

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)' component, showing its inputs and outputs.
- Wave:** A timing diagram showing the signals 'edt_clock', 'edt_update', and 'edt_channels_in' over time. The current time is 56 ns, and a cursor is positioned at 0 ns.
- Transcript:** A log of ATPG commands and their outputs, including failure reports for rule C3 and C4.
- Debug:** A detailed schematic diagram of the decompressor component, showing its internal logic and components.

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_update', and 'edt_channels_in', and outputs like 'edt_mask' and 'edt_scan_in(size=4)'.
- Wave:** A waveform viewer showing signals over time. The 'cycle' column shows values 6, 7, 8, 9. The 'Now' time is 56 ns.
- Transcript:** A log of ATPG commands and responses, including coverage rules and instance display commands.
- Debug:** A detailed circuit diagram of a portion of the design, showing logic gates, flip-flops, and multiplexers. This window 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
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 – Browser Window

The screenshot displays the DFTVisualizer interface for design.v (Tessent FastScan v9.2-prerelease). The Browser window is highlighted with a red border and contains the following table:

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

The Design window shows a block diagram of test_design_edt_decompressor_i (51) with inputs edt_clock, edt_update, and edt_channels_in, and outputs edt_mask and edt_scan_in(size=4).

The Wave window displays timing signals for cycle (0, 6, 7, 8, 9), edt..., and edat. The current time is 56 ns, and the cursor is at 0 ns.

The Debug window shows a logic diagram with various components like mux2L, and0, and1, and2, and3, and4, and5, and6, and7, and8, and9, and10, and11, and12, and13, and14, and15, and16, and17, and18, and19, and20, and21, and22, and23, and24, and25, and26, and27, and28, and29, and30, and31, and32, and33, and34, and35, and36, and37, and38, and39, and40, and41, and42, and43, and44, and45, and46, and47, and48, and49, and50, and51, and52, and53, and54, and55, and56, and57, and58, and59, and60, and61, and62, and63, and64, and65, and66, and67, and68, and69, and70, and71, and72, and73, and74, and75, and76, and77, and78, and79, and80, and81, and82, and83, and84, and85, and