

# Enhanced Tcl/Tk Widgets for EDA Applications

Gaurav Bansal, Roshni Lalwani  
gaurav\_bansal@mentor.com, roshni\_lalwani@mentor.com

Gaurav Bansal  
17'th Annual *Tcl/Tk* Conference (Tcl'2010)  
October 11 - October 15, 2010

# DFTVisualizer

DFTVisualizer - design.v (Tessent FastScan v9.2-prerelease)

File Edit Display Trace Data Tools Windows Help

Find:

Mentor Graphics

Browser

Instance Name	UO				Test Coverage	AU
	AAB	UDUNC				
test_design_edt_t...	3	0%	10	0%	65.01%	438
bsr_i1	0	0%	0	0%	NF	NF
core_i	3	0%	12	0%	65.54%	385
test_design...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_design_i	3	0%	12	0%	65.54%	385
addr_0	0	0%	0	0%	76.72%	27
addr_1	0	0%	0	0%	91.67%	10
clkmux	0	0%	0	0%	62.50%	3
datai	0	0%	0	0%	79.31%	24

Hierarchy Library Clocks

Design

Wave

cycle	0	1	2	3	4	5	6	7	8	9
edt...	X									
edt...	X									
edt										

Now 56 ns  
Cursor 1 0 ns

Transcript

```

decompressor_i/ix199/A1
ATPG> // command: add display instance "tap_i/ix77" -p "A1" -b
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix191" -p "Y" -f -one
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix199" -p "A0" -b
ATPG> // command: add display instance "/" -display data
ATPG> // command: add display instance "/bsr_i1" -display data
ATPG> // command: add display instance "/core_i" -display data
ATPG> // command: add display instance "/core_i/test_design_i" -display da
ta
ATPG> rep drc rule
C3: #fails=39 handling=warning (clock may capture data affected by its ca
ptured data)
C4: #fails=3 handling=warning (clock may be affected by its captured data
)
    
```

Debug

Pin: /core\_i/test\_design\_edt\_i/test\_design\_edt\_decompressor\_i/edt\_updat Instance Grouping : OFF

# DFTVisualizer – Design and Wave Window

The screenshot displays the DFTVisualizer interface for a design named 'design.v'. The main window is divided into several panes:

- Browser:** A table showing the design hierarchy and test coverage for various instances.
- Design:** A schematic diagram of the 'test\_design\_edt\_decompressor\_i (51)' block, showing its inputs and outputs.
- Wave:** A timing diagram showing the signals 'edt\_clock', 'edt\_update', and 'edt\_channels\_in' over time. The cursor is positioned at 56 ns.
- Transcript:** A log of ATPG commands and their outputs, including failure reports for rule C3 and C4.
- Debug:** A detailed schematic of the decompressor block, showing internal logic elements like multiplexers and XOR gates.

Instance Name	UO		Test Coverage	AU		
	AAB	UDUNC				
test_design_edt_t...	3	0%	10	0%	65.01%	438
bsr_i1	0	0%	0	0%	NF	NF
core_i	3	0%	12	0%	65.54%	385
test_design...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_design_i	3	0%	12	0%	65.54%	385
addr_0	0	0%	0	0%	76.72%	27
addr_1	0	0%	0	0%	91.67%	10
clkmux	0	0%	0	0%	62.50%	3
data1	0	0%	0	0%	79.31%	24

```
decompressor_i/ix199/A1
ATPG> // command: add display instance "tap_i/ix77" -p "A1" -b
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix191" -p "Y" -f -one
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix199" -p "A0" -b
ATPG> // command: add display instance "/" -display data
ATPG> // command: add display instance "/bsr_i1" -display data
ATPG> // command: add display instance "/core_i" -display data
ATPG> // command: add display instance "/core_i/test_design_i" -display da
ta
ATPG> rep drc rule
C3: #fails=39 handling=warning (clock may capture data affected by its ca
ptured data)
C4: #fails=3 handling=warning (clock may be affected by its captured data
)
```

# DFTVisualizer – Debug Window

The screenshot displays the DFTVisualizer interface for a design named 'design.v'. The main window is divided into several panes:

- Browser:** A table showing test coverage for various instances. The table has columns for Instance Name, UO (AAB, UDUNC), Test Coverage, and AU.
- Design:** A block diagram of the 'test\_design\_edt\_decompressor\_i (51)' component, showing inputs like 'edt\_clock', 'edt\_up date', and 'edt\_channels\_in', and outputs like 'edt\_mask' and 'edt\_scan\_in(size=4)'.
- Wave:** A timing diagram showing signals over time. The 'cycle' column shows values 6, 7, 8, 9. The 'Now' time is 56 ns, and the 'Cursor 1' is at 0 ns.
- Transcript:** A log of ATPG commands and responses, including 'rep drc rule' and 'handling-warning' messages.
- Debug:** A detailed circuit diagram of a portion of the design, showing logic gates, flip-flops, and interconnects. This pane is highlighted with a red border.

Instance Name	UO		Test Coverage	AU
	AAB	UDUNC		
test_design_edt_t...	3 0%	10 0%	65.01%	438
bsr_i1	0 0%	0 0%	NF	NF
core_i	3 0%	12 0%	65.54%	385
test_design...	0 0%	0 0%	NF	NF
test_des...	0 0%	0 0%	NF	NF
test_des...	0 0%	0 0%	NF	NF
test_des...	0 0%	0 0%	NF	NF
test_design_i	3 0%	12 0%	65.54%	385
addr_0	0 0%	0 0%	76.72%	27
addr_1	0 0%	0 0%	91.67%	10
clkmux	0 0%	0 0%	62.50%	3
datai	0 0%	0 0%	79.31%	24

```
decompressor_i/ix199/A1
ATPG> // command: add display instance "tap_i/ix77" -p "A1" -b
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix191" -p "Y" -f -one
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix199" -p "A0" -b
ATPG> // command: add display instance "/" -display data
ATPG> // command: add display instance "/bsr_i1" -display data
ATPG> // command: add display instance "/core_i" -display data
ATPG> // command: add display instance "/core_i/test_design_i" -display da
ta
ATPG> rep drc rule
C3: #fails=39 handling-warning (clock may capture data affected by its ca
ptured data)
C4: #fails=3 handling-warning (clock may be affected by its captured data
)
```



# DFTVisualizer – Transcript Window

The screenshot displays the DFTVisualizer interface with the following components:

- Browser:** A table showing test coverage for various instances.
- Design:** A block diagram of the test\_design\_edt\_decompressor\_i (51) component.
- Wave:** A timing diagram showing signals like cycle, edt..., and edt over time.
- Transcript:** A window showing ATPG commands and their outputs, including coverage rules and instance display commands.
- Debug:** A detailed circuit diagram showing internal components like mux2L, and various xor and nor gates.

Instance Name	UO		Test Coverage	AU
	AAB	UDUNC		
test_design_edt_t...	3 0%	10 0%	65.01%	438
bsr_i1	0 0%	0 0%	NF	NF
core_i	3 0%	12 0%	65.54%	385
test_design...	0 0%	0 0%	NF	NF
test_des...	0 0%	0 0%	NF	NF
test_des...	0 0%	0 0%	NF	NF
test_des...	0 0%	0 0%	NF	NF
test_design_i	3 0%	12 0%	65.54%	385
addr_0	0 0%	0 0%	76.72%	27
addr_1	0 0%	0 0%	91.67%	10
clkmux	0 0%	0 0%	62.50%	3
data1	0 0%	0 0%	79.31%	24

```
decompressor i/ix199/A1
ATPG> // command: add display instance "tap i/ix77" -p "A1" -b
ATPG> // command: add display instance "core i/test_design_edt i/test_desi
gn_edt_decompressor i/ix191" -p "Y" -f -one
ATPG> // command: add display instance "core i/test_design_edt i/test_desi
gn_edt_decompressor i/ix199" -p "A0" -b
ATPG> // command: add display instance "/" -display data
ATPG> // command: add display instance "/bsr_i1" -display data
ATPG> // command: add display instance "/core_i" -display data
ATPG> // command: add display instance "/core_i/test_design_i" -display da
ta
ATPG> rep drc rule
C3: #fails=39 handling=warning (clock may capture data affected by its ca
ptured data)
C4: #fails=3 handling=warning (clock may be affected by its captured data
)
```

# DFTVisualizer

## MtiHierarchy Widgets

The screenshot displays the DFTVisualizer interface with the following components:

- Browser:** A table showing test coverage for various instances. A blue oval highlights the 'MtiHierarchy Widgets' label, with an arrow pointing to the instance tree.
- Design:** A block diagram of the 'test\_design\_edt\_decompressor\_i (51)' component.
- Wave:** A timing diagram showing signals 'edt...', 'edt...', and 'edt' over time.
- Transcript:** A log of ATPG commands and results, highlighted with a red box. A red oval labeled 'Text Widgets' points to this area.

