also nested. In previous examples ":" character was used as a grouping separator, which is the default, but it can be easily changed.
Double click on the group goes (dives) into the group and shows only the records and groups in this group.

5 User Interface

As every powerful IDE also VisualDCT provides indispensable facilities as clipboard and undo support. A great effort was given to synchronization between the record instance database and its visualization. Every change done visually is immediately reflected in the database and vice versa; all actions like moving, renaming and deletion of records which affect links are automatically fixed by the VisualDCT.

5.1 Graphical User Interface

Graphical User Interface of the VisualDCT tool consists of 4 windows (see *Appendix - Figure 3*):

1. Main window
2. Inspector window
3. Console window

and Settings window

5.1.1 Main window

The main window consists of:

1. Main menu

2. Toolbar – makes access to the frequently used actions easier.
3. Workspace with *Navigator* – it is the main component of the VisualDCT tool, it provides visualization and the capability of editing the record instance database. Navigator is a miniature view of the whole workspace. Using mouse over the navigator you can easily move through the workspace.
4. Status bar with *Zoom scale slider* – shows the name of the active definition database and the name of the current group. Zoom scale slider is used to easily change zoom scale.

5.1.2 Inspector window

The inspector tool provides a capability of inspecting (examining) and modifying of all objects properties. Basically the inspector tool is already all needed to edit record instance databases – it replaces ordinary text editor.
The inspector window consists of:

1. Object combo-box – shows currently inspected object and allows user to choose another object.
2. Property table – name-value pair table allowing user to inspect or modify fields. Record fields are grouped according “promptgroup” field defined in definition database.
3. Comment text-area – shows record comment and allows user to modify it.
4. Status bar – provides basic help, the value of the “prompt” field defined in definition database is shown for fields and so helping uses to understand the meaning of the fields (e.g. LBRK – “Last beak point”).

A macro definition can be entered for any field, including menus and links. Any changes to fields take place immediately in the visual composition.

5.1.3 Console window

Console window is used to replace standard output of the JVM, which is often ignored by the user. All output is redirected to the console which pops up every time a new message appears in it and so informs user about the new message.

5.1.4 Settings window

Setting windows allows changing some additional settings of the VisualDCT tool. Currently only settings regarding grouping are available.

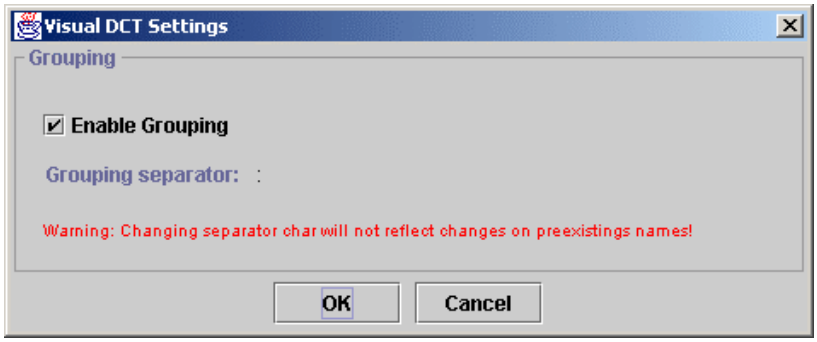

Figure 2 – Settings window

5.2 Command reference

This section describes all commands available by the VisualDCT tool.

5.2.1 Menu command reference

This section describes menu commands available by the VisualDCT tool.

5.2.1.1 File menu

- **New** – close the currently active record instance database (.db) file, and allow the user to create a new record instance database (.db) file.
- **Open** - close the currently active record instance database (.db) file, and provide a file selection window which will allow the user to open a new existing record instance database (.db) file. The record instance database will be checked for consistency with loaded definition database(s).
- **Import DB** - provide a file selection window which will allow the user to specify a new existing record instance database (.db) file which will be added (appended) to the existing active database.
- **Import DBD** - provide a file selection window which will allow the user to specify a new existing definition database (.dbd) file which will be added (appended) to the existing active database definition database.
- **Save** – save the currently active record instance database (.db) file.
- **Save As** – save the currently active record instance database (.db) file, and allow the user to specify a name of the file into which the database will be saved.
- **Save As Group** - save the currently active group of record instance database (.db) file as an standalone database, and allow the user to specify a name of the file into which the database will be saved.
- **Print** – print the current visible area of the record instance database.
- **Print Preview** – display a view of the active record instance database as it will be printed.
- **Page Setup** – change the printer page options.
- **Exit** - exit the VisualDCT tool.

5.2.1.2 Edit menu

- **Undo** – undo the last action.
- **Redo** – redo the previously undone action.
- **Cut** – cut the selection and put it on the clipboard.
- **Copy** – copy the selection and put it on the clipboard.
- **Paste** – insert the clipboard contents to the workspace.
- **Move/Rename** – move/rename the selection.
- **Group** – group the selection.
- **Ungroup** – ungroup the selection of groups.
- **Delete** – delete the selection.
- **Select All** – select all objects in the current group.

5.2.1.3 View menu

- **Flat View** – not implemented
- **Level up** – move to the parent group.
- **Zoom In** – increase zoom scale by 10%.
- **Zoom Out** – decrease zoom scale by 10%.
- **Zoom Selection** – zoom the selection to fit the screen.
- **Base view** – move to the centere of the workspace and set zoom scale to 100%.
- **Toolbar** – toggle toolbar visibility.
- **Statusbar** – toggle statusbar visibility.
- **Navigator** – toggle navigator visibility.
- **Show Grid** – toggle grid visibility on the workspace.
- **Snap to Grid** – snap objects to the grid.
- **Settings** – pop-up the settings dialog.

