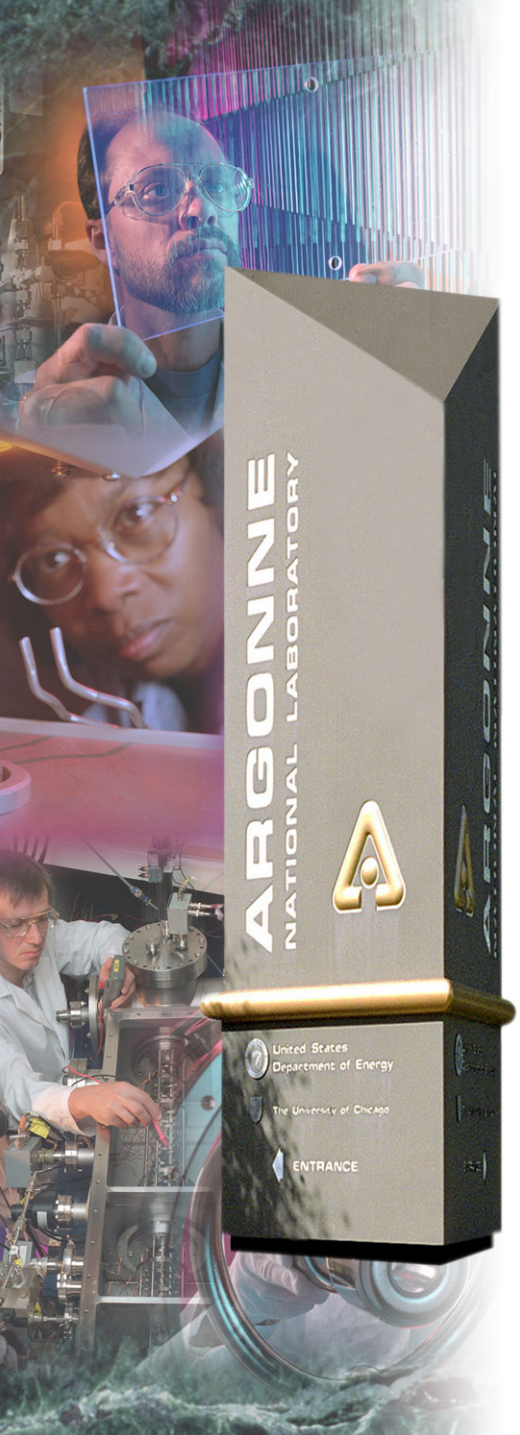


Getting Started with EPICS Lecture Series

Database Design with VisualDCT

Nicholas Di Monte
Controls Group
APS Accelerator System Division
November 02, 2004

Argonne National Laboratory

Office of Science
U.S. Department of Energy

A U.S. Department of Energy
Office of Science Laboratory
Operated by The University of Chicago

Database Design with VDCT

- **Outline**
  - Introduction to VDCT
  - Using VDCT
  - Converting database files to VDCT
  - New features/options
  - Some VDCT examples
  - Known problems with VDCT
  - Resources
  - Acknowledgements

What is VDCT?

- **VDCT is Visual Database Configuration Tool**
- **Beta version released summer of 2002, funded by SLS**
- **Developed by *Cosylab Ltd.***
- **Replacement for text editor, DCT, JDCT, GDCT or Capfast**
- **VDCT developed to provide missing features in Capfast(?) and GDCT.**
- **Supports hierarchical design**
- **Written in Java**
  - Therefore supported in various systems
  - Java Runtime Environment 2
- **Importing existing DB and DBD files**

What is VDCT?

- **VDCT features**
  - GUI features
    - *Clipboard, undo, redo, object inspector, visual linking*
    - *Data flow arrows, not process flow*
  - Supports hierarchal design
    - *Based on the pvname separator*
    - *Grouping “grp1:grp2:test1AO”*
    - *VDCT templates can be used.*
      - Separate VDB file as a template with ports and macros defined.

What is VDCT?

- **VDCT features**
  - Powerful DB parser
    - *Supports existing DB's*
    - *Preserves DB comments, record/field order*
      - # normal comments
      - #! VDCT layout comments
    - *DB's can be edited manually*
  - Single file which contains both DB and display data
    - *GDCT created two separate files*

What is VDCT?

- **VDCT features**
  - Rapid database development
  - Simple mouse-clicks
  - Visualization of record instances
    - *Easier to understand*
      - Yet no field description as with DCT, JDCT & GDCT
    - *Detect errors faster*
      - e.g. broken links shown with a cross
  - Database can be split into logical blocks (grouping)
    - *e.g. hierarchical design*
  - Printing ?