86, and87, and88, and89, and90, and91, and92, and93, and94, and95, and96, and97, and98, and99, and100, and101, and102, and103, and104, and105, and106, and107, and108, and109, and110, and111, and112, and113, and114, and115, and116, and117, and118, and119, and120, and121, and122, and123, and124, and125, and126, and127, and128, and129, and130, and131, and132, and133, and134, and135, and136, and137, and138, and139, and140, and141, and142, and143, and144, and145, and146, and147, and148, and149, and150, and151, and152, and153, and154, and155, and156, and157, and158, and159, and160, and161, and162, and163, and164, and165, and166, and167, and168, and169, and170, and171, and172, and173, and174, and175, and176, and177, and178, and179, and180, and181, and182, and183, and184, and185, and186, and187, and188, and189, and190, and191, and192, and193, and194, and195, and196, and197, and198, and199, and200, and201, and202, and203, and204, and205, and206, and207, and208, and209, and210, and211, and212, and213, and214, and215, and216, and217, and218, and219, and220, and221, and222, and223, and224, and225, and226, and227, and228, and229, and230, and231, and232, and233, and234, and235, and236, and237, and238, and239, and240, and241, and242, and243, and244, and245, and246, and247, and248, and249, and250, and251, and252, and253, and254, and255, and256, and257, and258, and259, and260, and261, and262, and263, and264, and265, and266, and267, and268, and269, and270, and271, and272, and273, and274, and275, and276, and277, and278, and279, and280, and281, and282, and283, and284, and285, and286, and287, and288, and289, and290, and291, and292, and293, and294, and295, and296, and297, and298, and299, and300, and301, and302, and303, and304, and305, and306, and307, and308, and309, and310, and311, and312, and313, and314, and315, and316, and317, and318, and319, and320, and321, and322, and323, and324, and325, and326, and327, and328, and329, and330, and331, and332, and333, and334, and335, and336, and337, and338, and339, and340, and341, and342, and343, and344, and345, and346, and347, and348, and349, and350, and351, and352, and353, and354, and355, and356, and357, and358, and359, and360, and361, and362, and363, and364, and365, and366, and367, and368, and369, and370, and371, and372, and373, and374, and375, and376, and377, and378, and379, and380, and381, and382, and383, and384, and385, and386, and387, and388, and389, and390, and391, and392, and393, and394, and395, and396, and397, and398, and399, and400, and401, and402, and403, and404, and405, and406, and407, and408, and409, and410, and411, and412, and413, and414, and415, and416, and417, and418, and419, and420, and421, and422, and423, and424, and425, and426, and427, and428, and429, and430, and431, and432, and433, and434, and435, and436, and437, and438, and439, and440, and441, and442, and443, and444, and445, and446, and447, and448, and449, and450, and451, and452, and453, and454, and455, and456, and457, and458, and459, and460, and461, and462, and463, and464, and465, and466, and467, and468, and469, and470, and471, and472, and473, and474, and475, and476, and477, and478, and479, and480, and481, and482, and483, and484, and485, and486, and487, and488, and489, and490, and491, and492, and493, and494, and495, and496, and497, and498, and499, and500, and501, and502, and503, and504, and505, and506, and507, and508, and509, and510, and511, and512, and513, and514, and515, and516, and517, and518, and519, and520, and521, and522, and523, and524, and525, and526, and527, and528, and529, and530, and531, and532, and533, and534, and535, and536, and537, and538, and539, and540, and541, and542, and543, and544, and545, and546, and547, and548, and549, and550, and551, and552, and553, and554, and555, and556, and557, and558, and559, and560, and561, and562, and563, and564, and565, and566, and567, and568, and569, and570, and571, and572, and573, and574, and575, and576, and577, and578, and579, and580, and581, and582, and583, and584, and585, and586, and587, and588, and589, and590, and591, and592, and593, and594, and595, and596, and597, and598, and599, and600, and601, and602, and603, and604, and605, and606, and607, and608, and609, and610, and611, and612, and613, and614, and615, and616, and617, and618, and619, and620, and621, and622, and623, and624, and625, and626, and627, and628, and629, and630, and631, and632, and633, and634, and635, and636, and637, and638, and639, and640, and641, and642, and643, and644, and645, and646, and647, and648, and649, and650, and651, and652, and653, and654, and655, and656, and657, and658, and659, and660, and661, and662, and663, and664, and665, and666, and667, and668, and669, and670, and671, and672, and673, and674, and675, and