Argonne
NATIONAL
LABORATORY

… for a brighter future

U.S. Department
of Energy

UChicago ▶
Argonne LLC

Office of
Science
U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC

Database Design with VisualDCT

Nicholas Di Monte

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 ?

Features of VDCT

- Look & feel for basic window commands
  - ex. copy, paste…..
- Multiple window planes, new to v2.6 build 1274
  - copy from one DB to another.
- Hierarchical designs
  - more on this later
- Templates
  - idea for repeated logic sections
- DBD manager
  - specify more than one DBD file
- Morph
  - ex. change a BI to a BO
- Spreadsheet
  - generate a spreadsheet showing all pvs
  - categorized by pv types
  - all modified fields displayed

Using VisualDCT

Mouse & Keyboard controls	
New Record	Left button double-click on blank space
Inspect Object	Left button double-click on record or field of record
Go into Group	Left button double-click on group
Popup Menu	Right button on record/field/connector
Move Desktop	1) Left button drag on blank space while holding Shift Note: Setting “Window Pan Direction” under “View” will reverse direction of move. 2) Left button drag in Navigator 3) Hold CTL and then arrow buttons
Move object/selection	Left button drag over record or group of items
Start Linking	Left button click on record
Link target to VAR field	Left button click on record
Link target to selected field	Left button click on variable field
Choose variable field	Right button click on record
Select objects	Left button selection
Zoom-in selection	Shift and Space bar
Zoom-in by 10%	Shift and Right arrow
Zoom-out by 10%	Shift and Left arrow
Magnify record or field	Hold Shift, move mouse over object Note: Only works when zoom is less than 100%