Using VisualDCT

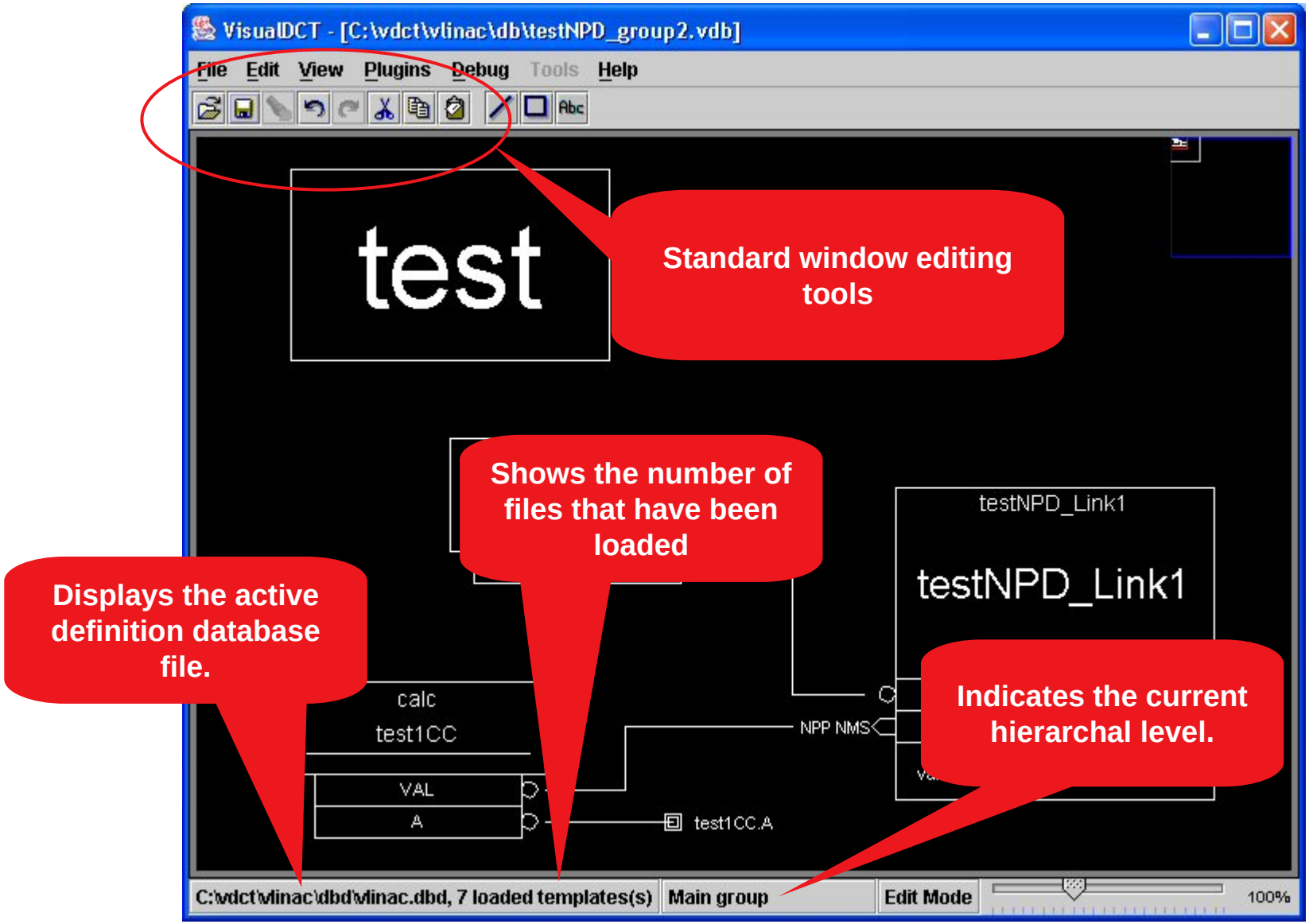
Using VisualDCT

- **Launch VisualDCT by executing jar file**
  - Only file provided in the zip file
  - Current version is build 1249, installed on solaris.
    - *Problem with build 1250*
    - *Currently testing build 1253, used in this presentation.*
- **On solaris just type vdct**
  - This script defines the Class Path and then runs the latest version, at this time build 1249 will be executed.