676, and677, and678, and679, and680, and681, and682, and683, and684, and685, and686, and687, and688, and689, and690, and691, and692, and693, and694, and695, and696, and697, and698, and699, and700, and701, and702, and703, and704, and705, and706, and707, and708, and709, and710, and711, and712, and713, and714, and715, and716, and717, and718, and719, and720, and721, and722, and723, and724, and725, and726, and727, and728, and729, and730, and731, and732, and733, and734, and735, and736, and737, and738, and739, and740, and741, and742, and743, and744, and745, and746, and747, and748, and749, and750, and751, and752, and753, and754, and755, and756, and757, and758, and759, and760, and761, and762, and763, and764, and765, and766, and767, and768, and769, and770, and771, and772, and773, and774, and775, and776, and777, and778, and779, and780, and781, and782, and783, and784, and785, and786, and787, and788, and789, and790, and791, and792, and793, and794, and795, and796, and797, and798, and799, and800, and801, and802, and803, and804, and805, and806, and807, and808, and809, and810, and811, and812, and813, and814, and815, and816, and817, and818, and819, and820, and821, and822, and823, and824, and825, and826, and827, and828, and829, and830, and831, and832, and833, and834, and835, and836, and837, and838, and839, and840, and841, and842, and843, and844, and845, and846, and847, and848, and849, and850, and851, and852, and853, and854, and855, and856, and857, and858, and859, and860, and861, and862, and863, and864, and865, and866, and867, and868, and869, and870, and871, and872, and873, and874, and875, and876, and877, and878, and879, and880, and881, and882, and883, and884, and885, and886, and887, and888, and889, and890, and891, and892, and893, and894, and895, and896, and897, and898, and899, and900, and901, and902, and903, and904, and905, and906, and907, and908, and909, and910, and911, and912, and913, and914, and915, and916, and917, and918, and919, and920, and921, and922, and923, and924, and925, and926, and927, and928, and929, and930, and931, and932, and933, and934, and935, and936, and937, and938, and939, and940, and941, and942, and943, and944, and945, and946, and947, and948, and949, and950, and951, and952, and953, and954, and955, and956, and957, and958, and959, and960, and961, and962, and963, and964, and965, and966, and967, and968, and969, and970, and971, and972, and973, and974, and975, and976, and977, and978, and979, and980, and981, and982, and983, and984, and985, and986, and987, and988, and989, and990, and991, and992, and993, and994, and995, and996, and997, and998, and999, and1000.