Instance Name	UO				Test Coverage	AU
	AAB		UDUNC			
test_design_edt_t...	3	0%	10	0%	65.01%	438
bsr_i1	0	0%	0	0%	NF	NF
core_i	3	0%	12	0%	65.54%	385
test_design...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_des...	0	0%	0	0%	NF	NF
test_design_i	3	0%	12	0%	65.54%	385
addr_0	0	0%	0	0%	76.72%	27
addr_1	0	0%	0	0%	91.67%	10
clkmux	0	0%	0	0%	62.50%	3
datai	0	0%	0	0%	79.31%	24

```
decompressor_i/ix199/A1
ATPG> // command: add display instance "tap_i/ix77" -p "A1" -b
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix191" -p "Y" -f -one
ATPG> // command: add display instance "core_i/test_design_edt_i/test_desi
gn_edt_decompressor_i/ix199" -p "A0" -b
ATPG> // command: add display instance "/" -display data
ATPG> // command: add display instance "/bsr_i1" -display data
ATPG> // command: add display instance "/core_i" -display data
ATPG> // command: add display instance "/core_i/test_design_i" -display da
ta
ATPG> rep drc rule
C3: #fails=39 handling=warning (clock may capture data affected by its ca
ptured data)
C4: #fails=3 handling=warning (clock may be affected by its captured data
)
```

# Agenda

- Enhancement in MtiHierarchy widget
  - Callback support for improved performance
  - Sub column support
  - Making tree column frozen in place (non-scrollable)
- Enhancement in Text widget
  - Hyperlinks
  - Incremental Parsing

# Agenda

## ■ Enhancement in MtiHierarchy widget

- Callback support for improved performance

- Sub column support

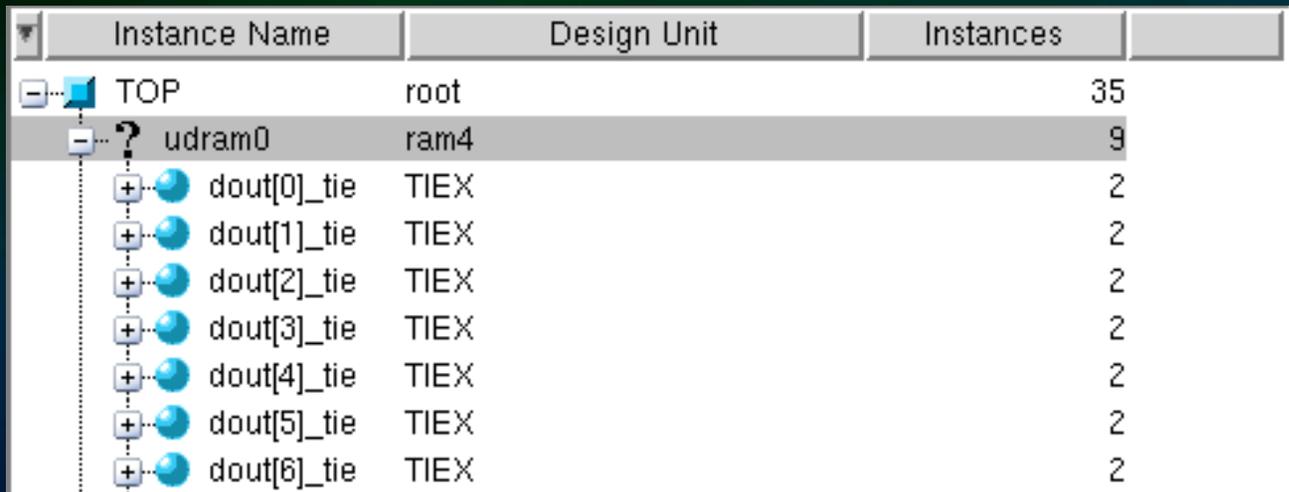
- Making tree column frozen in place (non-scrollable)

## ■ Enhancement in Text widget

- Hyperlinks

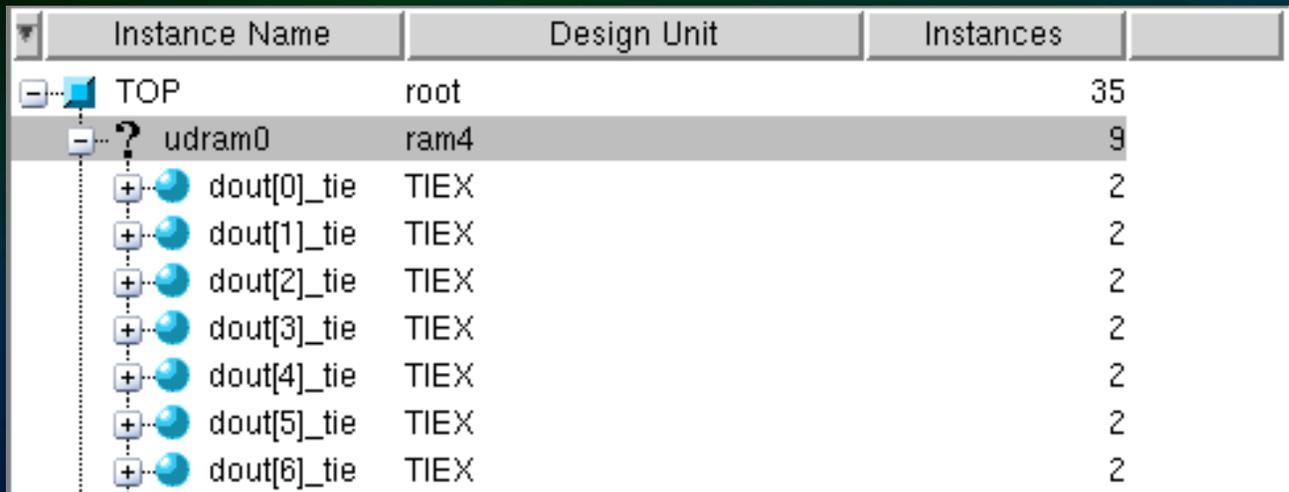
- Incremental Parsing

# Browser Window – MtiHierarchy Widget



Instance Name	Design Unit	Instances
TOP	root	35
udram0	ram4	9
dout[0]_tie	TIEX	2
dout[1]_tie	TIEX	2
dout[2]_tie	TIEX	2
dout[3]_tie	TIEX	2
dout[4]_tie	TIEX	2
dout[5]_tie	TIEX	2
dout[6]_tie	TIEX	2

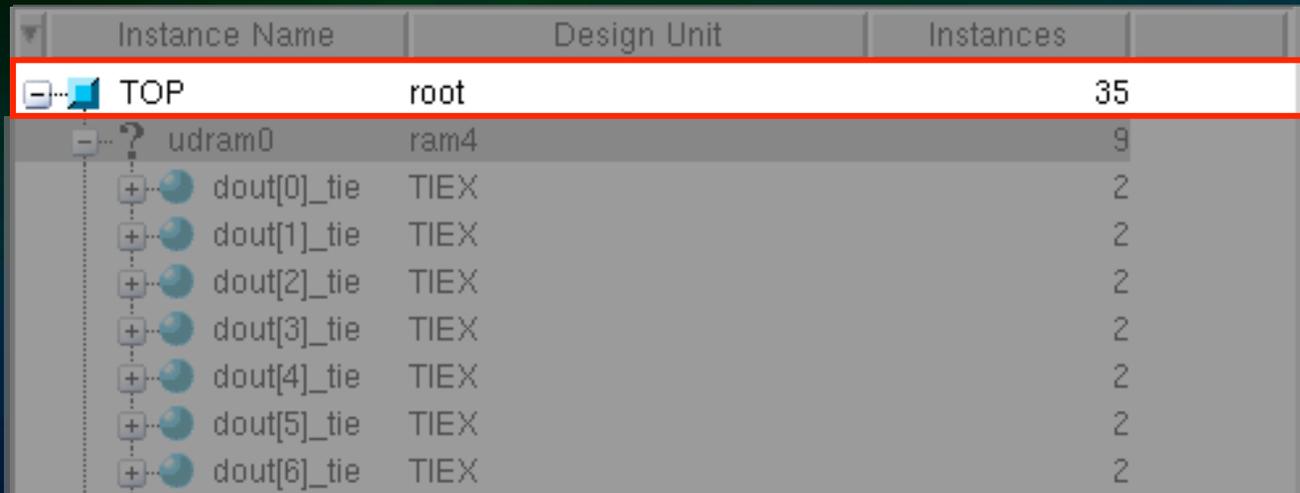
# Browser Window – MtiHierarchy Widget



Instance Name	Design Unit	Instances
TOP	root	35
udram0	ram4	9
dout[0]_tie	TIEX	2
dout[1]_tie	TIEX	2
dout[2]_tie	TIEX	2
dout[3]_tie	TIEX	2
dout[4]_tie	TIEX	2
dout[5]_tie	TIEX	2
dout[6]_tie	TIEX	2

Format used to pass information to MtiHierarchy widget for each row is  
<name> <app data> <label> <tag> <icon> <label> <tag> <icon>...

# Browser Window – MtiHierarchy Widget



Instance Name	Design Unit	Instances
TOP	root	35
udram0	ram4	9
dout[0]_tie	TIEX	2
dout[1]_tie	TIEX	2
dout[2]_tie	TIEX	2
dout[3]_tie	TIEX	2
dout[4]_tie	TIEX	2
dout[5]_tie	TIEX	2
dout[6]_tie	TIEX	2

Format used to pass information to MtiHierarchy widget for each row is  
<name> <app data> <label> <tag> <icon> <label> <tag> <icon>...

# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35

0xAAAAAA {HINST IN 4 -1} TOP branch vlog\_inst



<name>

# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35

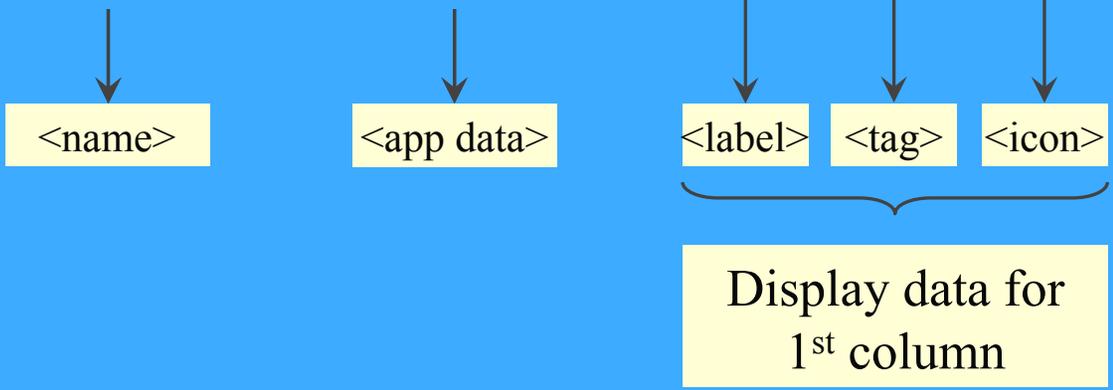
0xAAAAAA {HINST IN 4 -1} TOP branch vlog\_inst



# MtiHierarchy Widget – Callback Support

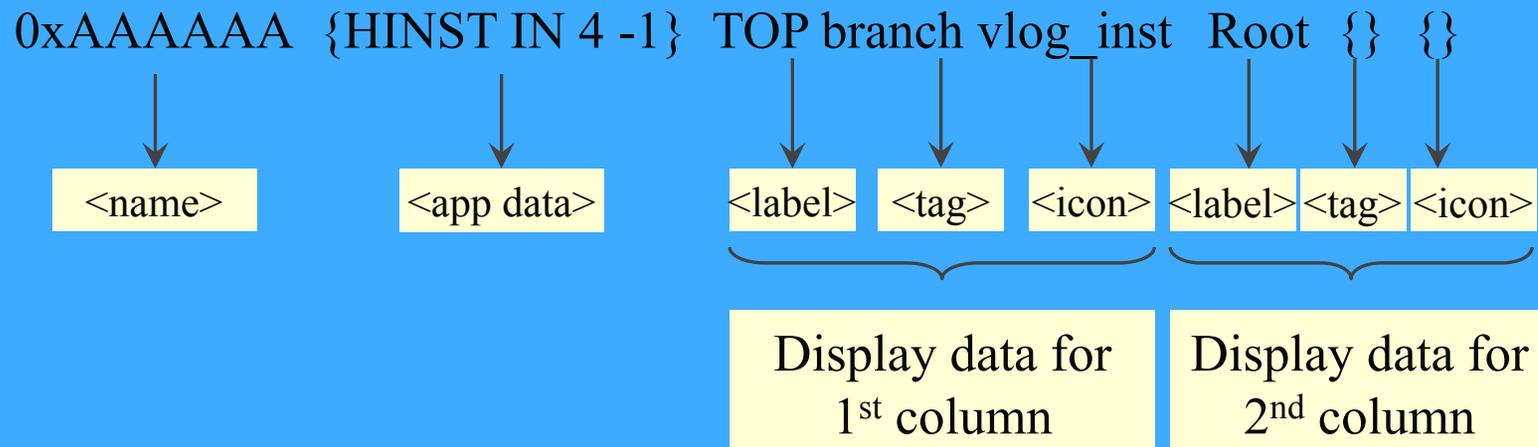
Instance Name	Design Unit	Instances
TOP	root	35

0xAFFFFFFF {HINST IN 4 -1} TOP branch vlog\_inst



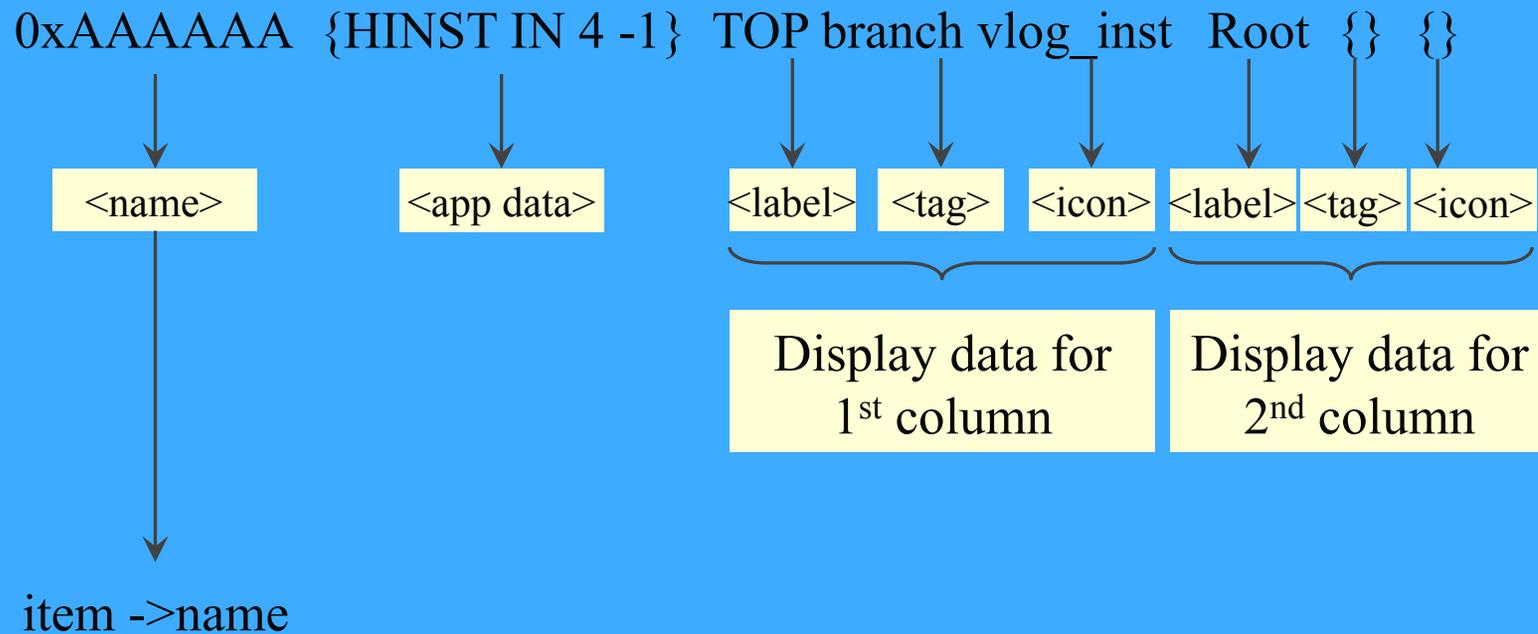
# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



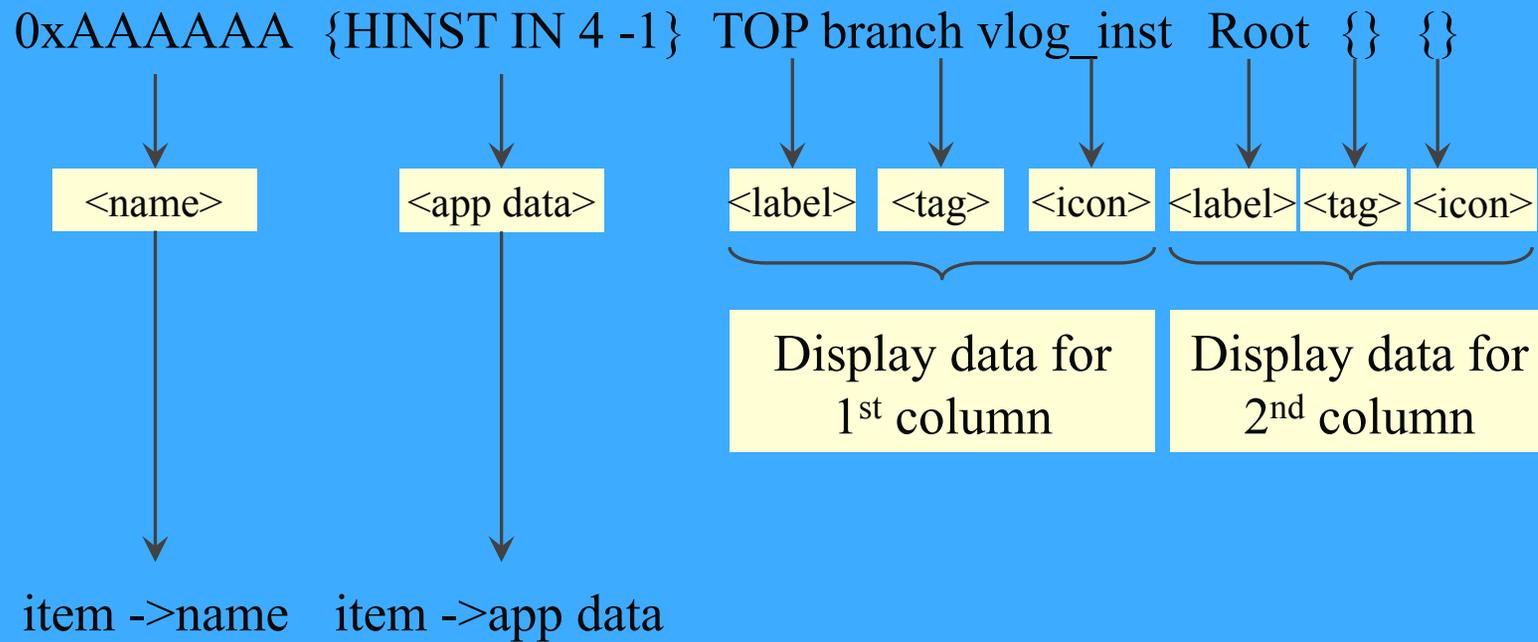
# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