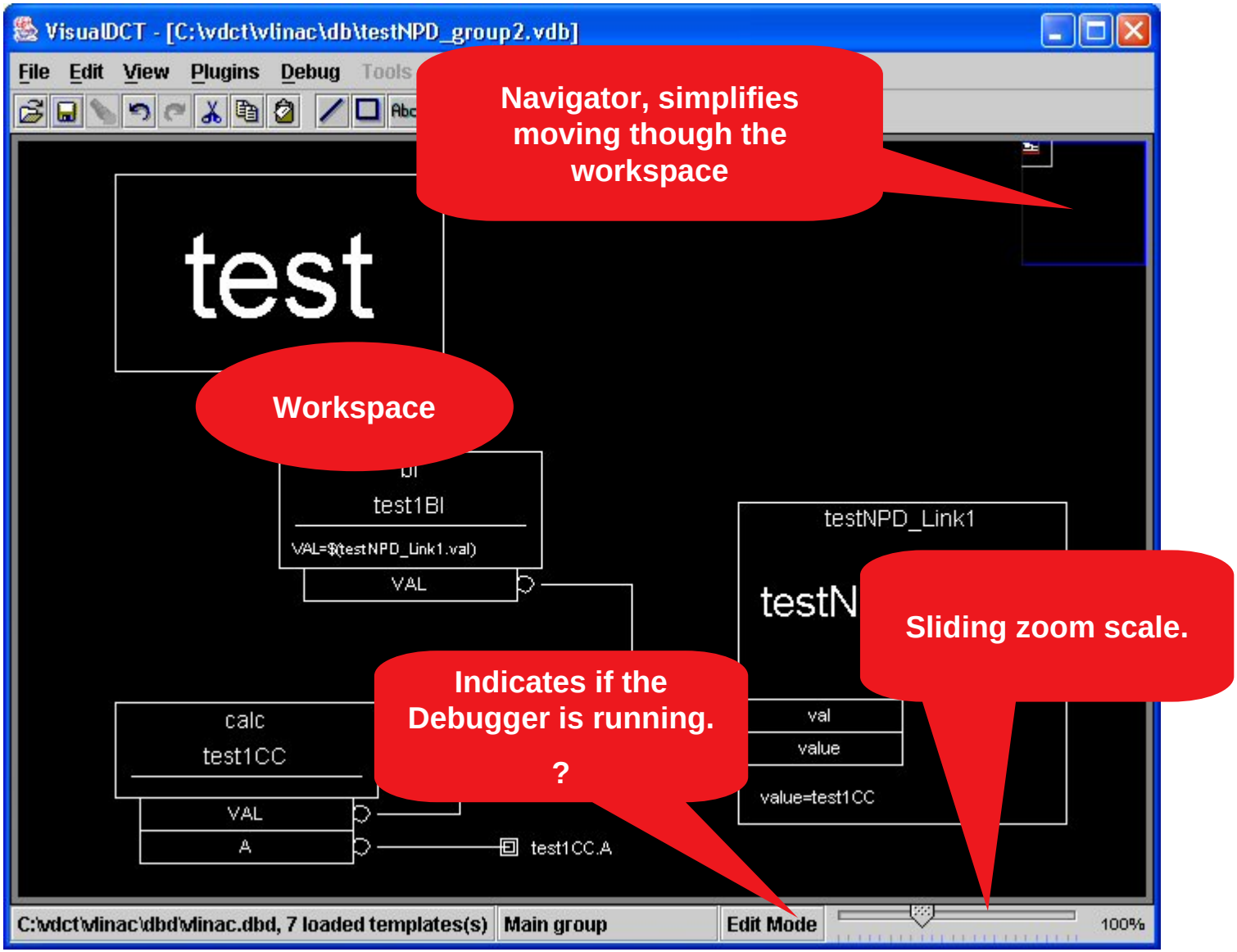
Using VisualDCT

- **To start VisualDCT in Windows**
  - Execute (double click) “VisualDCT2.4.1253.jar”
  - Or, use command line options
    - *VisualDCT2.4.1253.jar [<DBDs>] [<DB>]*
- **Load DBD file(s)**
  - Recommend selecting save option in “DBD Manager”
- **Load DB or VDB file.**
- **Save work with a VDB extension. (recommended)**
- **Once a VDB file is created and saved, no need to specify DBD files, DBD files will be included at the beginning of a VDB file.**
  - #! DBDSTART
  - #! DBD(“../../dbd/vlinac.dbd”)
  - ……
  - #! DBD(“other DBD file”)
  - #! DBDEND

Using VisualDCT

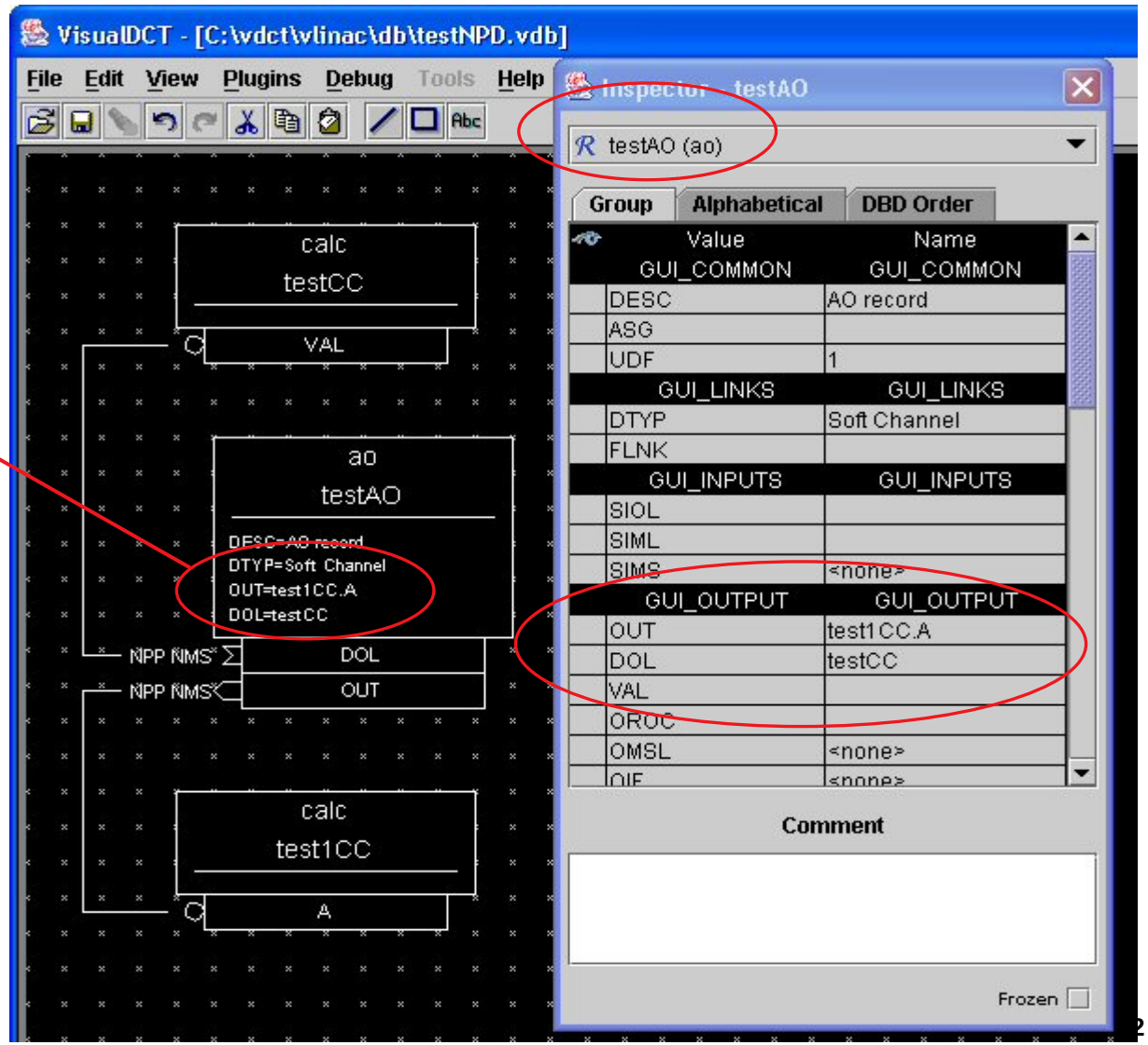


Using VisualDCT



Using VisualDCT

- **Inspector**
- **Records**
  - Fields
    - *Visible*
- **Links**
  - Data flow