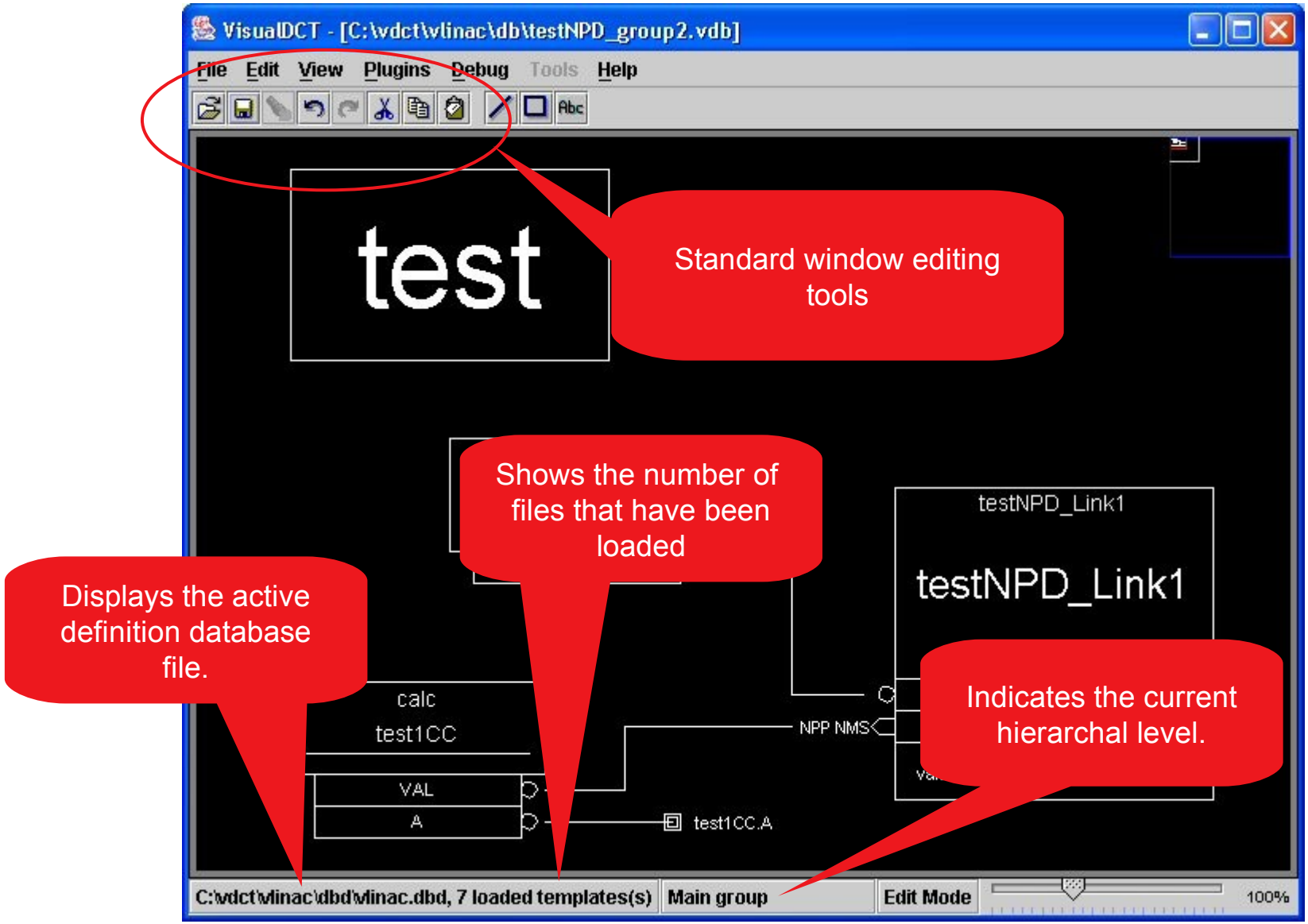
Using VisualDCT

- Launch VisualDCT by executing jar file
- On most machines just type vdct
  - This script defines the Class Path and then runs the latest version.
- To start VisualDCT in Windows
  - Execute (double click) “VisualDCT.jar”
  - Or, use command line options
    - *VisualDCT.jar [<DBDs>] [<DB>]*