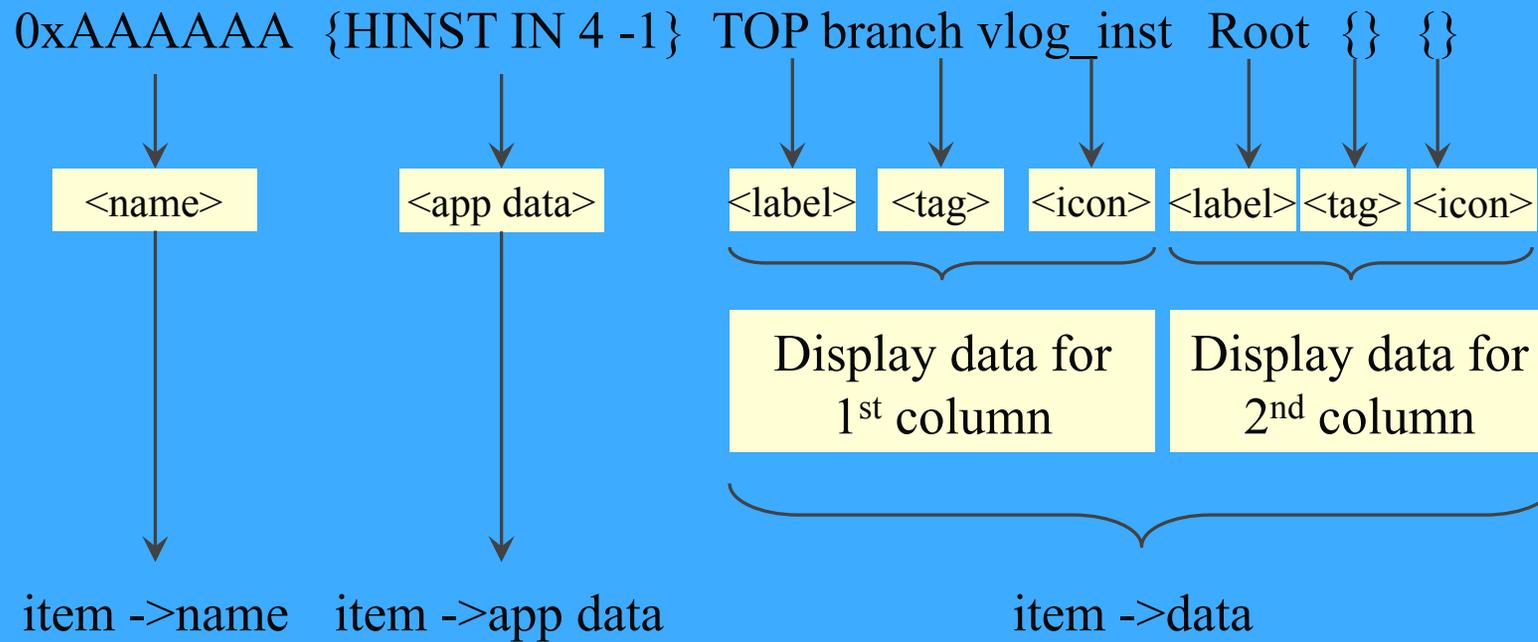
# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



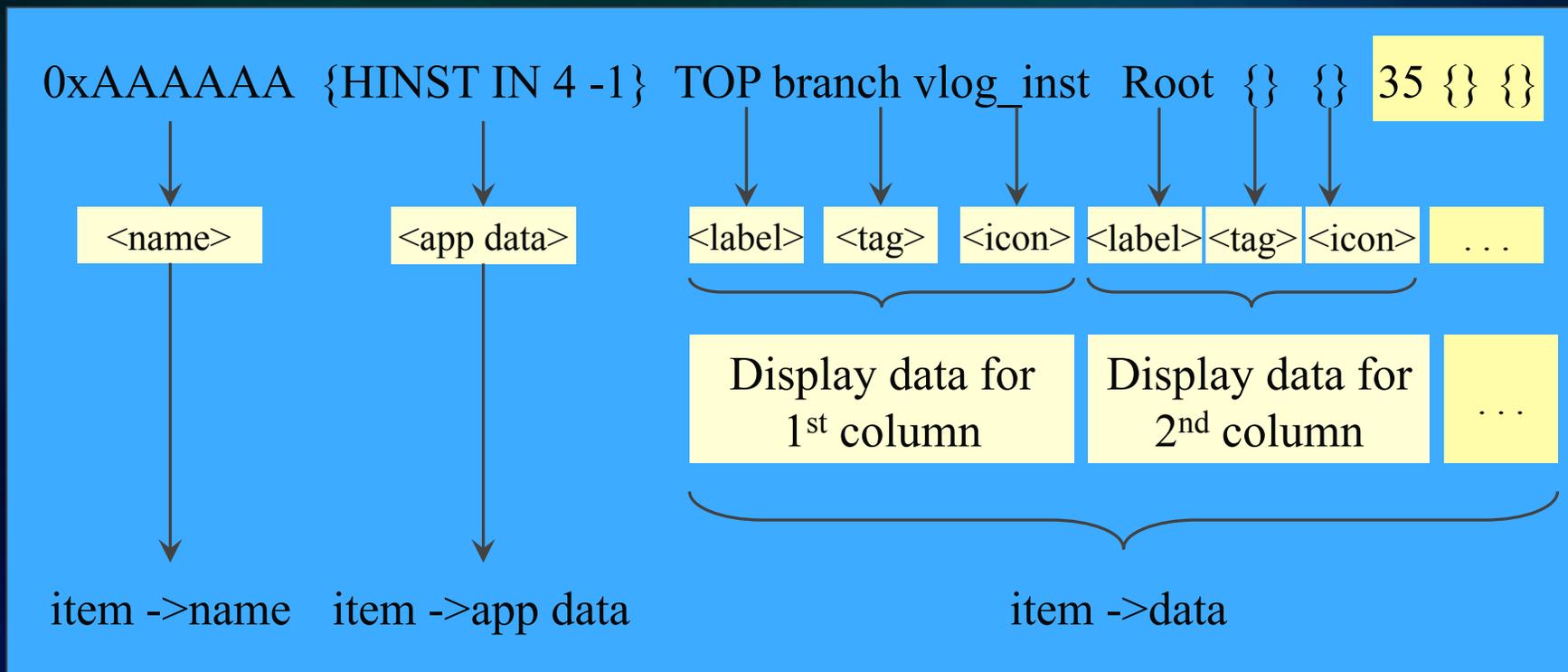
# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35
udram0	ram4	9
dout[0]_tie	TIEX	2
dout[1]_tie	TIEX	2
dout[2]_tie	TIEX	2
dout[3]_tie	TIEX	2
dout[4]_tie	TIEX	2
dout[5]_tie	TIEX	2
dout[6]_tie	TIEX	2

Static Data

<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>

<icon12>...

<name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>

<icon22>...

.

.

<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>

# MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35
udram0	ram4	9
dout[0]_tie	TIEX	2
dout[1]_tie	TIEX	2
dout[2]_tie	TIEX	2
dout[3]_tie	TIEX	2
dout[4]_tie	TIEX	2
dout[5]_tie	TIEX	2
dout[6]_tie	TIEX	2

Static Data

Dynamic Data

```

<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>
<icon12>...
<name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>
<icon22>...
.
.
.
<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>
<iconN2>...
    
```

# MtiHierarchy Widget – Callback Support

Widget stores

Static Data

Dynamic Data

```
<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>  
<icon12>...  
<name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>  
<icon22>...  
.  
.  
.  
<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>  
<iconN2>...
```

# MtiHierarchy Widget – Callback Support

Widget stores

Static Data

```
<name1> <app data1>  
<icon12>...  
<name2> <app data2>  
<icon22>...
```

Widget gets from application at runtime

Dynamic Data

```
<label11> <tag11> <icon11> <label12> <tag12>  
<label21> <tag21> <icon21> <label22> <tag22>  
.  
.  
<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>  
<iconN2>...
```

# MtiHierarchy Widget – Callback Support

- Changes like addition/deletion of a column does not require any iterations by the widget
- Display data is provided by the application - takes fraction of a second

## Widget stores

Static Data

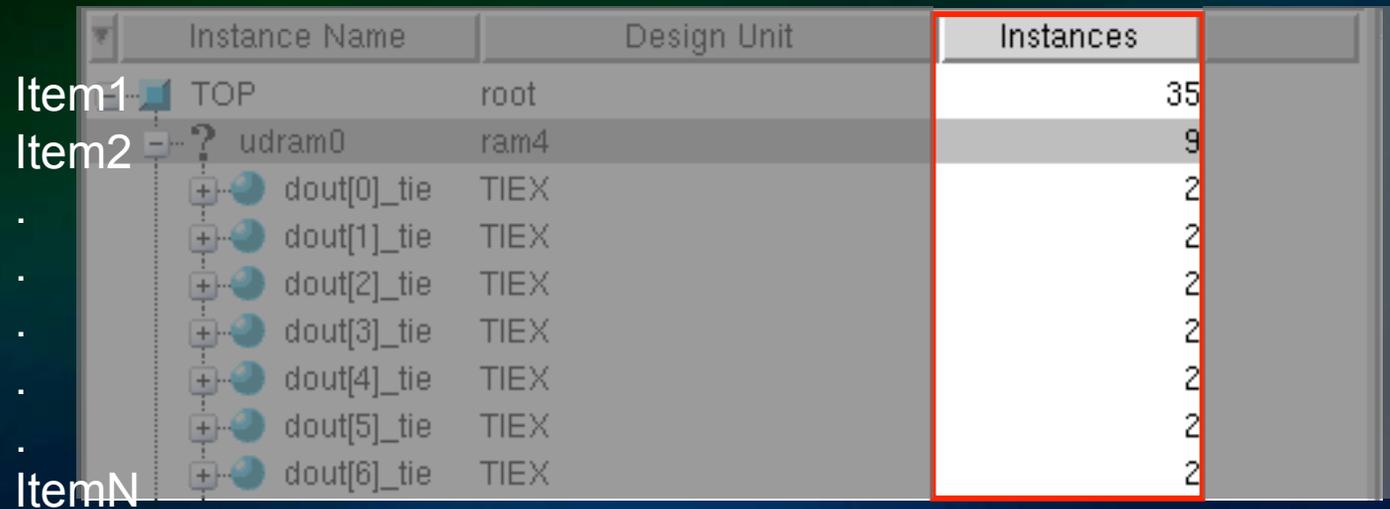
```
<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>  
<icon12>...  
<name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>  
<icon22>...  
.  
.  
.  
<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>  
<iconN2>...
```