The Transcript window shows the following commands and output:

```
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
)
```

The status bar at the bottom indicates: Pin: /core_i/test_design_edt_i/test_design_edt_decompressor_i/edt_updat Instance Grouping : OFF

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...
- Transcript:** A window showing ATPG commands and their outputs, including coverage reports for rules C3 and C4.
- 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 data for various instances.
- Design:** A block diagram of the test_design_edt_decompressor_i (51) component.
- Wave:** A timing diagram showing signals over cycles.
- Transcript:** A log of ATPG commands and their outputs.
- Debug:** A detailed circuit diagram of the decompressor logic.

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
)
```

Text Widgets

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

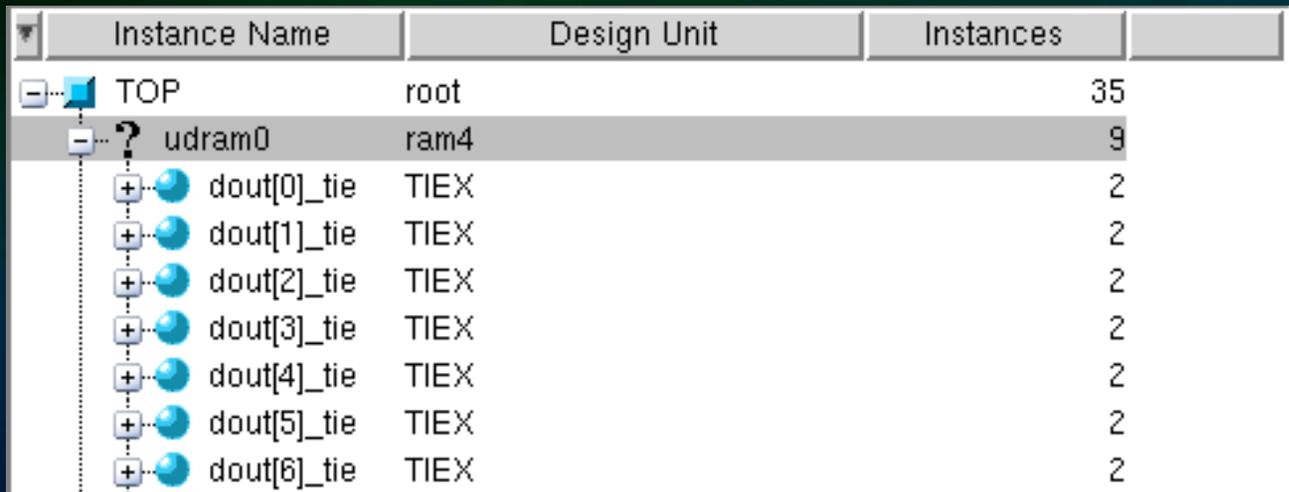
- Making tree column frozen in place (non-scrollable)

■ Enhancement in Text widget

- Hyperlinks

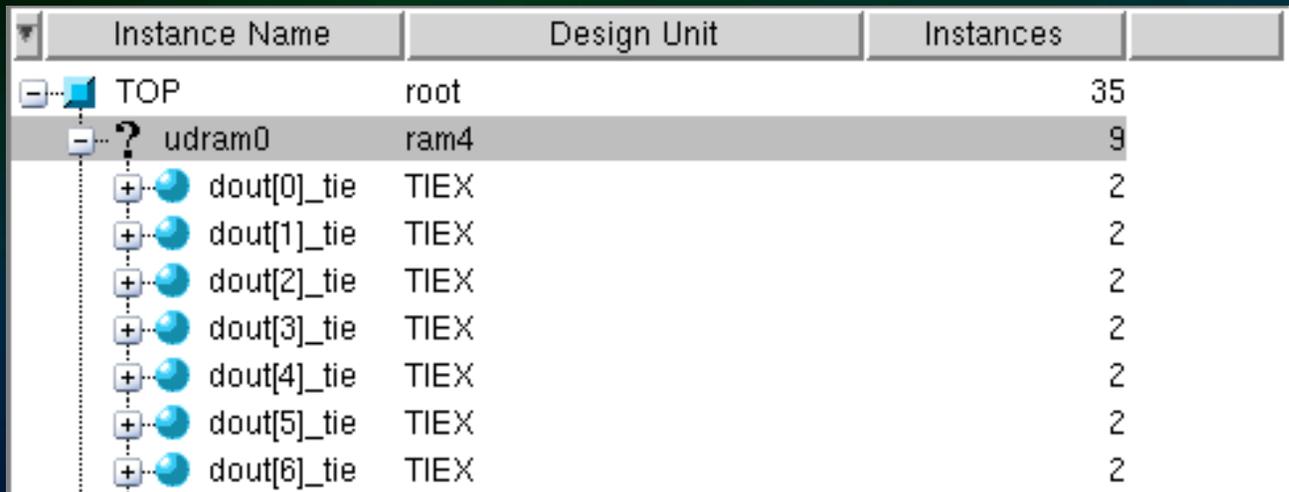
- Incremental Parsing

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

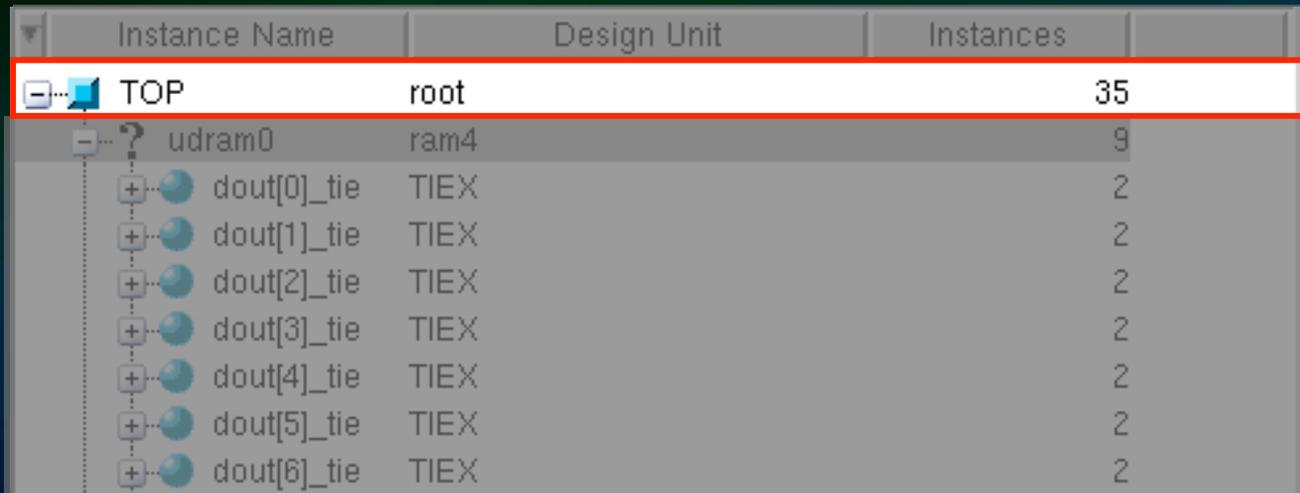
Browser Window – MtiHierarchy Widget



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

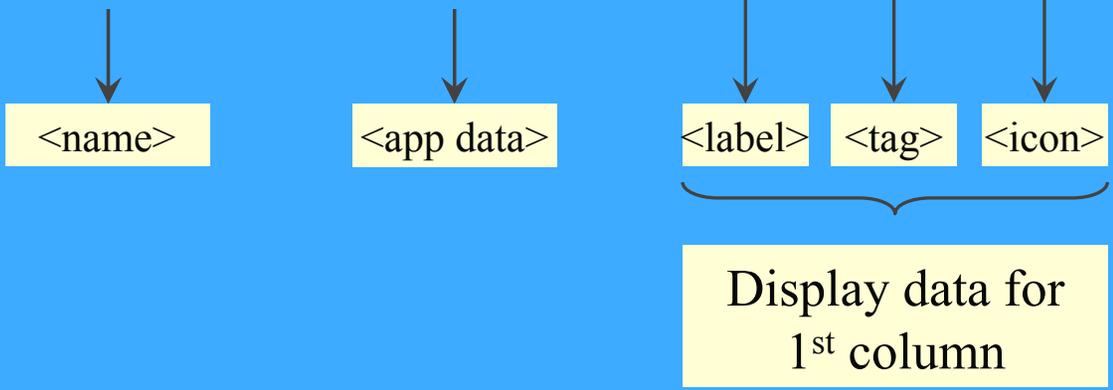
0xAAAAAA {HINST IN 4 -1} TOP branch vlog_inst



MtiHierarchy Widget – Callback Support

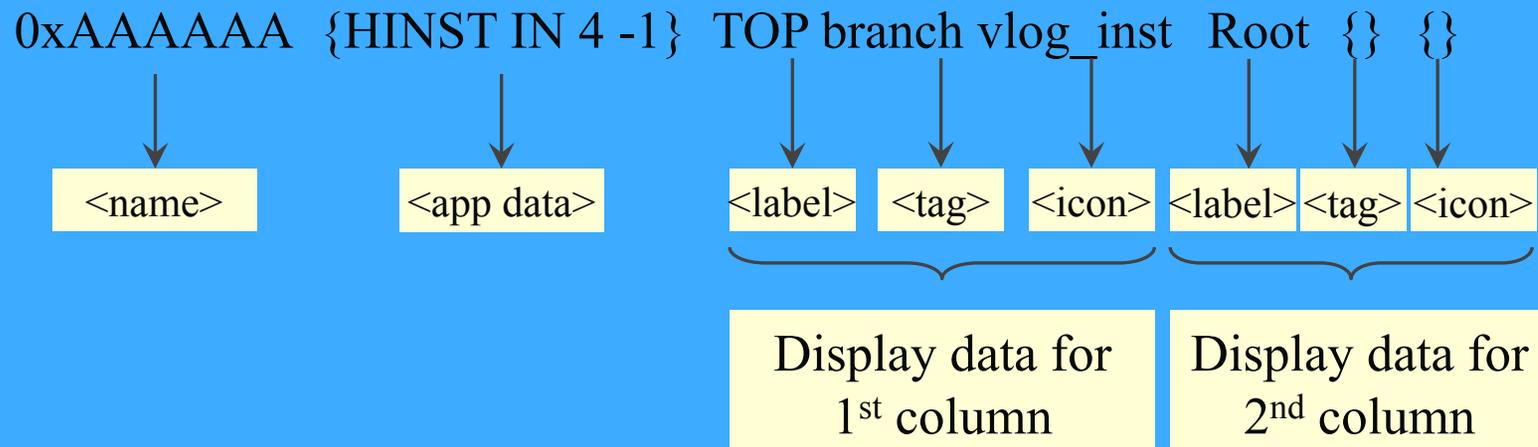
Instance Name	Design Unit	Instances
TOP	root	35

0xAAAAAA {HINST IN 4 -1} TOP branch vlog_inst



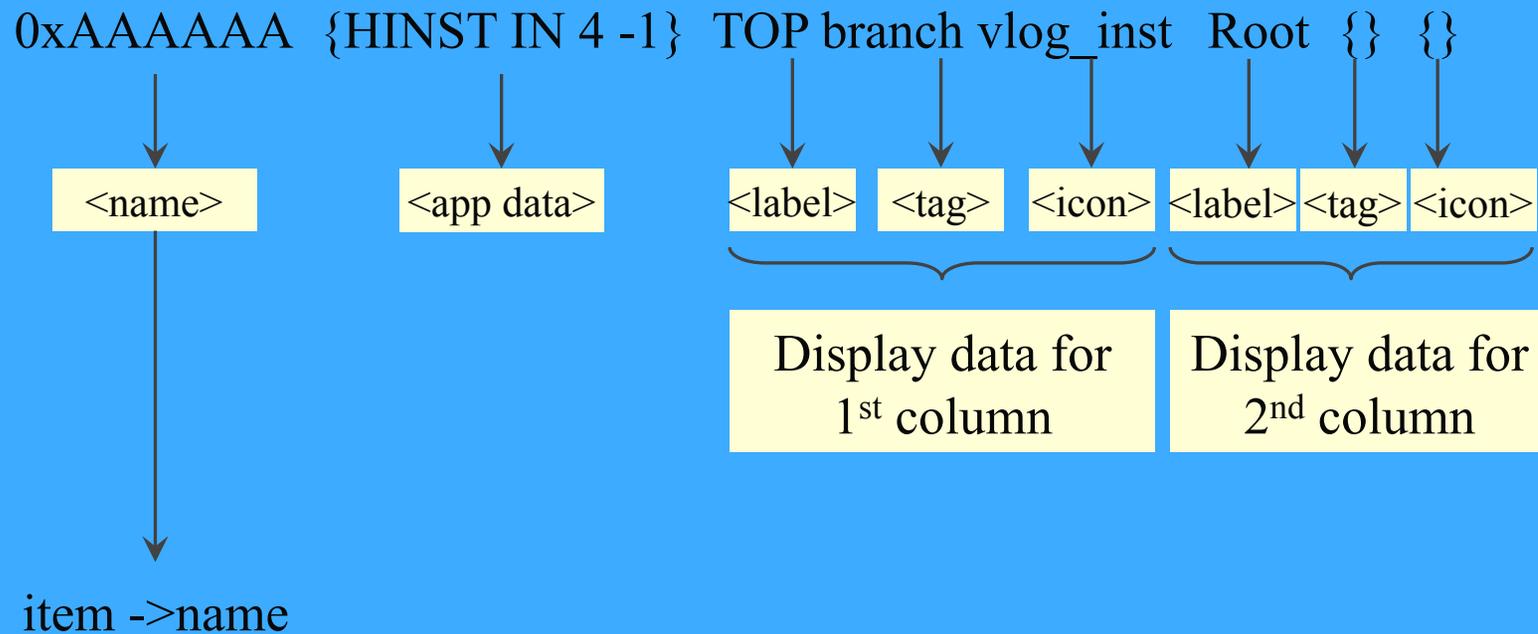
MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



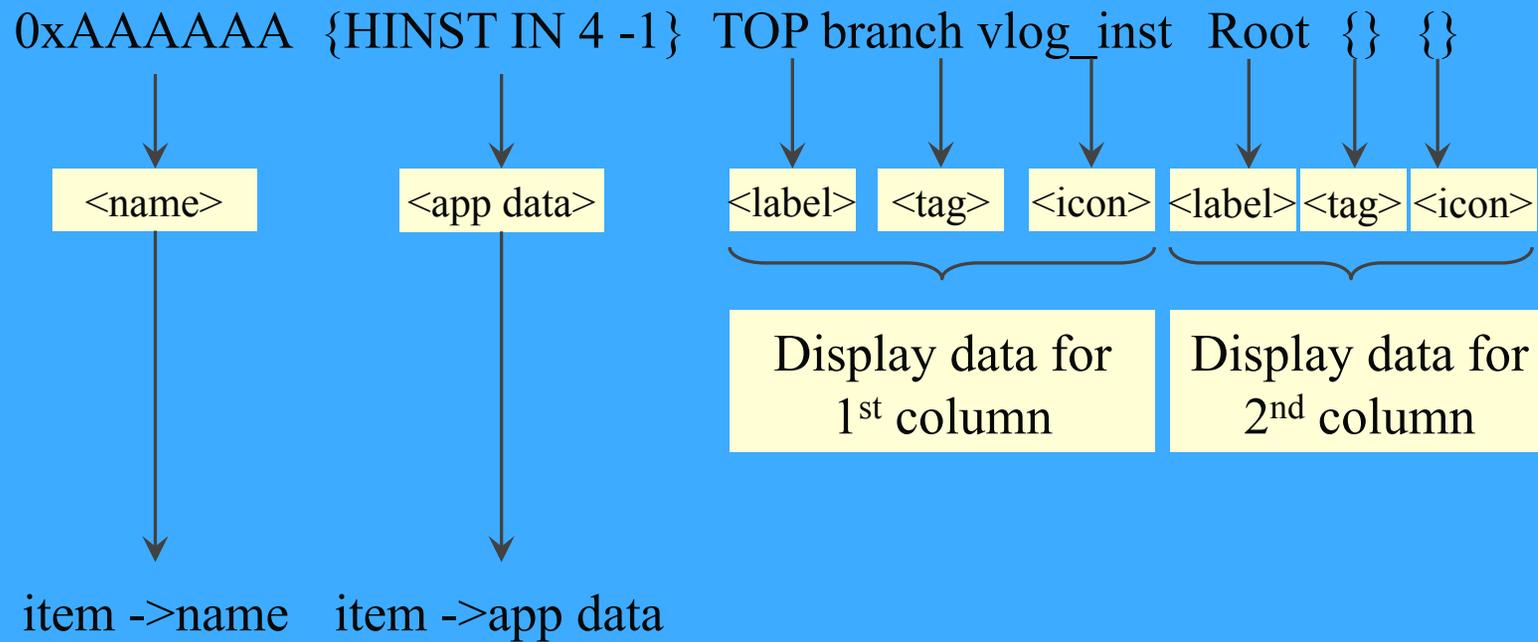
MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



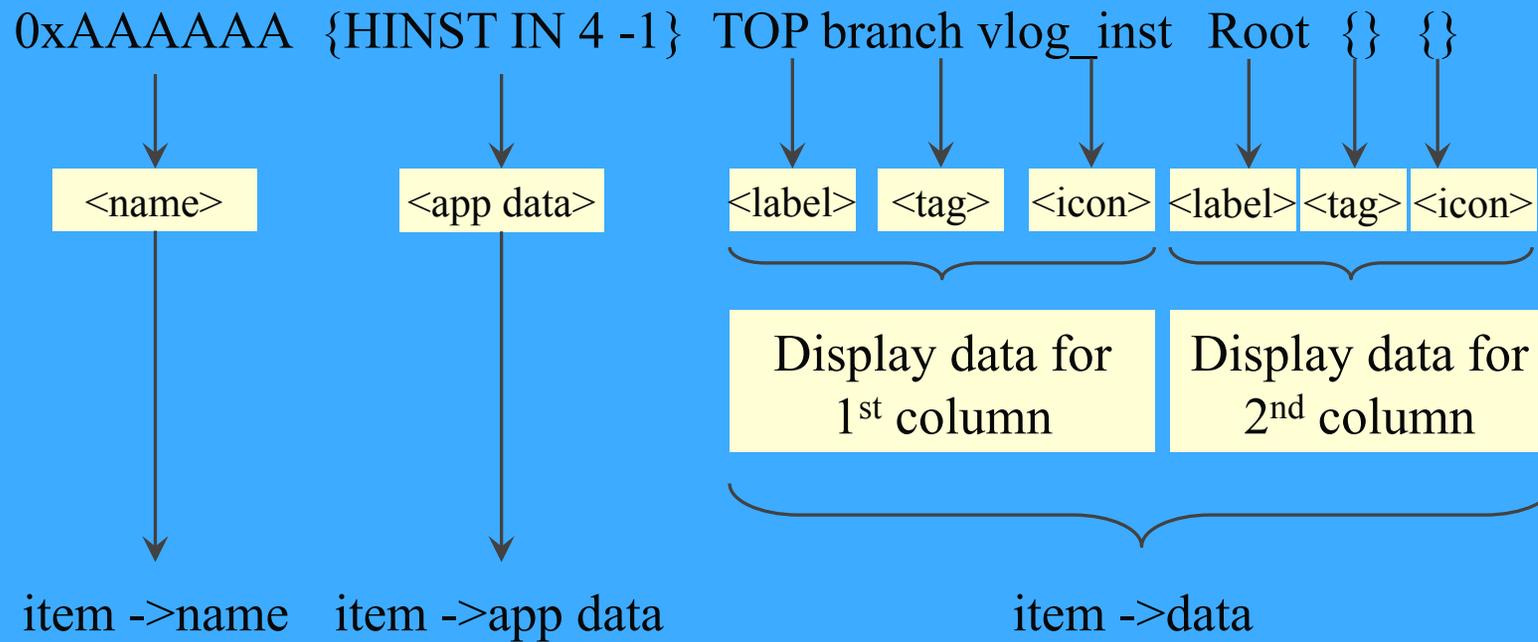
MtiHierarchy Widget – Callback Support

Instance Name	Design Unit	Instances
TOP	root	35



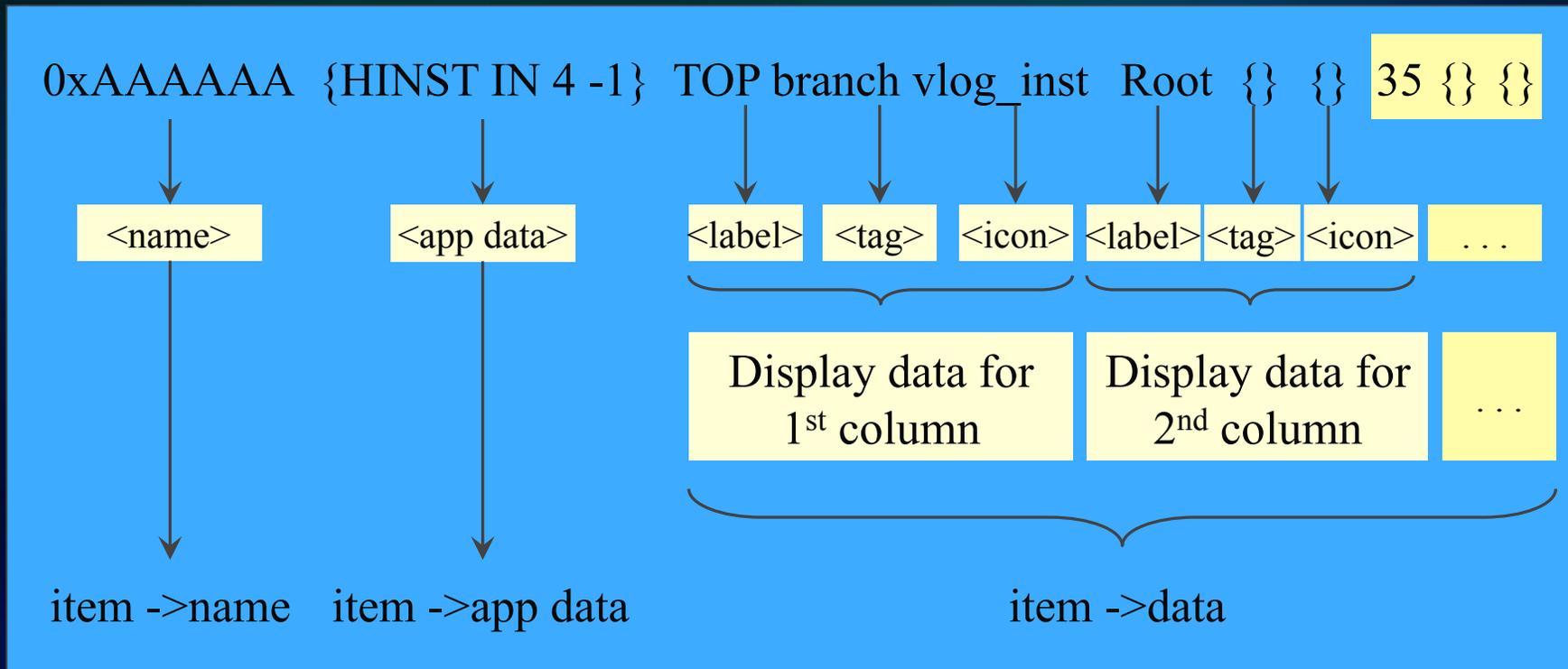
MtiHierarchy Widget – Callback Support

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

```
<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>...