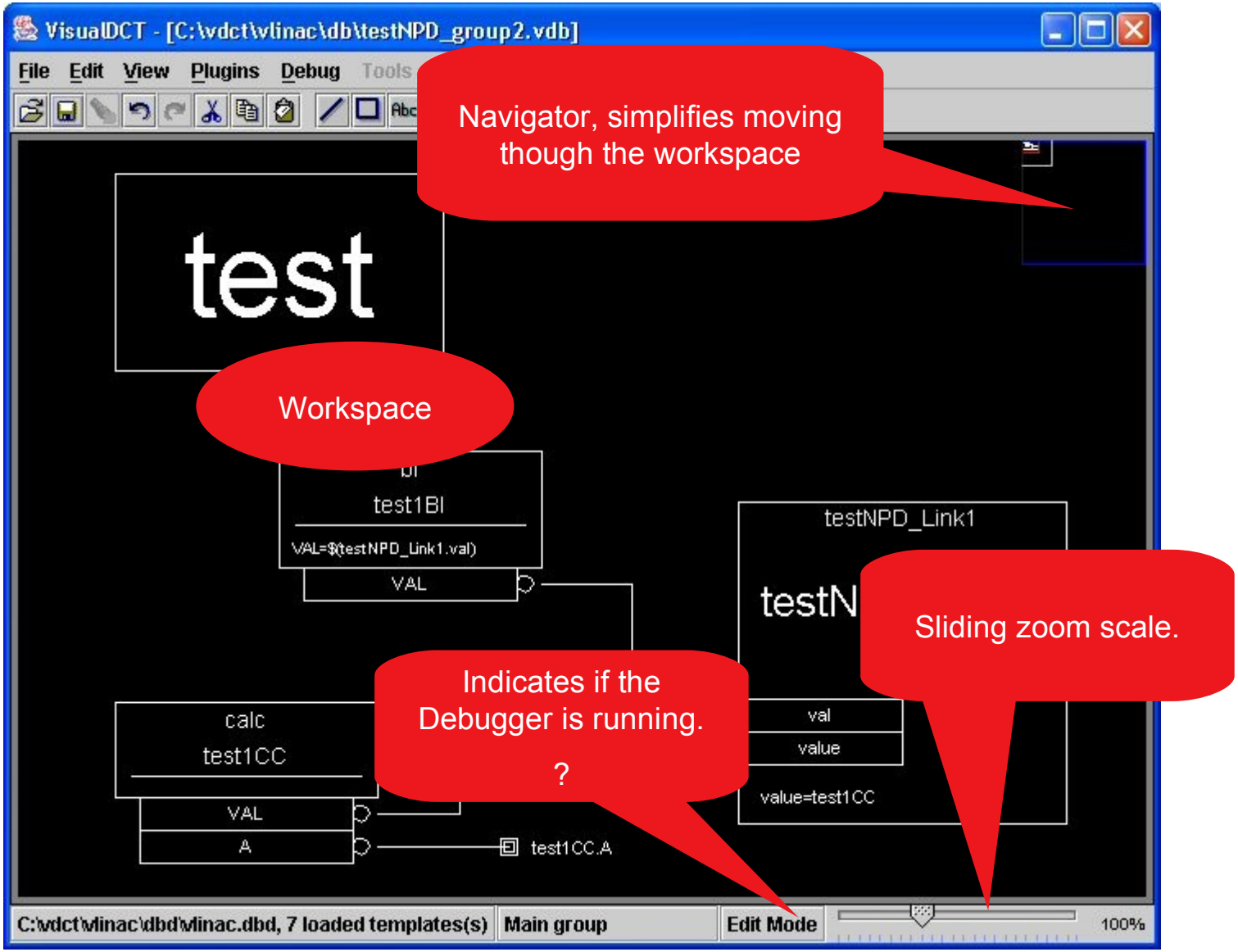
Using VisualDCT

- 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

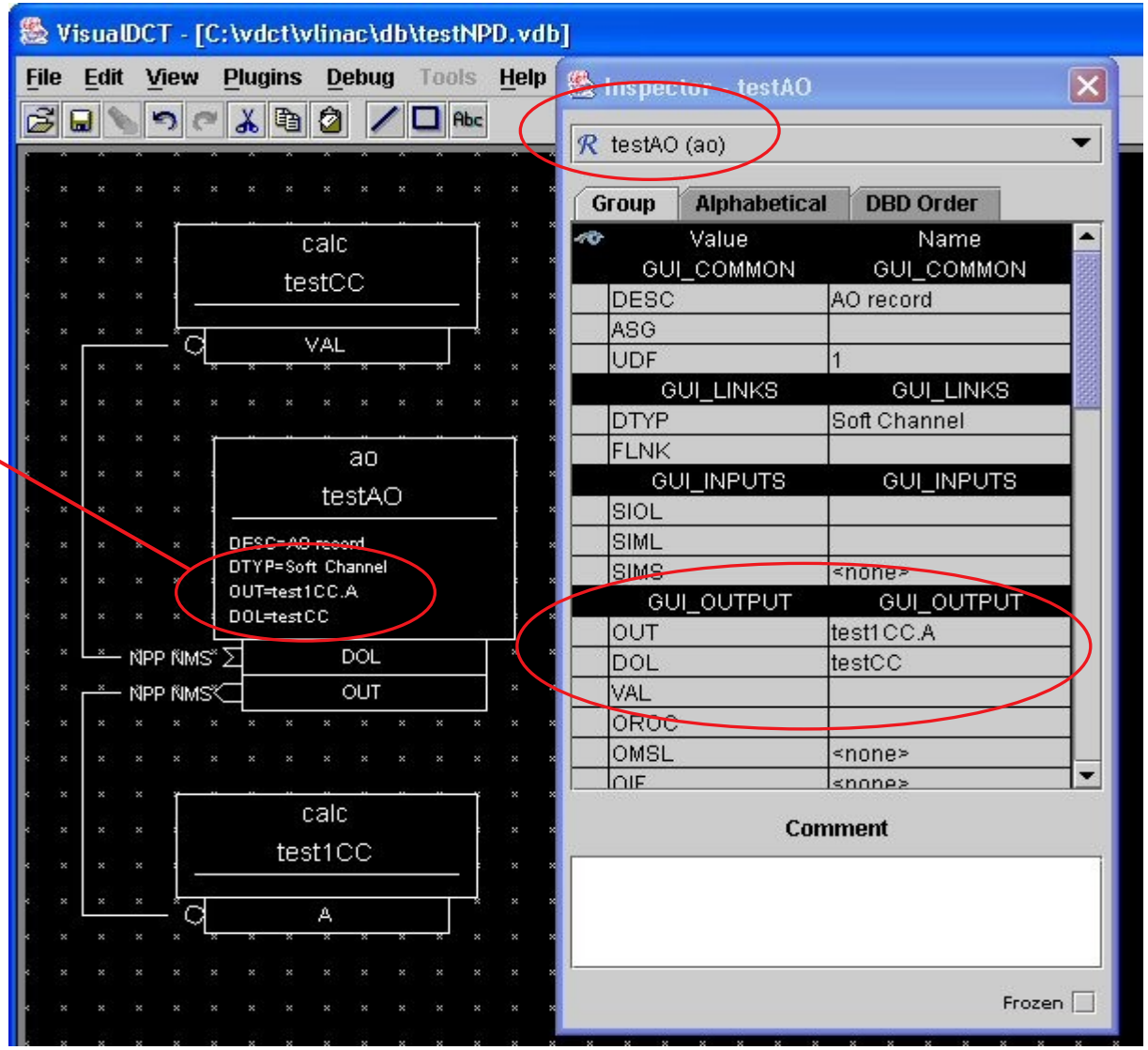


Using VisualDCT