## Widget gets from application at runtime

Dynamic Data

# Side Effects

# Side Effects – Sorting



The screenshot shows a table with three columns: 'Instance Name', 'Design Unit', and 'Instances'. The 'Instances' column is highlighted with a red border. The table lists several instances, including 'TOP' (root) with 35 instances, 'udram0' (ram4) with 9 instances, and a series of 'dout[i]\_tie' (TIEX) instances, each with 2 instances. The table is partially obscured by labels 'Item1', 'Item2', and 'ItemN' on the left side.

Instance Name	Design Unit	Instances
Item1 TOP	root	35
Item2 udram0	ram4	9
+ dout[0]_tie	TIEX	2
+ dout[1]_tie	TIEX	2
+ dout[2]_tie	TIEX	2
+ dout[3]_tie	TIEX	2
+ dout[4]_tie	TIEX	2
+ dout[5]_tie	TIEX	2
ItemN + dout[6]_tie	TIEX	2

# Side Effects – Sorting

Instance Name	Design Unit	Instances	
Item1	TOP	root	35
Item2	udram0	ram4	9
	+ dout[0]_tie	TIEX	2
	+ dout[1]_tie	TIEX	2
	+ dout[2]_tie	TIEX	2
	+ dout[3]_tie	TIEX	2
	+ dout[4]_tie	TIEX	2
	+ dout[5]_tie	TIEX	2
ItemN	+ dout[6]_tie	TIEX	2

Static Data

Dynamic Data

<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>  
 <icon12>...  
 <name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>  
 <icon22>...

<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>  
 <iconN2>

# Side Effects – Sorting

Instance Name	Design Unit	Instances	
Item1	TOP	root	35
Item2	udram0	ram4	9
	+ dout[0]_tie	TIEX	2
	+ dout[1]_tie	TIEX	2
	+ dout[2]_tie	TIEX	2
	+ dout[3]_tie	TIEX	2
	+ dout[4]_tie	TIEX	2
	+ dout[5]_tie	TIEX	2
ItemN	+ dout[6]_tie	TIEX	2

Static Data

Dynamic Data

```

<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>
<icon12>...
<name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>
<icon22>...

```

```

<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>
<iconN2>

```

# Side Effects – Sorting

```
<name  
1>  
<name  
2>  
.  
.  
.  
.  
.  
<name  
N>
```

- Application gives us sorted list of "<name>" (item->name pointers)

## Side Effects – Sorting

```
<name  
1>  
<name  
2>  
.  
.  
.  
.
```

```
.  
<name  
N>
```

```
lte  
m1  
lte  
m2  
.  
.  
.  
.
```

```
.  
lte  
mN
```

- Application gives us sorted list of “<name>” (item->name pointers)
- We need the list of sorted “item” pointers.

# Side Effects – Sorting

```
<name  
1>  
<name  
2>  
.  
.  
.  
.  
.  
<name  
N>
```

Item1 ->name	
Item1	
Item2 ->name	
Item2	
.	
.	
.	
.	
.	
ItemN ->name	
ItemN	

- Application gives us sorted list of “<name>” (item->name pointers)
- We need the list of sorted “item” pointers.
- So we create a hash table for item->name and item pointer

# Agenda

## ■ Enhancement in MtiHierarchy widget

- Callback support for improved performance

- Sub column support

- Making tree column frozen in place (non-scrollable)

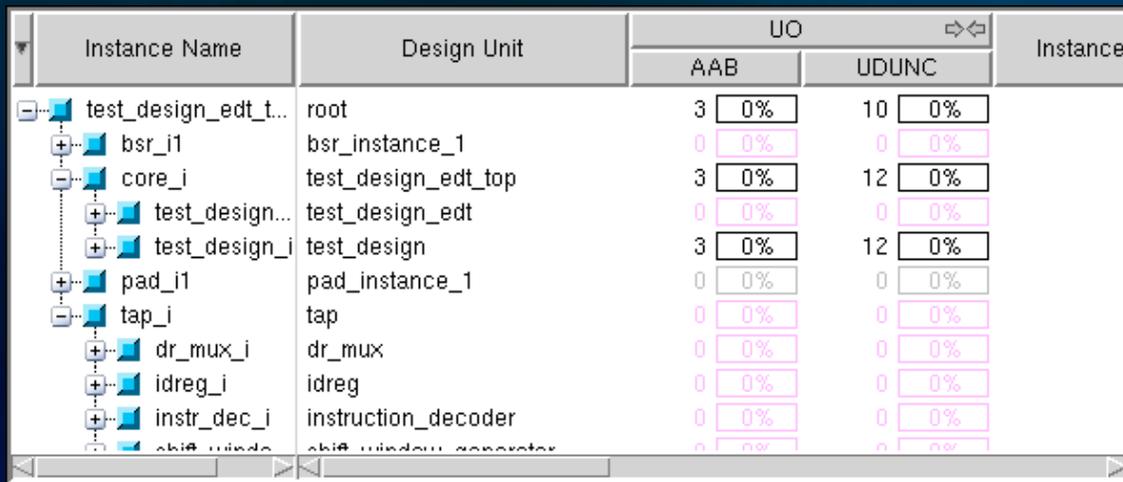
## ■ Enhancement in Text widget

- Hyperlinks

- Incremental Parsing

# Sub-columns

- Show the fault distribution of instances in various categories and sub-categories



Instance Name	Design Unit	UO		Instance:
		AAB	UDUNC	
test_design_edt_t...	root	3 0%	10 0%	
bsr_i1	bsr_instance_1	0 0%	0 0%	
core_i	test_design_edt_top	3 0%	12 0%	
test_design...	test_design_edt	0 0%	0 0%	
test_design_i	test_design	3 0%	12 0%	
pad_i1	pad_instance_1	0 0%	0 0%	
tap_i	tap	0 0%	0 0%	
dr_mux_i	dr_mux	0 0%	0 0%	
idreg_i	idreg	0 0%	0 0%	
instr_dec_i	instruction_decoder	0 0%	0 0%	
shift_windo	shift_window_generator	0 0%	0 0%	

# Sub-columns

- Show the fault distribution of instances in various categories and sub-categories
- Categories and sub-categories should be displayed as in-place expandable/collapsible column.

The screenshot displays a software interface with a hierarchical tree on the left and a summary table on the right. The tree shows a root node 'test\_design\_edt\_t...' with several sub-nodes like 'bsr\_i1', 'core\_i', 'test\_design...', 'pad\_i1', 'tap\_i', 'dr\_mux\_i', 'idreg\_i', 'instr\_dec\_i', and 'shift\_windo...'. The summary table has columns for 'Instance Name', 'Design Unit', 'UO' (with sub-columns 'AAB' and 'UDUNC'), and 'Instances'. The 'UO' column contains percentage values, some highlighted in red (100%, 92%, 92%) and others in pink (0%, 0%, 0%, 0%, 0%, 0%, 0%, 0%, 0%).

Instance Name	Design Unit	UO		Instances
		AAB	UDUNC	
test_design_edt_t...	root	3 0%	10 0%	4
bsr_i1	bsr_instance_1	0 0%	0 0%	91
core_i	test_design_edt_top	3 0%	12 0%	2
test_design...	test_design_edt	0 0%	0 0%	3
test_design_i	test_design	0 0%	0 0%	30
pad_i1	pad_instance_1	0 0%	0 0%	63
tap_i	tap	NF 0%	NF 0%	16
dr_mux_i	dr_mux	NF 0%	NF 0%	3
idreg_i	idreg	NF 0%	NF 0%	76
instr_dec_i	instruction_decoder	NF 0%	NF 0%	9
shift_windo...	shift_window_generator	NF 0%	NF 0%	2
tan_ctrl_i	tan_ctrl	NF 0%	NF 0%	61

# Sub-columns

- Overview of changes

- Enhanced MtiHeirarchy widget protocol to support the sub-column labels as a list of labels instead of column label

`<name> <appdata> <sub-label1 sub-label2 sub-label3> <tag> <icon>`

# Sub-columns

## ■ Overview of changes

- Enhanced MtiHeirarchy widget protocol to support the sub-column labels as a list of labels instead of column label

`<name> <appdata> <sub-label1 sub-label2 sub-label3> <tag> <icon>`

- Provide APIs for sub-column operations like add, delete, configure, move, etc

# Sub-columns

## ■ Overview of changes

- Enhanced MtiHeirarchy widget protocol to support the sub-column labels as a list of labels instead of column label

`<name> <appdata> <sub-label1 sub-label2 sub-label3> <tag> <icon>`

- Provide APIs for sub-column operations like add, delete, configure, move, etc
- Expandable/ collapsible we need to provide a clickable icon