Using VisualDCT

Field Visibility Values

-0: NON_DEFAULT_VISIBLE

-Blank for build 1249

-0: VISIBILITY_SELECT

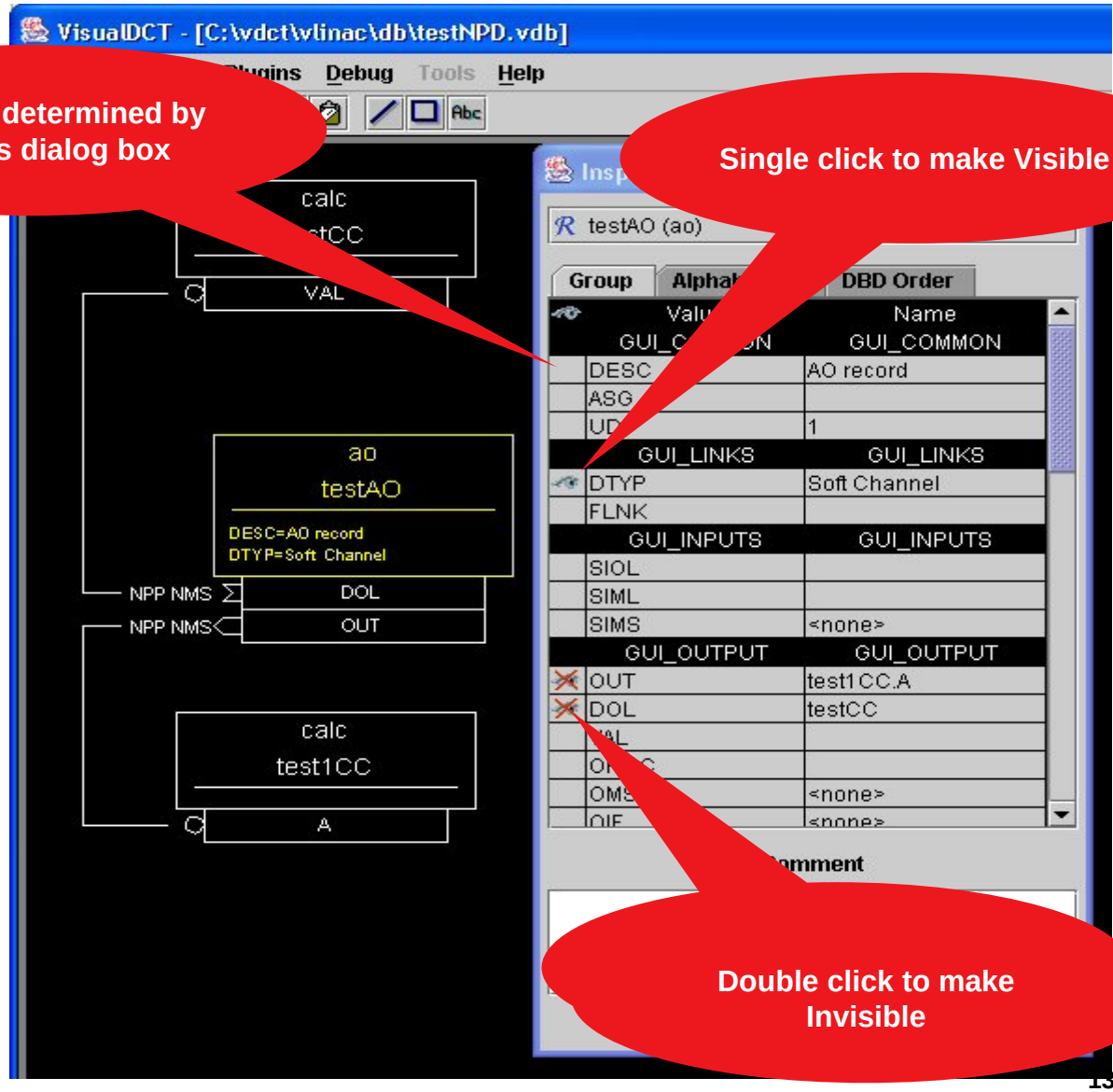
-Blank for build 1250

-1: ALWAYS_VISIBLE

-Eye

-2: NEVER_VISIBLE

-Eye w/Red X



Using VisualDCT

Visibility text in vdb file

#! Visibility("testAO.DTYP",1)

#! Visibility("testAO.OUT",2)

#! Visibility("testAO.DOL",2)

Visibility Defined:

#! Visibility(“fieldname”, visibility)

Where visibility:

- 0 – NON_DEFAULT_VISIBLE
  - Build 1249 and earlier.
- 0 – VISIBILITY_SELECT
  - Build 1250 and later.
- 1 – ALWAYS_VISIBLE
- 2 – NEVER_VISIBLE

Using VisualDCT

Visibility text in vdb file

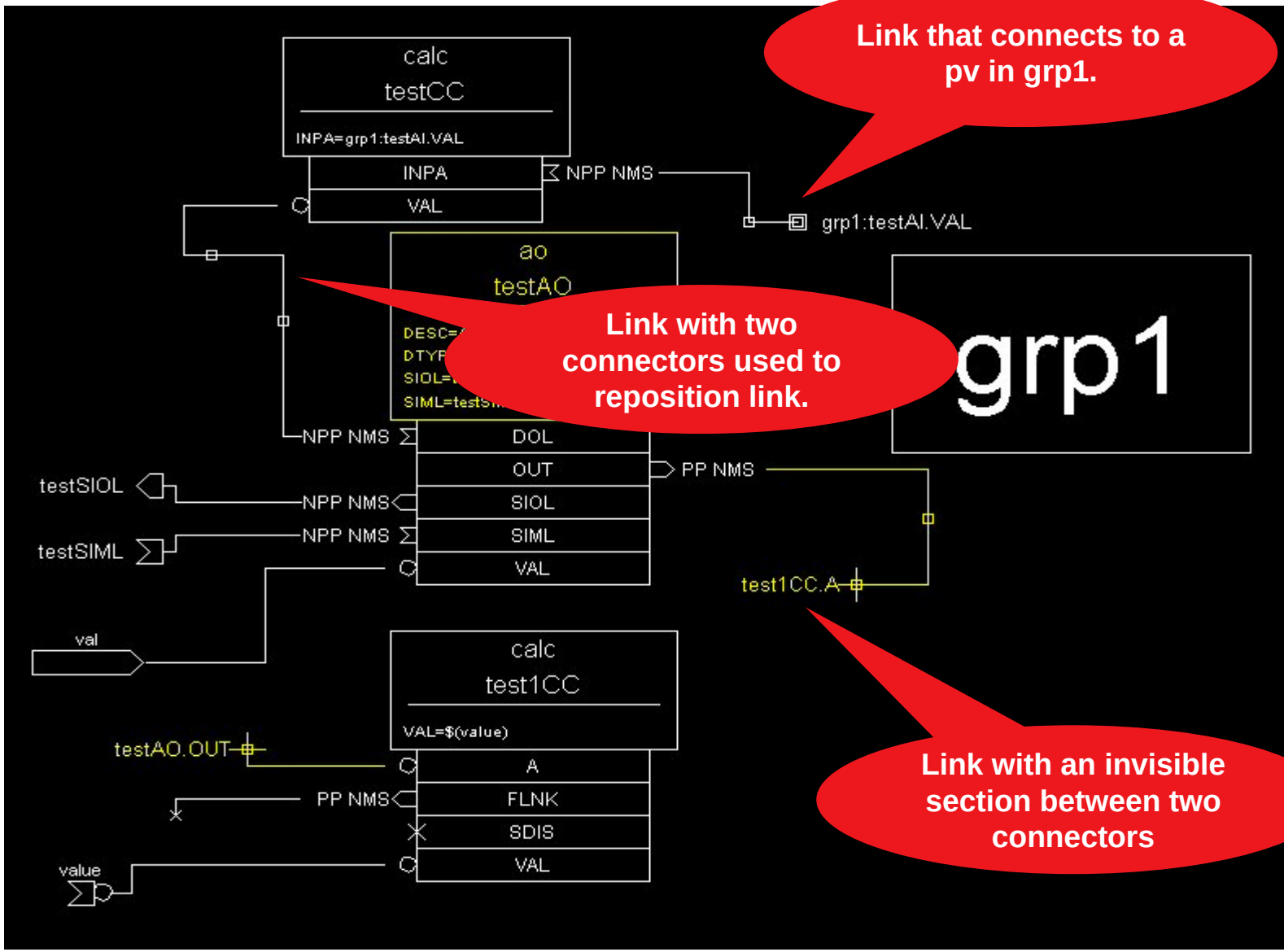
Why Option #0 should be NON_DEFAULT_INVISIBLE for build 1249