```

Dynamic Data

MtiHierarchy Widget – Callback Support

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

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

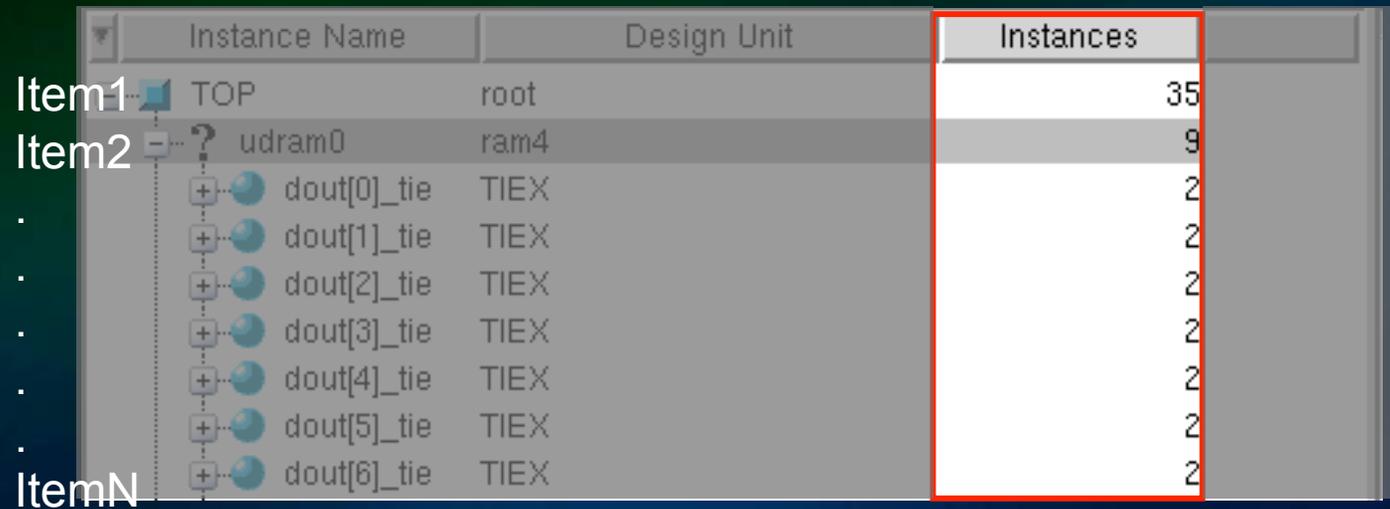
```
<name1> <app data1> <label11> <tag11> <icon11> <label12> <tag12>  
<icon12>...  
<name2> <app data2> <label21> <tag21> <icon21> <label22> <tag22>  
<icon22>...  
.  
.  
.  
<nameN> <app dataN> <labelN1> <tagN1> <iconN1> <labelN2> <tagN2>  
<iconN2>...
```

Widget gets from application at runtime

Dynamic Data

Side Effects

Side Effects – Sorting



The image shows a screenshot of a design tool's instance hierarchy table. The table has three columns: 'Instance Name', 'Design Unit', and 'Instances'. The 'Instances' column is highlighted with a red border. The table contains the following data:

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
+ 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)

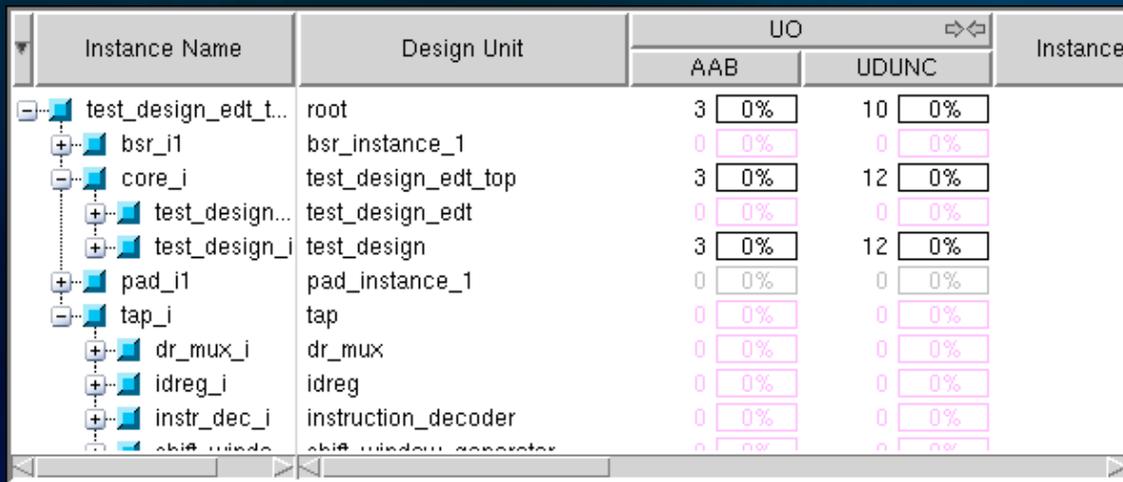
■ Enhancement in Text widget

- Hyperlinks

- Incremental Parsing

Sub-columns

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



The screenshot displays a software interface with a hierarchical tree on the left and a data table on the right. The tree shows a root instance 'test_design_edt_t...' with several sub-instances: 'bsr_i1', 'core_i', 'test_design...', 'test_design_i', 'pad_i1', 'tap_i', 'dr_mux_i', 'idreg_i', 'instr_dec_i', and 'shift_windo...'. The table below shows the fault distribution for each instance, categorized by Design Unit and UO (AAB and UDUNC).