# Sub-columns – Approach

Instance Name	Design Unit	UO				Instance:
		AAB		UDUNC		
test_design_edt_t...	root	3	0%	10	0%	
bsr_i1	bsr_instance_1	0	0%	0	0%	
core_i	test_design_edt_top	3	0%	12	0%	
test_design...	test_design_edt	0	0%	0	0%	
test_design_i	test_design	3	0%	12	0%	
pad_i1	pad_instance_1	0	0%	0	0%	
tap_i	tap	0	0%	0	0%	
dr_mux_i	dr_mux	0	0%	0	0%	
idreg_i	idreg	0	0%	0	0%	
instr_dec_i	instruction_decoder	0	0%	0	0%	
shift_winde	shift_window_generator	0	0%	0	0%	

# Sub-columns – Approach

Instance Name	Design Unit	UO		Instance:	
		AAB	UDUNC		
test_design_edt_t...	root	3	10	0%	0%
bsr_i1	bsr_instance_1	0	0	0%	0%
core_i	test_design_edt_top	3	12	0%	0%
test_design...	test_design_edt	0	0	0%	0%
test_design_i	test_design	3	12	0%	0%
pad_i1	pad_instance_1	0	0	0%	0%
tap_i	tap	0	0	0%	0%
dr_mux_i	dr_mux	0	0	0%	0%
idreg_i	idreg	0	0	0%	0%
instr_dec_i	instruction_decoder	0	0	0%	0%
shift_winde	shift_window_generator	0	0	0%	0%

# Sub-columns – Approach

Instance Name	Design Unit	UO		Instance:	
		AAB	UDUNC		
test_design_edt_t...	root	3	10	0%	0%
bsr_i1	bsr_instance_1	0	0	0%	0%
core_i	test_design_edt_top	3	12	0%	0%
test_design...	test_design_edt	0	0	0%	0%
test_design_i	test_design	3	12	0%	0%
pad_i1	pad_instance_1	0	0	0%	0%
tap_i	tap	0	0	0%	0%
dr_mux_i	dr_mux	0	0	0%	0%
idreg_i	idreg	0	0	0%	0%
instr_dec_i	instruction_decoder	0	0	0%	0%
shift_winde	shift_window_generator	0	0	0%	0%

# Agenda

## ■ Enhancement in MtiHierarchy widget

- Callback support for improved performance
- Sub column support

- Making tree column frozen in place (non-scrollable)

## ■ Enhancement in Text widget

- Hyperlinks
- Incremental Parsing

# Frozen Tree Column

Instance Name	Design Unit	Instances	UO		Primitive	
			AAB	UDUNC		
[-] test_design_edt_t...		4	3	0%	10	0%
[+] bsr_i1	e_1	91	0	0%	0	0%
[-] core_i	_edt_top	2	3	0%	12	0%
[+] test_design..._edt		3	0	0%	0	0%
[+] test_design_i		30	3	0%	12	0%
[+] pad_i1	ce_1	63	0	0%	0	0%
[-] tap_i		16	0	0%	0	0%
[+] dr_mux_i		3	0	0%	0	0%
[+] idreg_i		76	0	0%	0	0%
[+] instr_dec_i	decoder	9	0	0%	0	0%
[-] shift_windo	u_generator	2	0	0%	0	0%

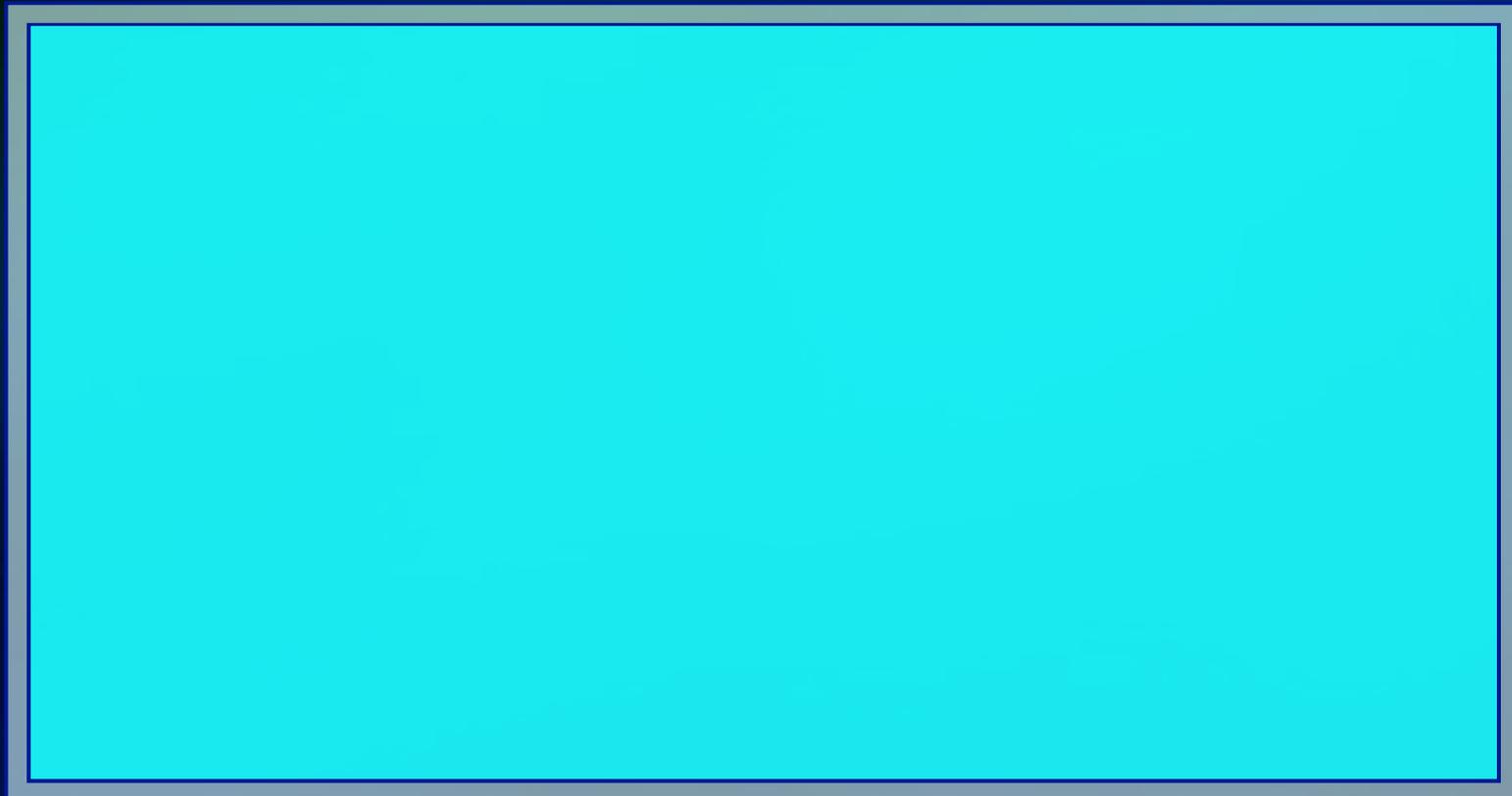
# Frozen Tree Column

- Clipper (a frame)