- The visual graphics should convey the logic of the design.
- When importing an old text database, almost all fields have been defined with at least the default value.
- The designer should highlight the fields of interest to clarify the logic flow.
- Limited work space in the graphical window.

Build 1250 added an option to change this:

- Defined in: View => Settings => Visual TAB
- Uncheck "Show value of fields when it is not default"
  - *All fields are now hidden except for those fields set to visible*

Using VisualDCT (links)



Using VisualDCT (links)

Link/Connector text in vdb file

```
#! Field("testAO.OUT",255,1,"testAO.OUT")
#! Link("testAO.OUT","testAO/OUT2")
#! Connector("testAO/OUT2","testAO/OUT1",660,340,255,"",0)
#! Connector("testAO/OUT1","testAO/OUT",620,380,255,"",1)
#! Connector("testAO/OUT","test1CC.A",220,480,255,"",0)
```


Using VisualDCT (links)

Link/Connector text in vdb file

#! Field(“fieldname", color, rotated, “description")

Where rotated: (*not documented*)

- 0 – Left side of field box
- 1 – Right side of field box

#! Link(“fieldname", “inLinkID")

#! Connector(" inLinkID "," outLinkID ", x, y, color, “desc", option)

Where option: (*not documented*)

- 0 – Visible
- 1 – Invisible
- 2 – External Input
- 3 – External Output

Using VisualDCT

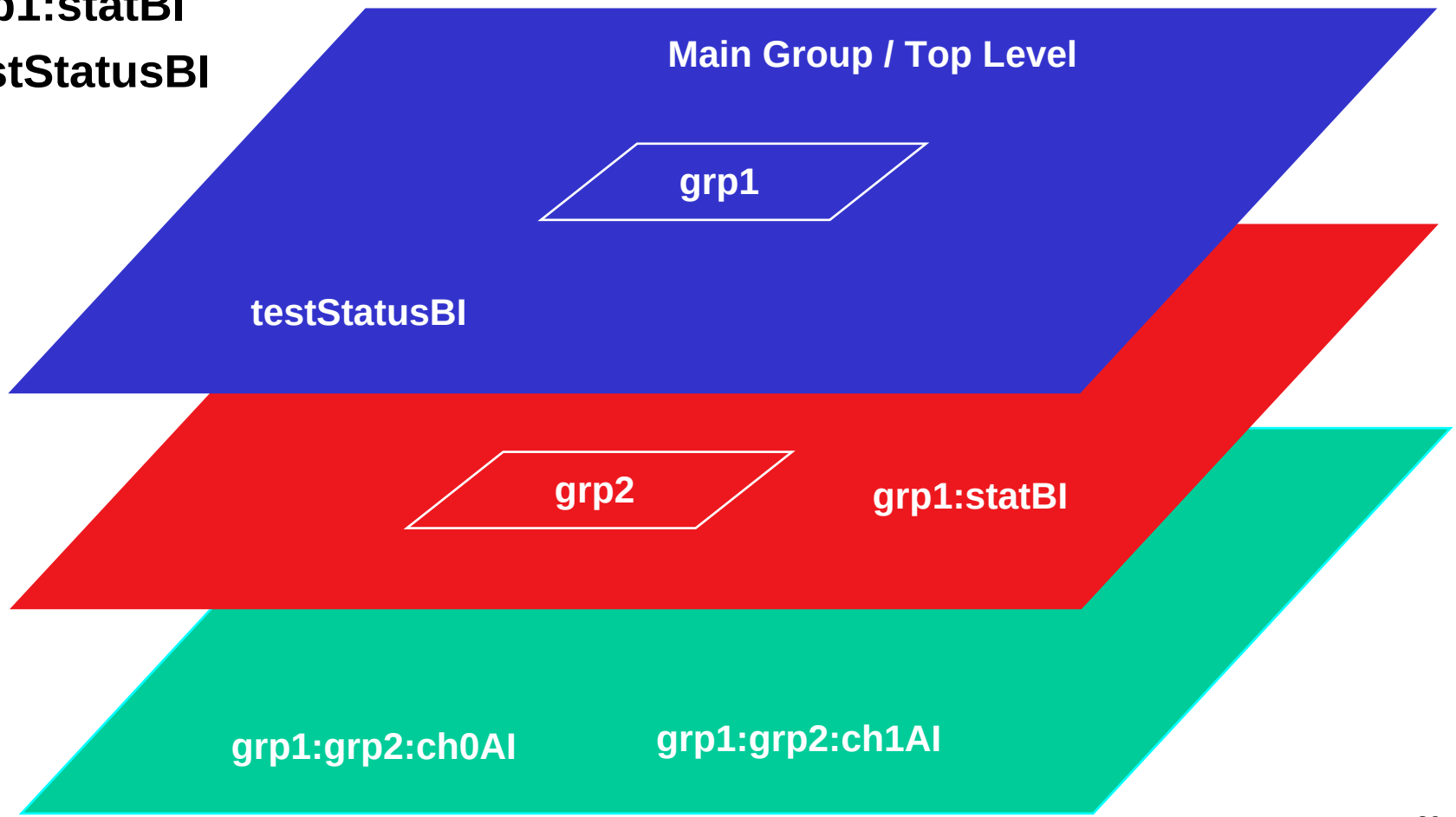
Hierarchy Support

- **Based on the pvname separator**
- **Grouping “grp1:grp2:test1AO”**
  - This will create three levels
- **Grouping must be enabled before loading DB**