5.2.1.4 Help menu

- **Help Topics** – list help topics (temporarily only mouse commands are shown).
- **Books Online** – not implemented
- **About Box** – display program information, version number and copyright.

5.2.2 Mouse command reference

Table of all available mouse commands:

BUTTON	*TRIGGER*	*ACTOR*	*ACTION*
left	click	group, record	Select object
	double-click	group	Go into group
		record, field	Inspect object
		blank workspace	Create new record
	click, drag	group, record / selection	Move object / selection
		navigator	Move through the workspace
	Shift + drag	blank workspace	
	drag	blank workspace	Select groups, records
right	click	record, field, connector	Popup object specific menu
	Shift + click	field with more than one link	Rotate link
	drag	blank workspace	Zoom in selection

5.2.3 Keyboard command reference

Table of all available keyboard commands:

Ctrl-A – Select all objects | Ctrl-Q – Exit VisualDCT | Ctrl-V – Paste from clipboard

Ctrl-C – Copy to clipboard | Ctrl-S – Save current DB | Ctrl-X – Cut to clipboard

Ctrl-Y – Redo action

Ctrl-Z – Undo action

Delete – Delete selected object

Shift-↑ – Move to parent group

Shift-← – Zoom out by 10%

Shift-→ – Zoom in by 10%

Shift-Space – Zoom selection

6 Future plans

Since VisualDCT is still an active project, there are some features to be implemented in the future releases of VisualDCT:

- XML support
- Add arbitrary text, lines, rectangles for documentation
- Handling more than one DB at the same time
- Wrapper to handle a so called startup scripts files
- Optimize print preview performance
- Debugging (with active connection to EPICS DB)
- Hierarchy support like in Capfast
- … and any user feature requests

7 Conclusion

If this manual did not meet all of your expectations or if you have any questions or suggestions, please feel free to contact the author at the following address: matej.sekoranja@ijs.si.
Thank you for reading this document.

8 Appendix

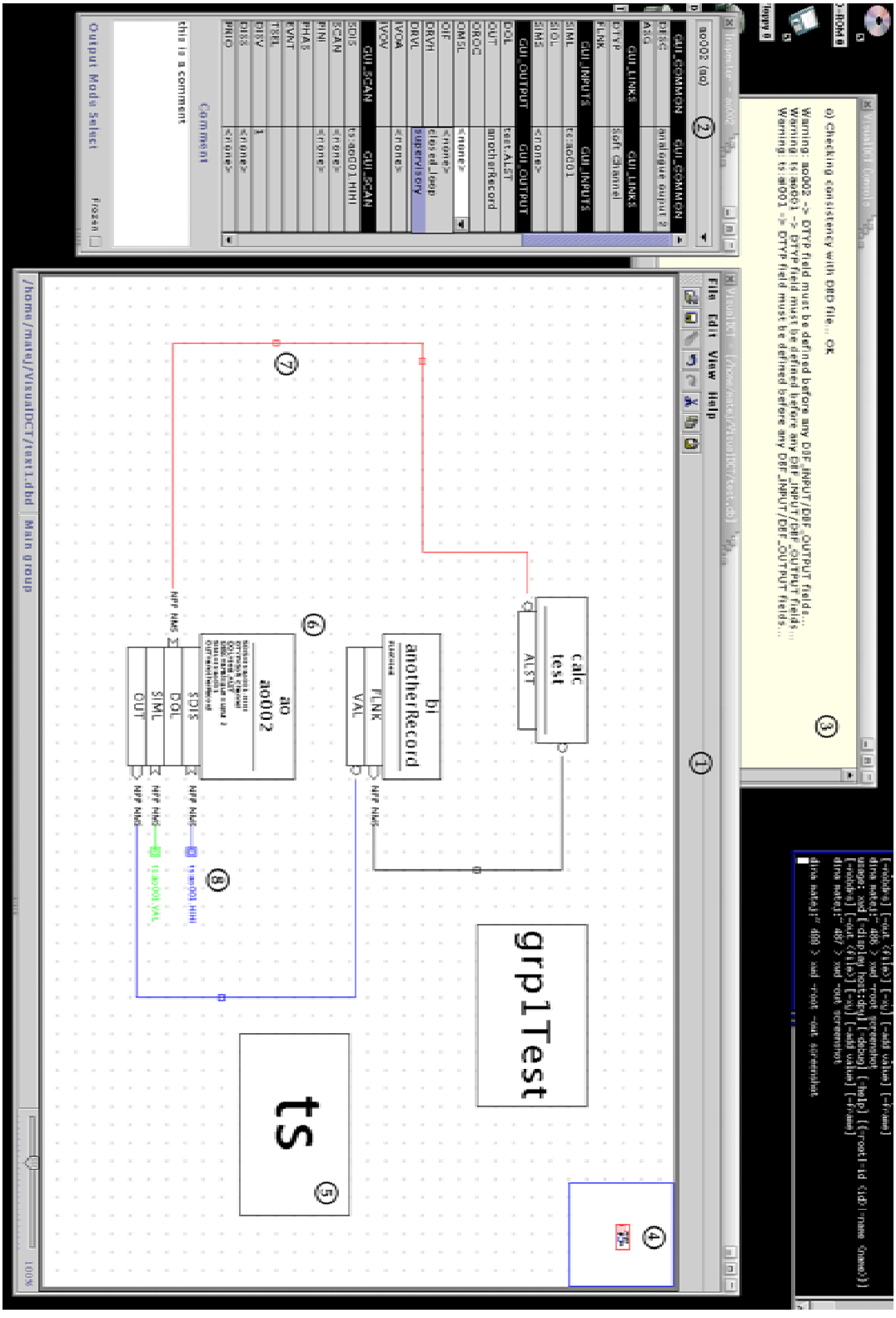

Figure 3 – Graphical User Interface of the VisualDCT tool