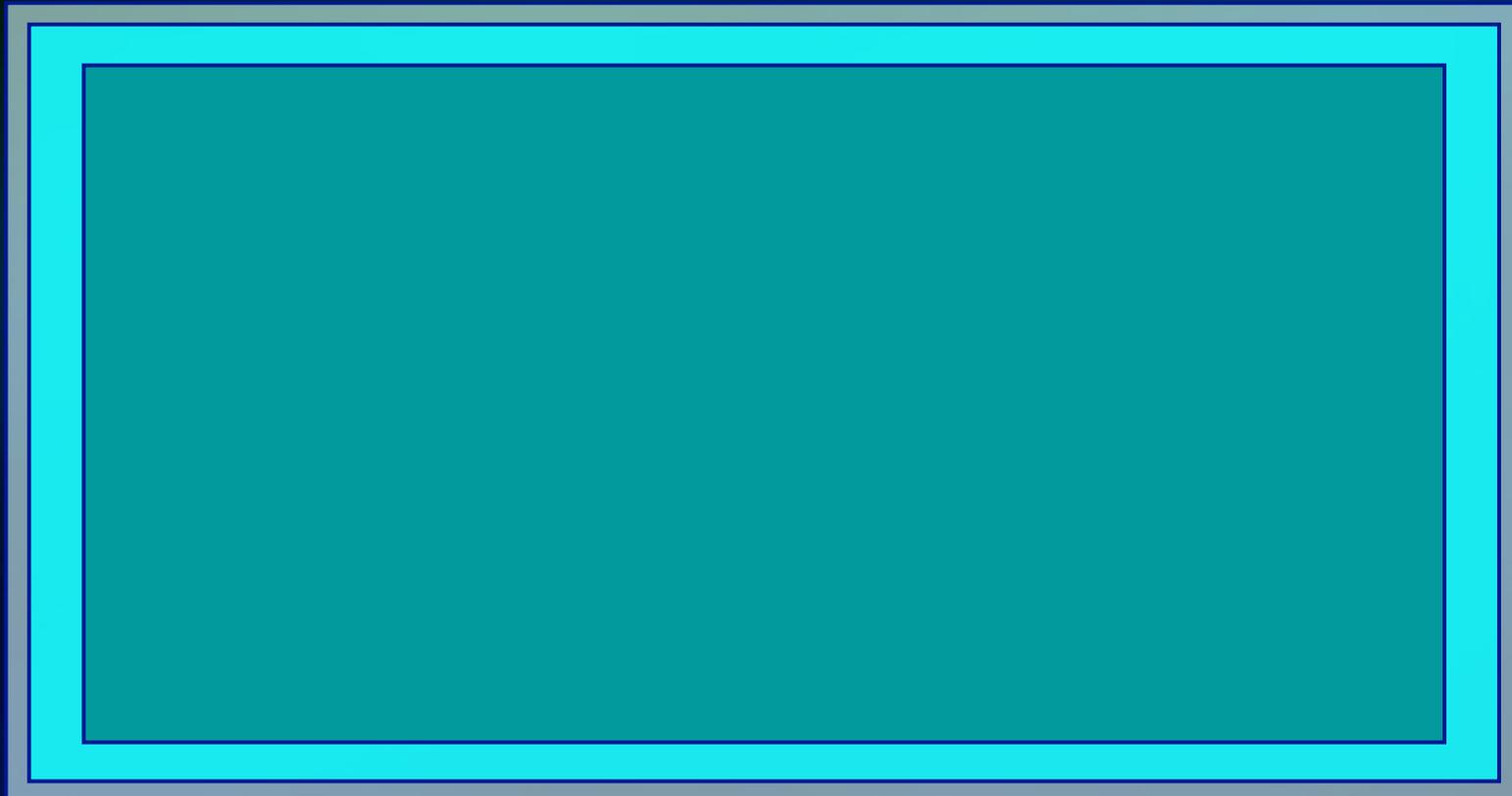
# Frozen Tree Column

- Clipper (a frame) → Canvas



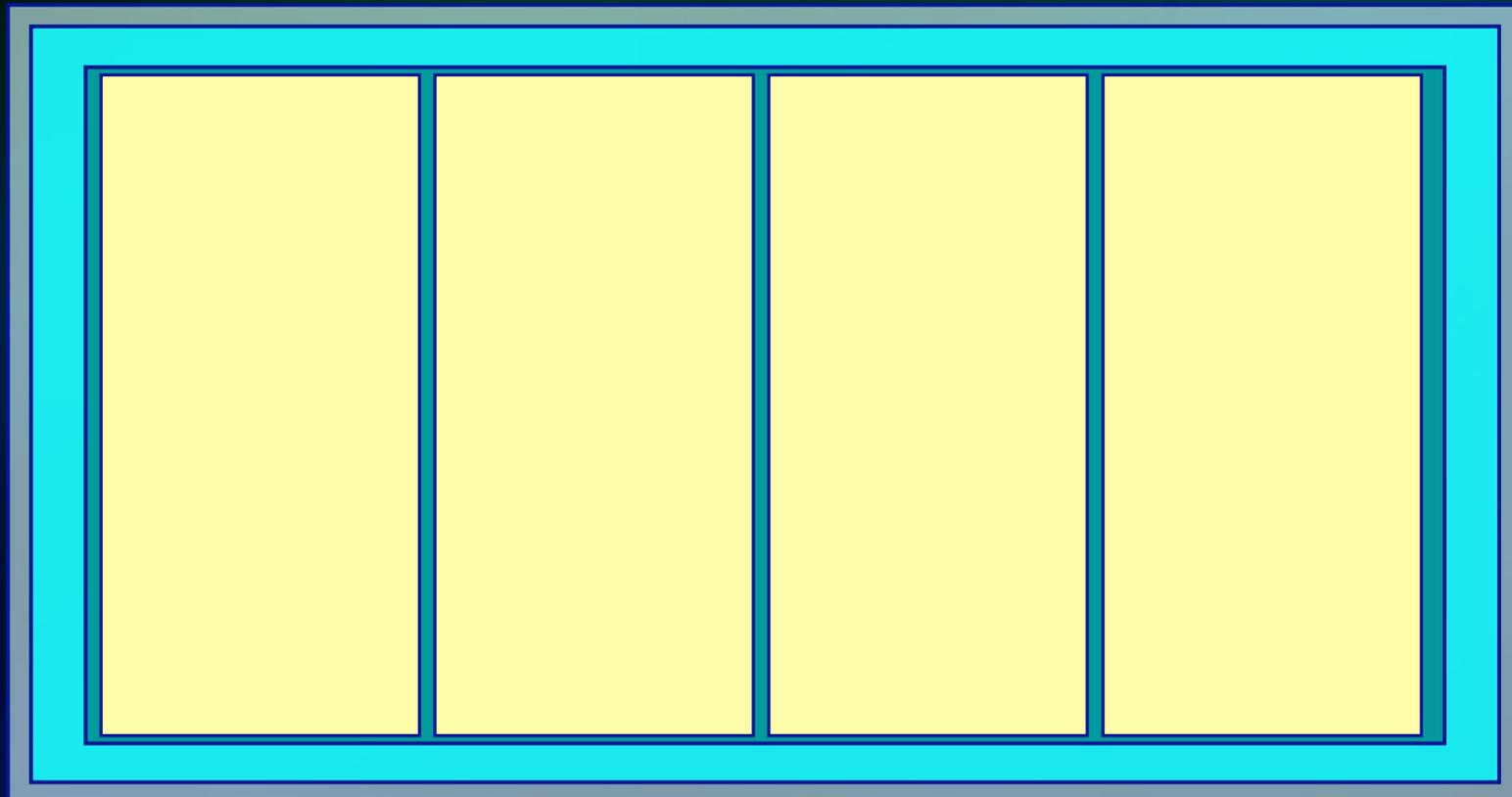
# Frozen Tree Column

- Clipper (a frame) → Canvas → Sfschildsite (a frame)