  - Separator must also be defined
- **Support templates**
  - Use Generate… command to flatten vdb with templates
  - *Macros* pass information into a template
  - *Ports* pass information upwards out of a template
  - Use import command to add template

Using VisualDCT

Hierarchy

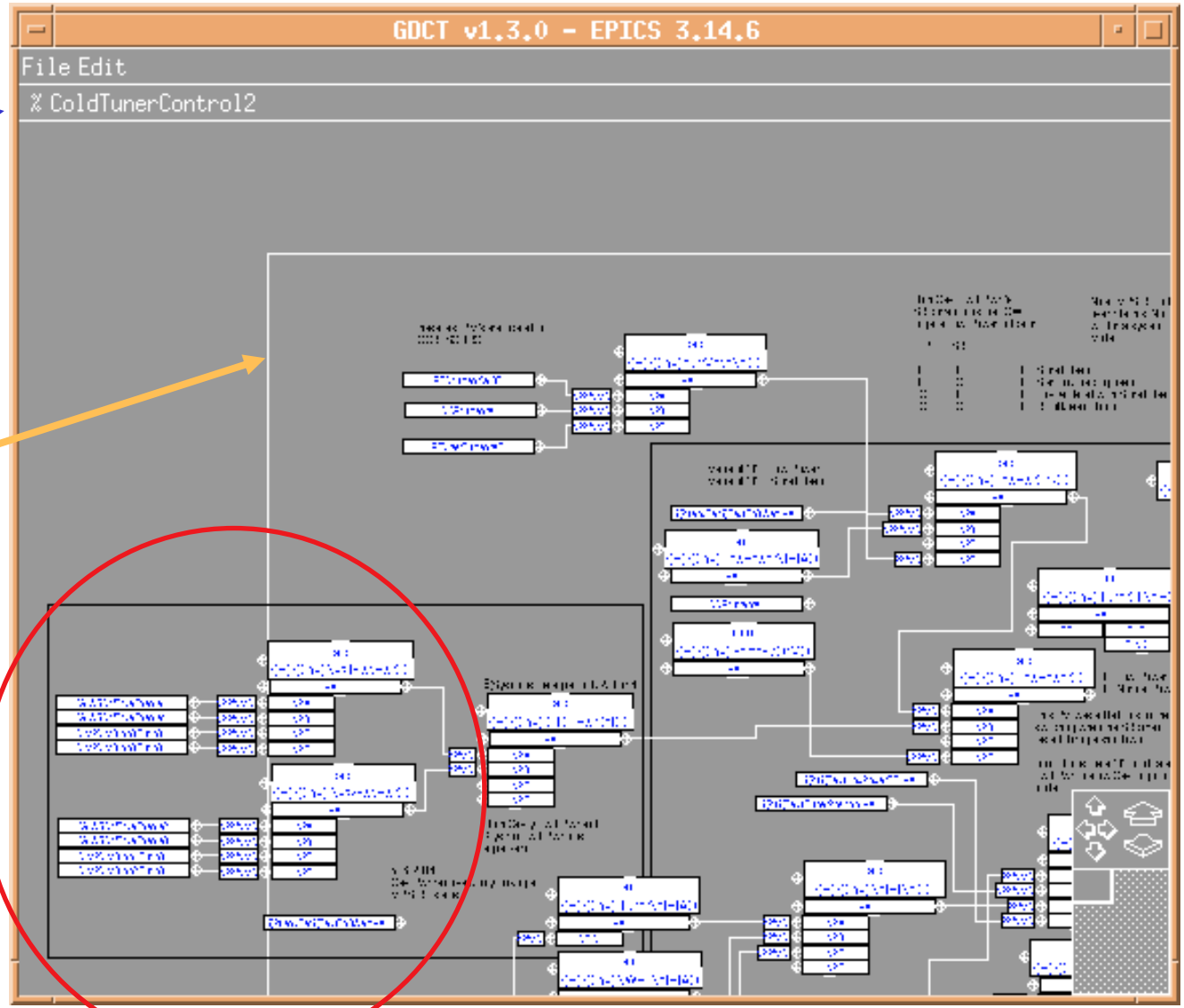
- grp1:grp2:ch0AI, grp1:grp2:ch1AI
- grp1:statBI
- testStatusBI



Converting files

- **Converting from GDCT313 to VisualDCT**
  - From the File menu, select "Save As VDCT…"
  - Minor touch up maybe needed.
  - All graphic items **must** be in the defined workspace outlined by the white border in GDCT

Converting files



Converting files

- **Converting a DB text file to VisualDCT**
  - In VisualDCT select View menu, then Settings
    - *Then select Visual Tab*
      - Uncheck "Show value of fields when it is not default"
  - Load DB file
  - Rearrange display for clarity
  - Save with a .vdb extension (recommended)
    - *The Generate command will only create a .db file*

Converting files

Tools not to use on VDB files, JDCT & DCT313

- **They remove all display formats**

Caution when using “vi” or text editor

New command/options

New features in Build #1253

- **Morph command**
  - Allows a record to be converted from AI to AO, etc.
- **Speed option**
  - Silhouette when moving a record
- **Field Visibility**
  - Option to change the default visibility
- **Window Pan Direction**
  - Push or pull while moving around workspace.