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

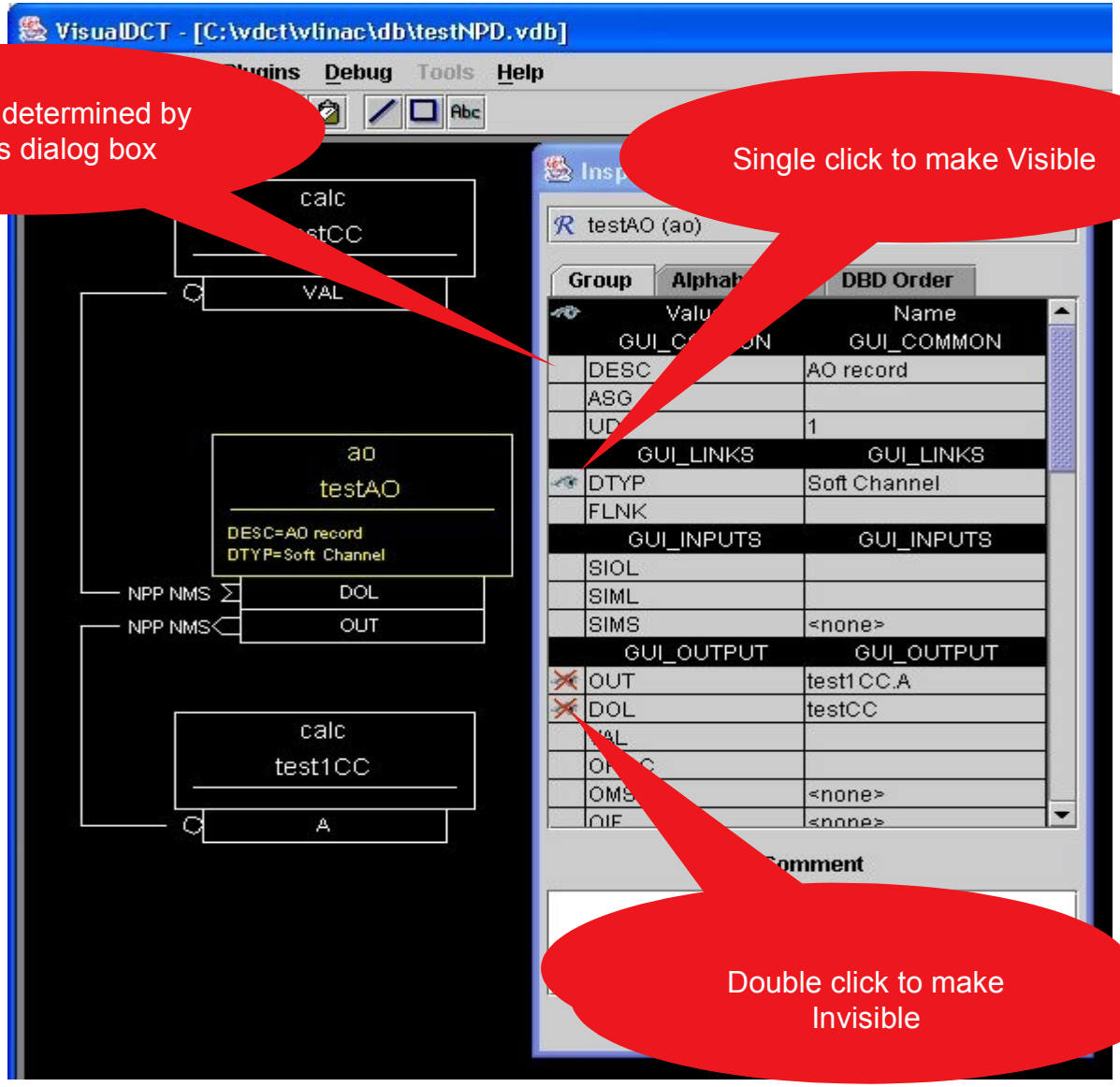
-Blank for build 1250

-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)