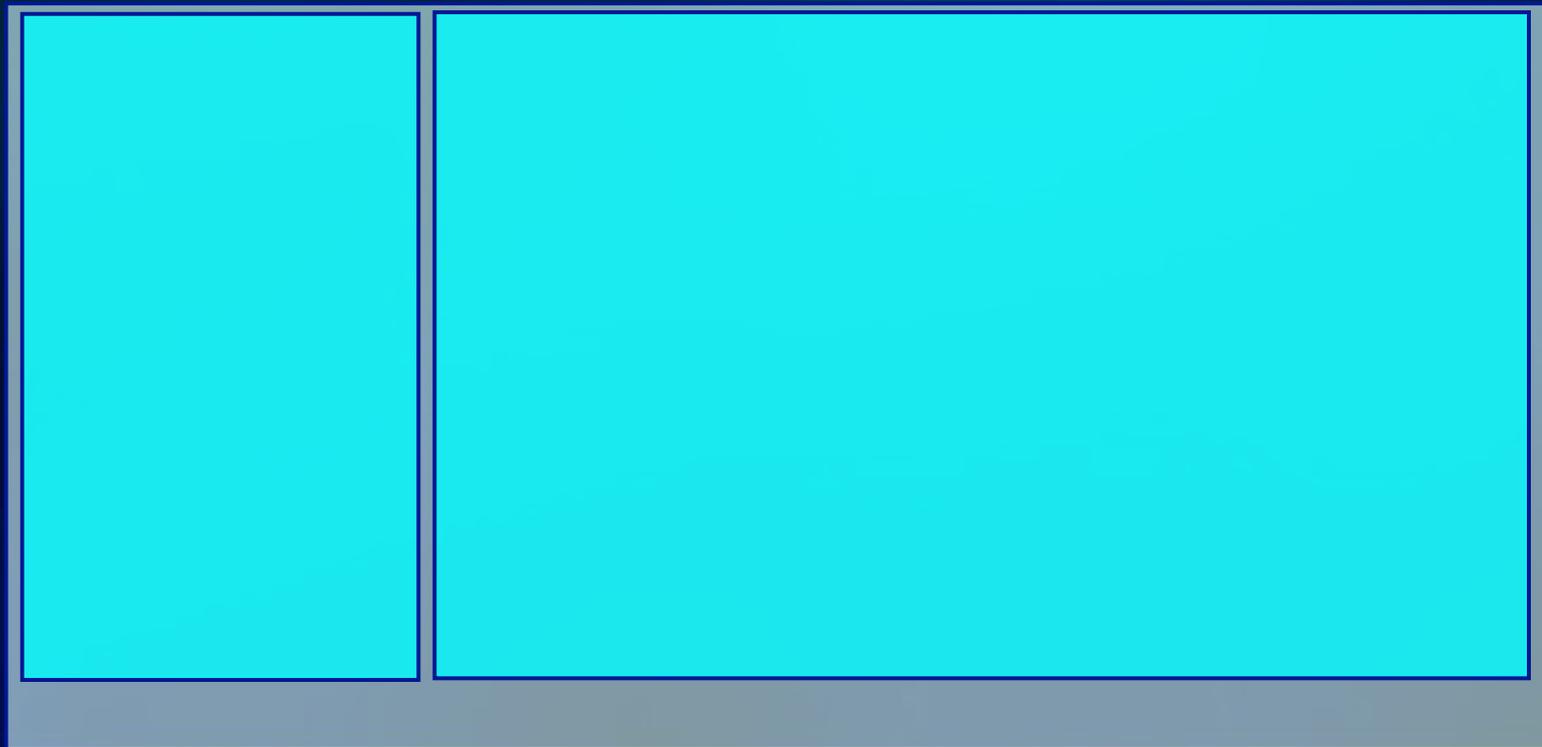
# Frozen Tree Column

- Clipper (a frame) → Canvas → Sfchildsite (a frame) → Columns



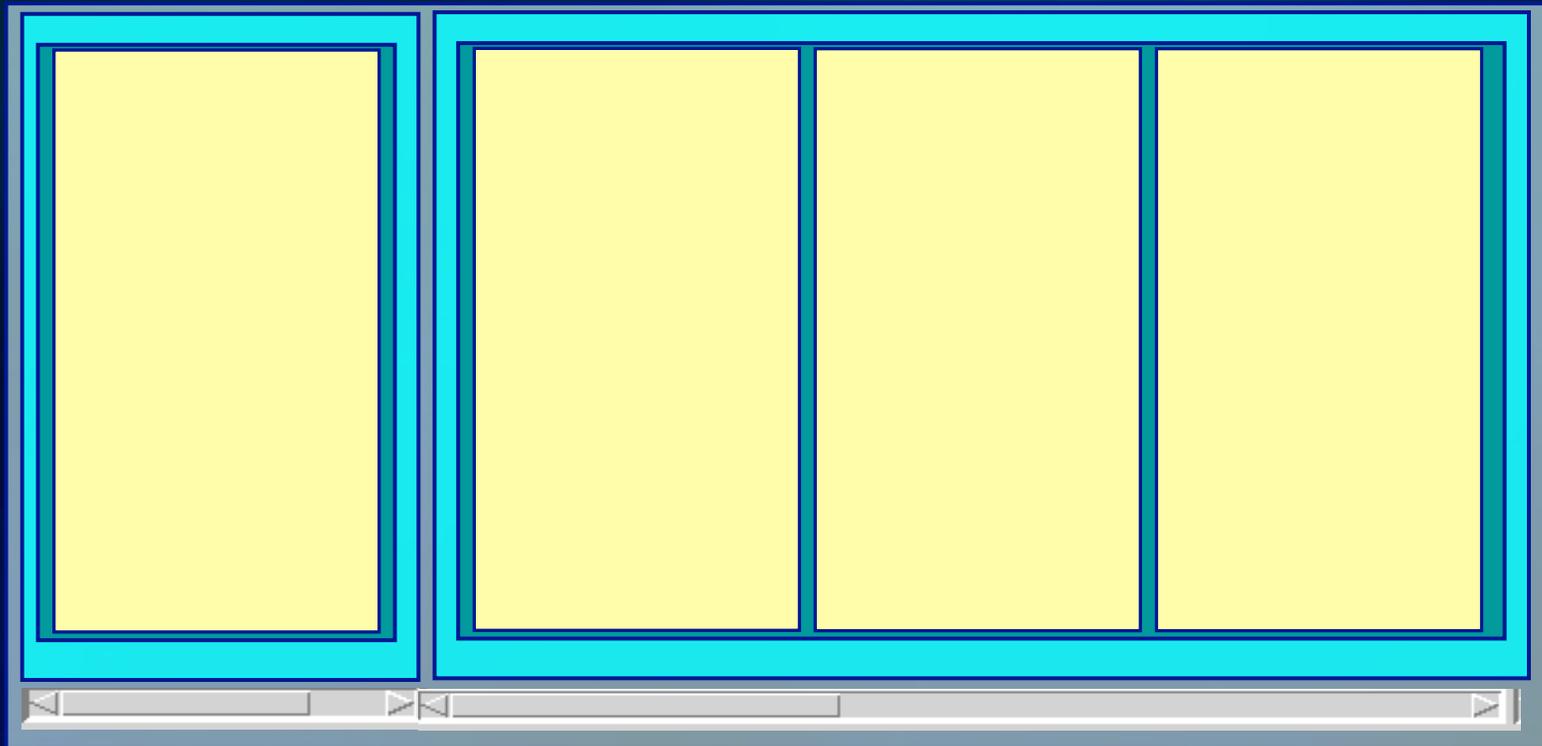
# Frozen Tree Column

- Create two canvases and grid them inside the clipper as column 0 and column 1 of row 0



# Frozen Tree Column

- 1st scrollbar alters the xview of the column 0 such that it scrolls the instance names of the tree column and not the column itself.
- 2nd scrollbar alters the xview of the 2nd canvas so that the rest of the columns scroll past the tree column.



# Agenda

- Enhancement in MtiHierarchy widget
  - Callback support for improved performance
  - Sub column support
  - Making tree column frozen in place (non-scrollable)
- Enhancement in Text widget
  - Hyperlinks
  - Incremental Parsing

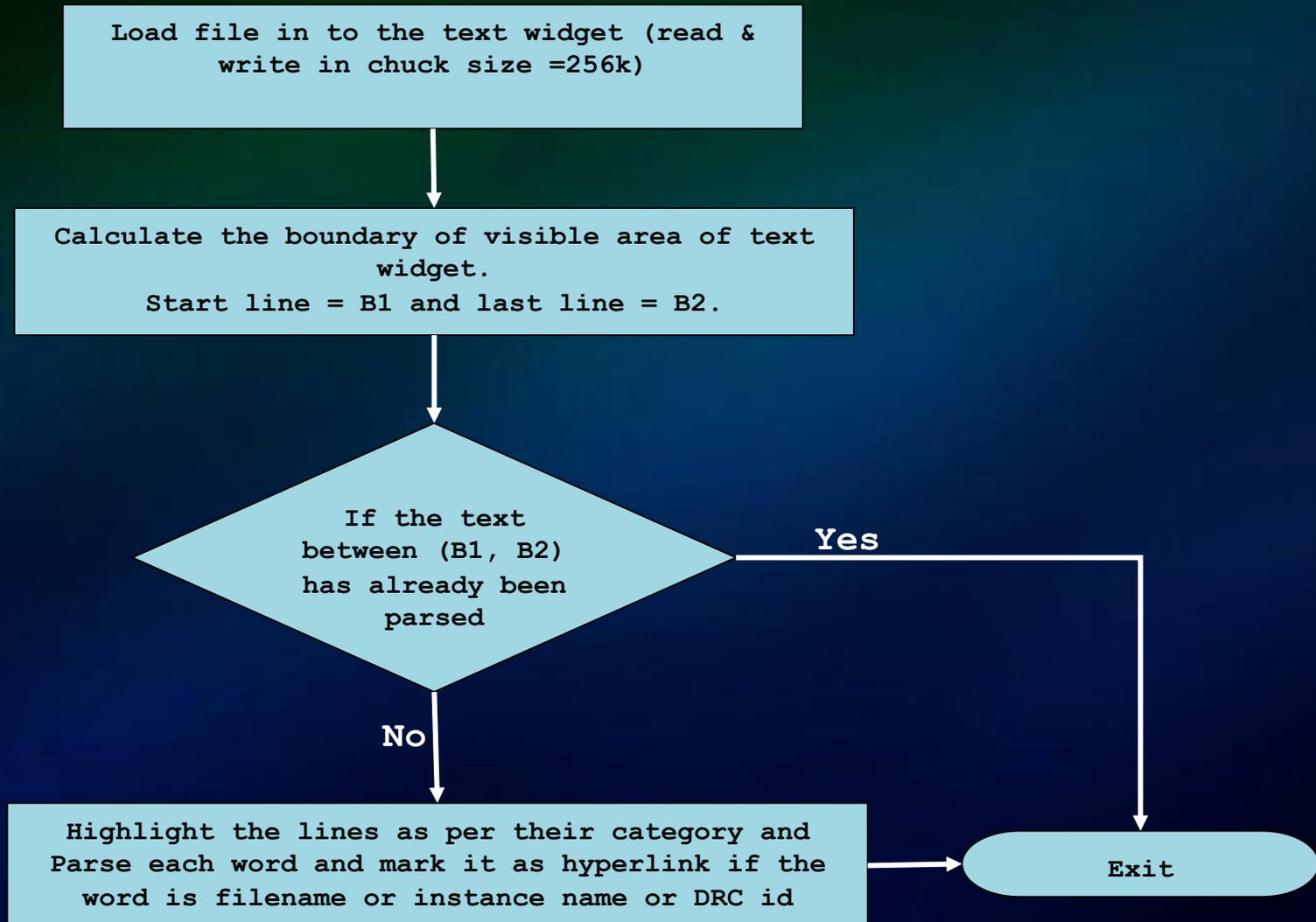
# Enhanced Text Widget

- Text Highlighting
  - Text highlighting helps to point out the issues easily
- Hyperlinks
  - Get to the relevant file or information in other windows of the tool to debug the issues very quickly.
- **Incremental parsing**

# Enhanced Text Widget

```
Transcript
// Reading group test procedure file stimulus/clock_c3_c4.g1.
// Simulating load/unload procedure in g1 test procedure file.
// Chain = c1 successfully traced with scan_cells = 2.
// 2 scan cells have been identified in 1 scan chain.
// Scan group g1 successfully passed master_observe procedure audit.
// -----
// Begin transparent latch checking for 51 latches.
// -----
// Warning: 22 latches not transparent due to all clocks off. (D6)
// 1 TLAs are involved in feedback networks.
// Number transparent latches = 29.
// 1 feedback networks were identified.
// -----
// Begin scan clock rules checking.
// -----
// 3 scan clock/set/reset lines have been identified.
// All scan clocks successfully passed off-state check.
// 22 sequential cells passed clock stability checking.
// All scan clocks successfully passed capture ability check.
// Error: Clock /PH1 failed rule c3 on input 3 of /I_8516_I_582 (560). (C3-1)
// Source of violation: input 2 of /I_9415_1 (583).
// Error: Rules checking unsuccessful, cannot exit SETUP mode.
// command: set gate level primitive
// command: analyze drc violation c3-1 -display
// Note: Gate report now set to clock_cone (clock=/PH1).
SETUP> // 'DOFile ./stimulus/test.do' aborted at line 17
```

# Enhanced Text Widget – Incremental Parsing



**Thank you.**