VisualDCT examples

- **Virtual LINAC**
  - Database in original text form
  - DB imported to VDCT
  - DB modified for appearance, logical flow
    - *Color links*
    - *Invisible links*
    - *Text boxes*

VisualDCT examples

- **Database conversion, with build #1253**
  - Start VDCT, load DBD file
  - Grouping turned off
  - Set “Field Visibility”
  - Load a text only DB (rfD1_Vx2.db)
  - Clear “Field Visibility”
  - Load a modified DB (rfD1_Vx2.vdb)

VisualDCT examples

- **Editing commands, with build #1253**
  - Load a modified DB (rfD1_Vx2.vdb)
  - Edit commands……
  - Moving around workspace, (Ctl-arrow)
    - *Navigator window*
  - Zoom slider
    - *Keyboard (Shift-[left/right]arrow)*
    - *Selected zoom, (Shift-space)*
  - Inspector
  - DBD manager
    - *Saving*
    - *Invalid DBD*

VisualDCT examples

- **Creating a new database**
  - Start VDCT, load a DBD file
  - Create records
  - Add links, using mouse
  - Use inspector on records, define CALC field
  - Add “connector” on a link
    - *Show invisibility option*
  - Add an external input and output links
  - Demonstrate morph feature

VisualDCT examples

- **Hierarchy, grouping function, with build #1253**
  - Restart VDCT with grouping enabled
  - Load a modified DB (rfD1_Vx2.vdb)
    - *Why this is annoying*
      - Designed to be small flat DB
      - Most records create their own group in this example
  - Accessing the different group levels
    - *Double clicking on object will descend into group*
    - *Shift-Up will ascend one level at a time*
  - Status bar indicates the current level in the design

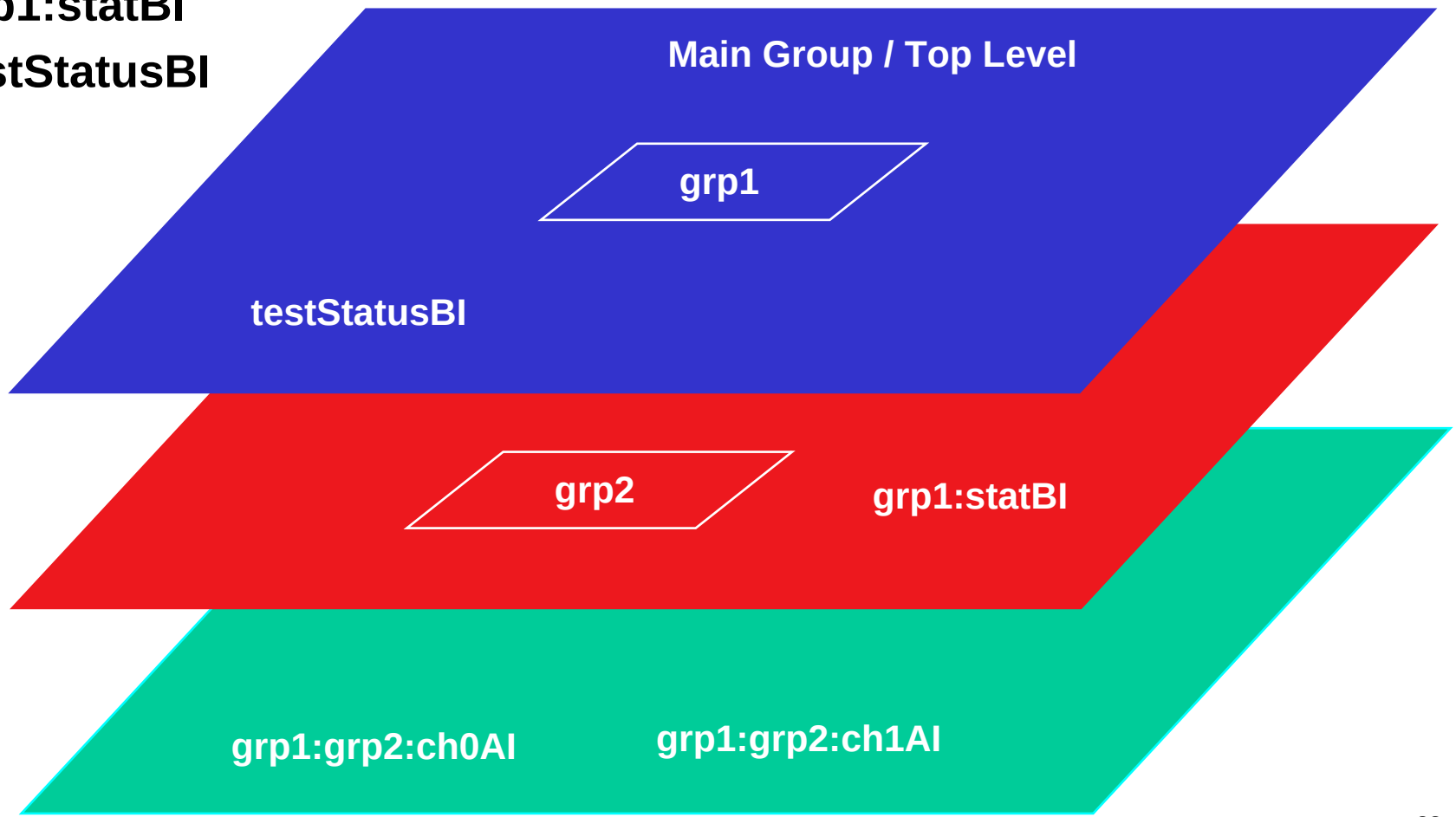
VisualDCT examples

- **Hierarchy, cont.**
  - Load testNPD_group2.vdb
  - With grouping enabled, create a pv with several Grouping separators, we will use “ **:** ”, as in gpr1:gpr2:gpr3:ch0AI
  - Descend into group test
  - Descend into template testNPD_Link1
- **Groups are maintained in a single file**
- **Template is maintained in an external file**