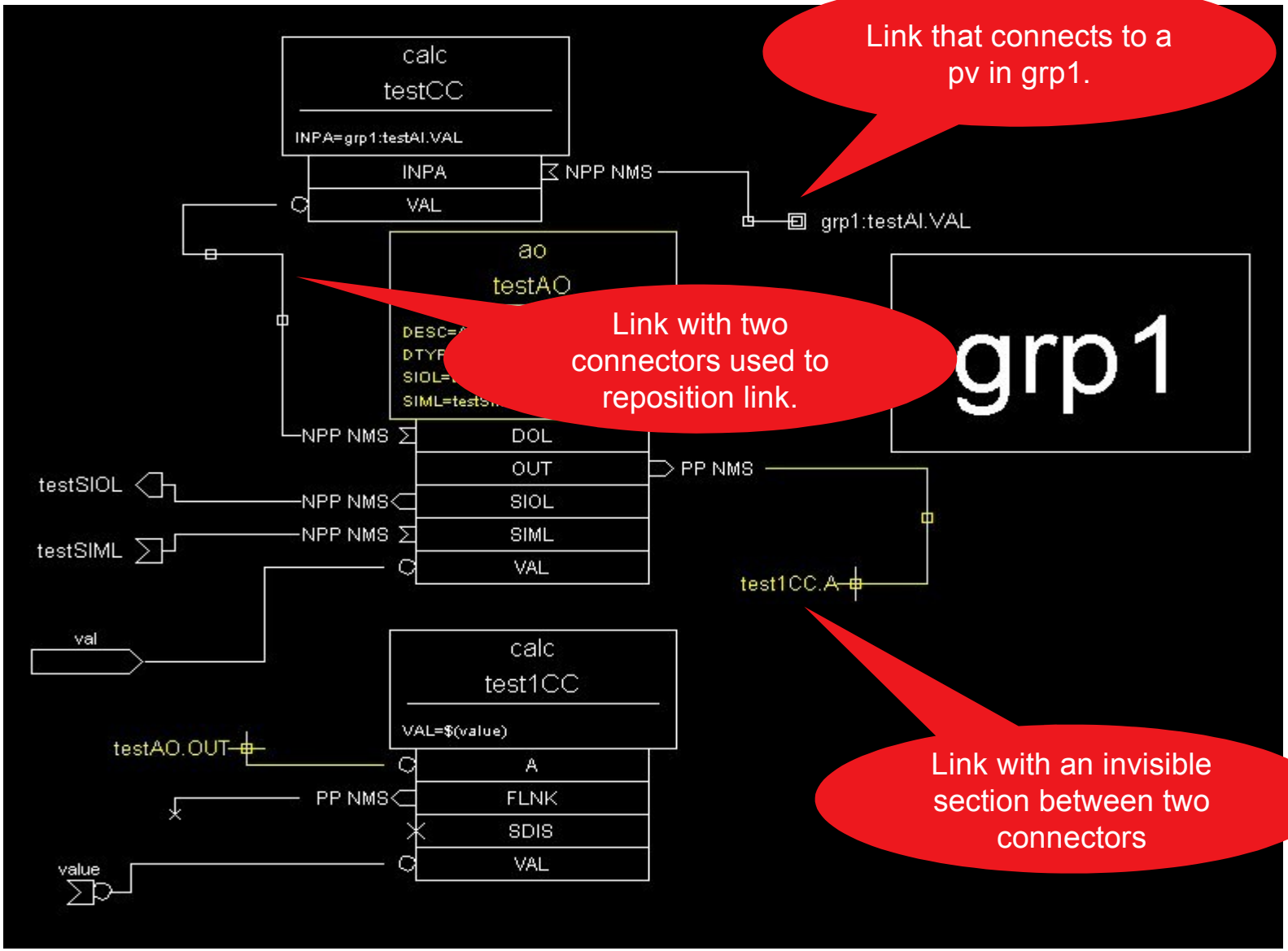
Visibility Defined:

#! 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 (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

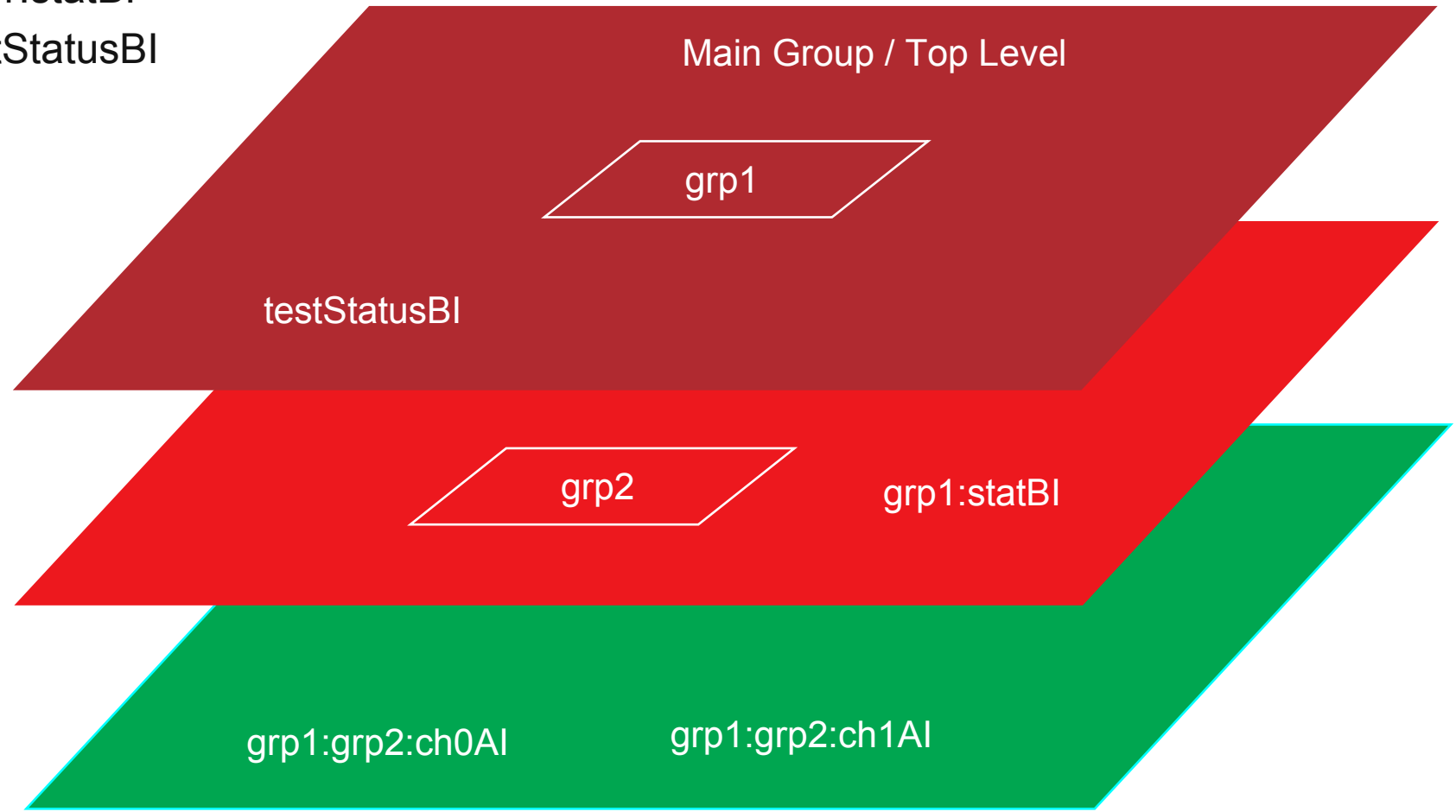
Hierarchy Support

- 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 downward 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 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

VisualDCT Documentation

http://visualdct.cosylab.com/builds/VisualDCT

- Select 2.6.1274, is the latest and most stable version
  - Ver 3 beta is not recommended
- User manual
- Plugins info
- Hierarchy Additions
- Java Channel Access (JCA) plugin
- Other info on VDCT

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*
  - Multiple DB windows
    - *With V2.6 build 1274*
    - *Allows you to copy and paste from one vdb to another vdb.*