VisualDCT examples

Hierarchy

- grp1:grp2:ch0AI, grp1:grp2:ch1AI
- grp1:statBI
- testStatusBI

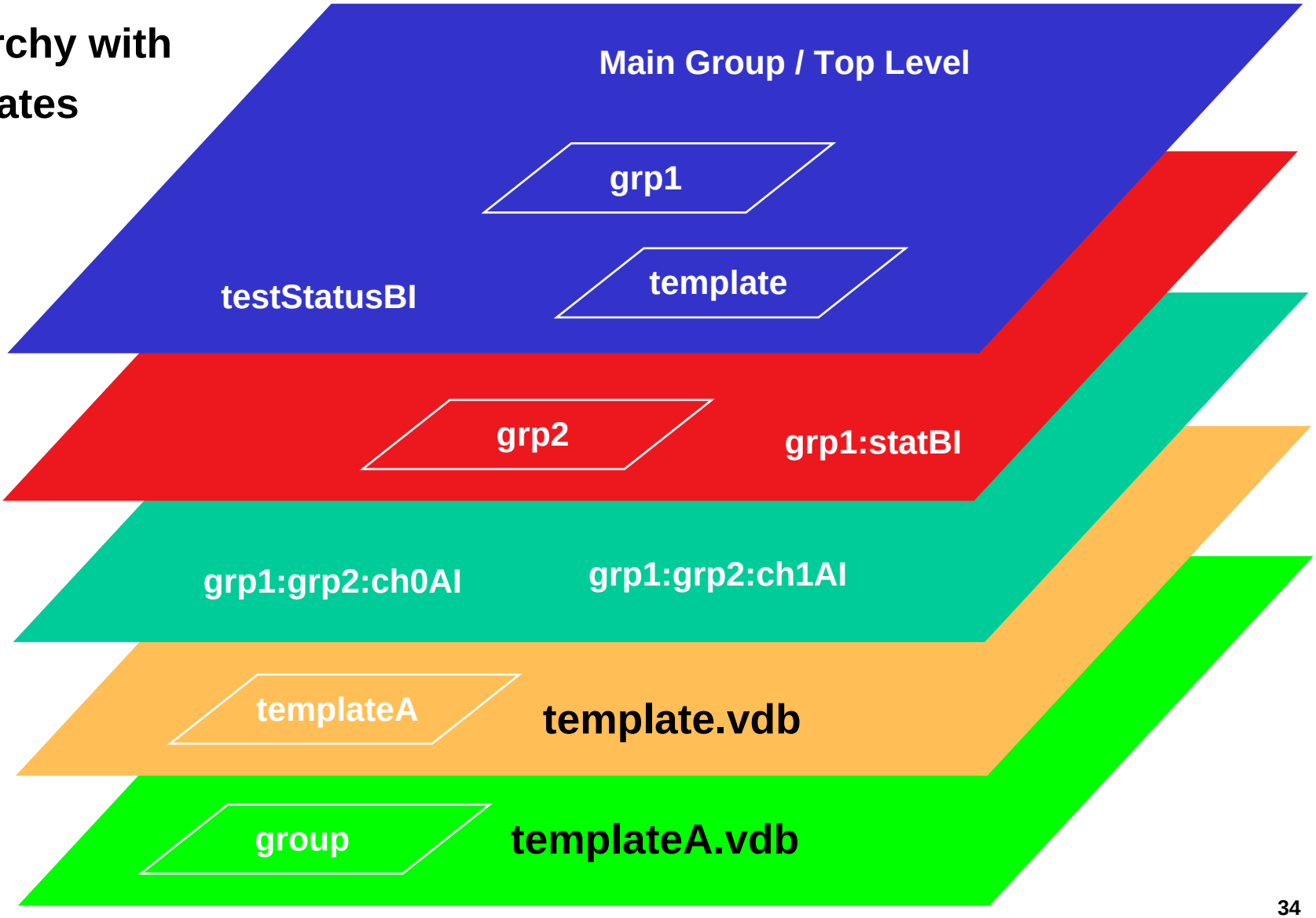


VisualDCT examples

- **Templates, brief overview**
  - Restart VDCT
  - Import testNPD_Link1.vdb
    - *Or load testNPD_Link1.vdb first.*
    - *Every DB load is considered a template.*
      - Not all templates have access ports
  - Place a template object on the workspace
  - Save database
  - Descend into template
  - Discuss template properties
  - The “Generate…” comand
    - *Combines multiple template files with the main file and generates a single DB file.*
  - Template can have templates
    - *Example in hierarchy/dbs/test.vdb, grouping enabled*

VisualDCT examples

Hierarchy with templates



VisualDCT examples

Hierarchy, cont.

Pros & cons

- **Groups are maintained in a single file**
- **Template is maintained in an external file**

- **Groups have no ports/macros to connect visual links**
- **Templates can define as many as needed for visual links**

- **Groups can define templates**
- **Templates can define groups**

VisualDCT examples

Hierarchy, cont.

Design consideration

- **Is the design large enough to use grouping?**
  - Is the pvname creating single pv groups?
    - *Text editor is just as useful then.*
  - Visual links jumping to other group levels should be kept to a minimum.
- **Can the design be broken up into smaller files (templates)**
  - Main file is then dependent on other files.
  - A port definition will only use the first instance of a template, all ports in subsequent repeated templates will be ignored. That is, all of these same templates will have the same port connected to the same link.

Known problems/concerns

- **Printing ?**
  - What you see is what you get
- **Field visibility**
  - Problem in build 1249 and earlier
  - Option in 1250 and later
- **Grouping option**
  - This should be set in the vdb file
  - Design preference, flat or hierarchy
- **Field box and visible links changed in build #1253**
  - DBs created with 1249 or earlier

Resources

- **Cosylab VisualDCT Project Info**
  - Provides related articles and presentations, links to Builds, demo, bug reports, etc.
  - http://control.cosylab.com/customer_area.php?mid=6&sid=26&ssid=2

- **VisualDCT download page**
  - Download the latest builds, documentation, examples and plugins
  - http://www.cosylab.com/visualdct/builds/VisualDCT/
    - *Build #1250 not recommended for download.*

Acknowledgements

- **Matej Sekoranja is the primary developer of VisualDCT. He has implemented several changes that I have asked for to help import old DB files and has implemented other requested features. He has been a great help.**

- **Ken Evan modified GDCT to provide an option in GDCT to convert a GDCT file to a VDCT file format.**

Visual DCT

Thank you for your time.

Nick Di Monte