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	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%

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	..._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	...generator	2	0	0%	0	0%

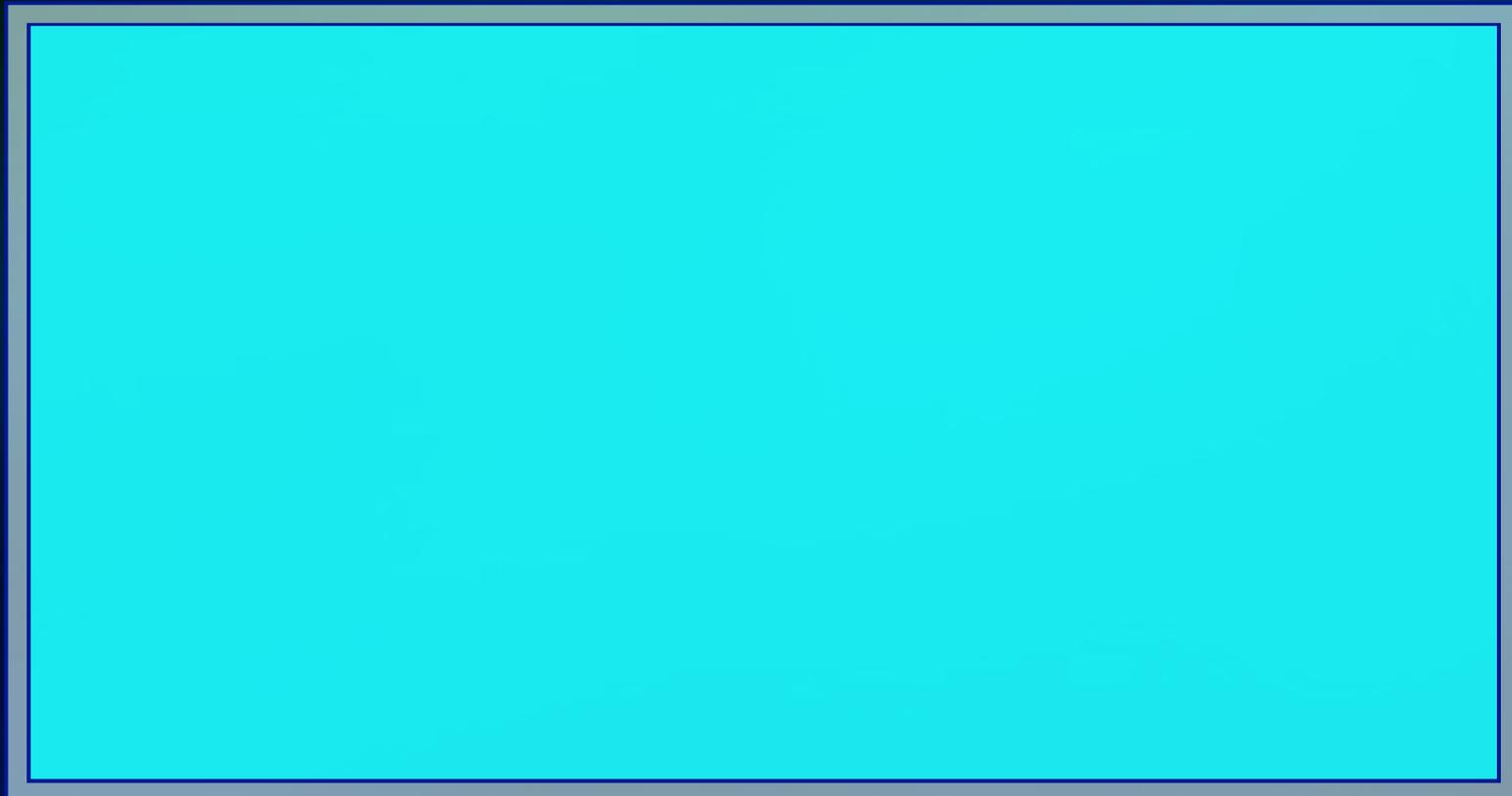
Frozen Tree Column

- Clipper (a frame)



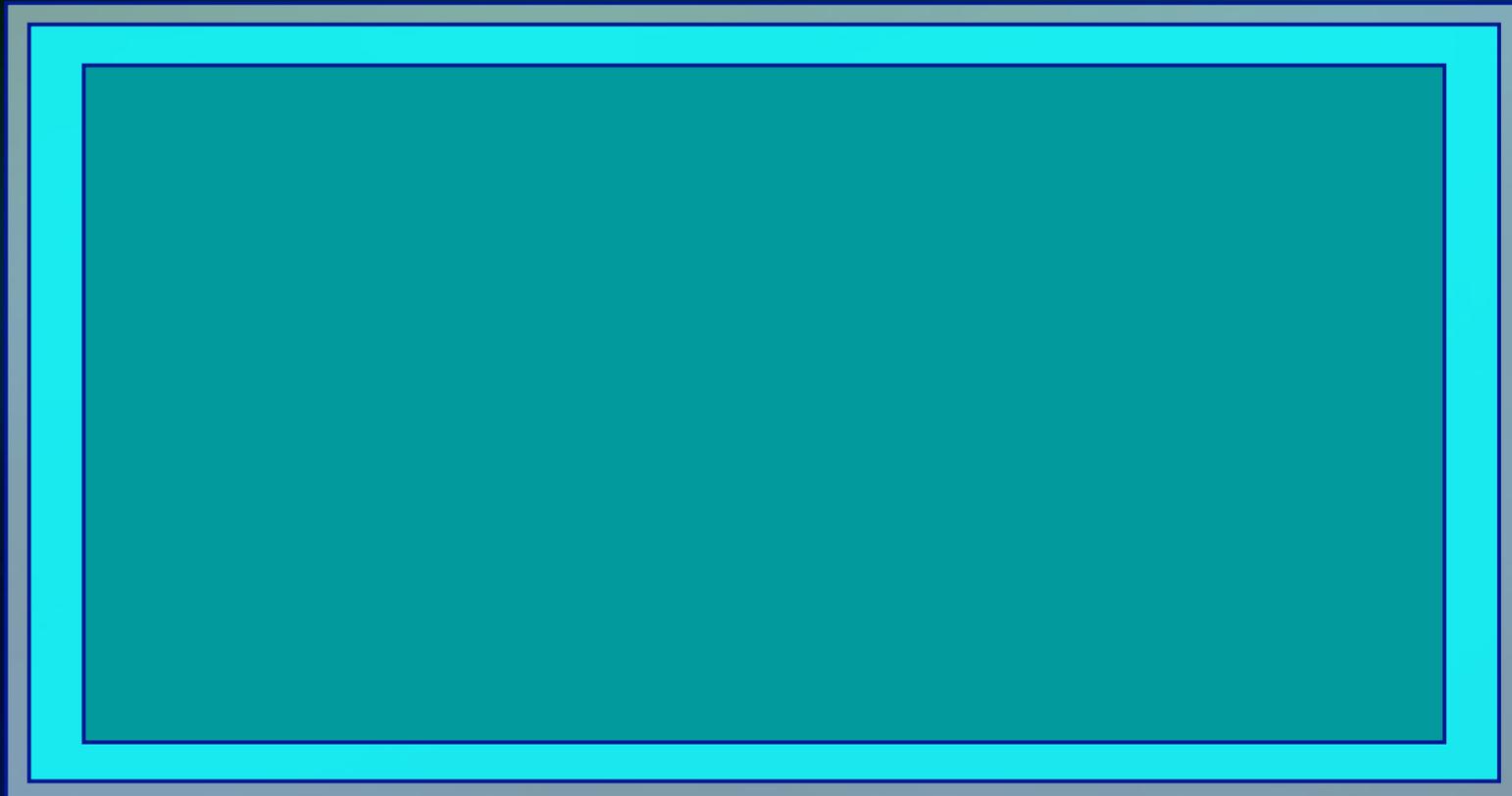
Frozen Tree Column

- Clipper (a frame) → Canvas



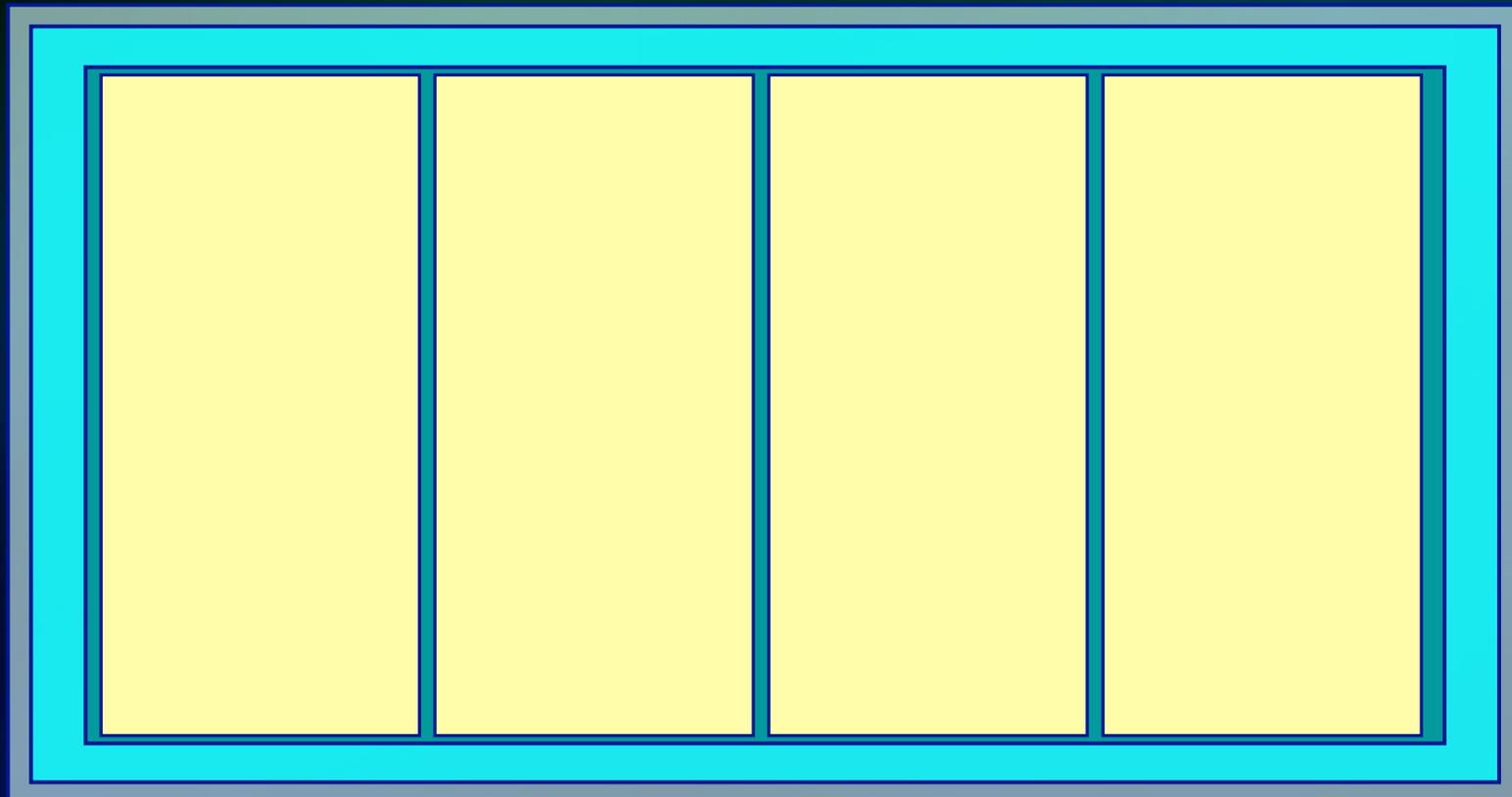
Frozen Tree Column

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



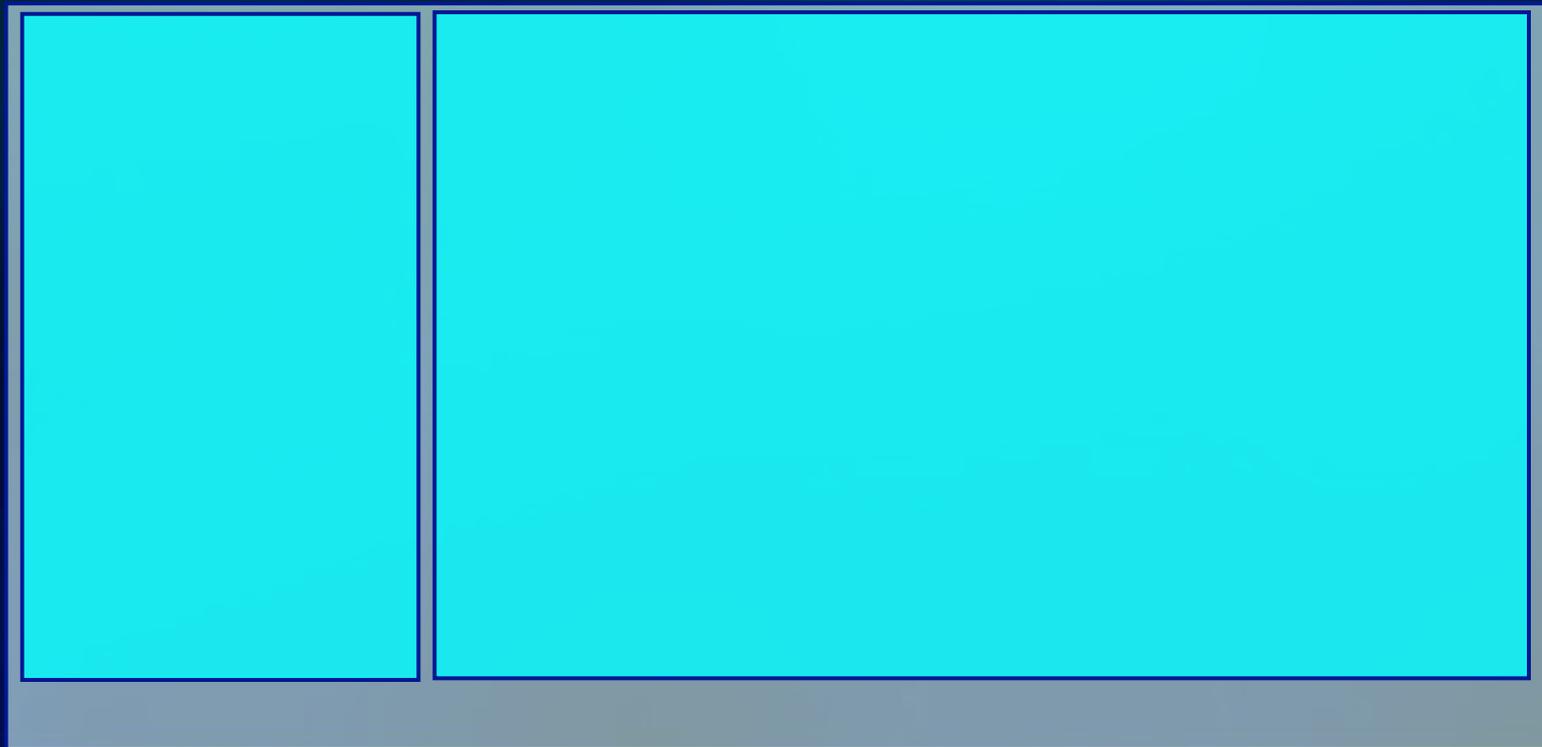
Frozen Tree Column

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



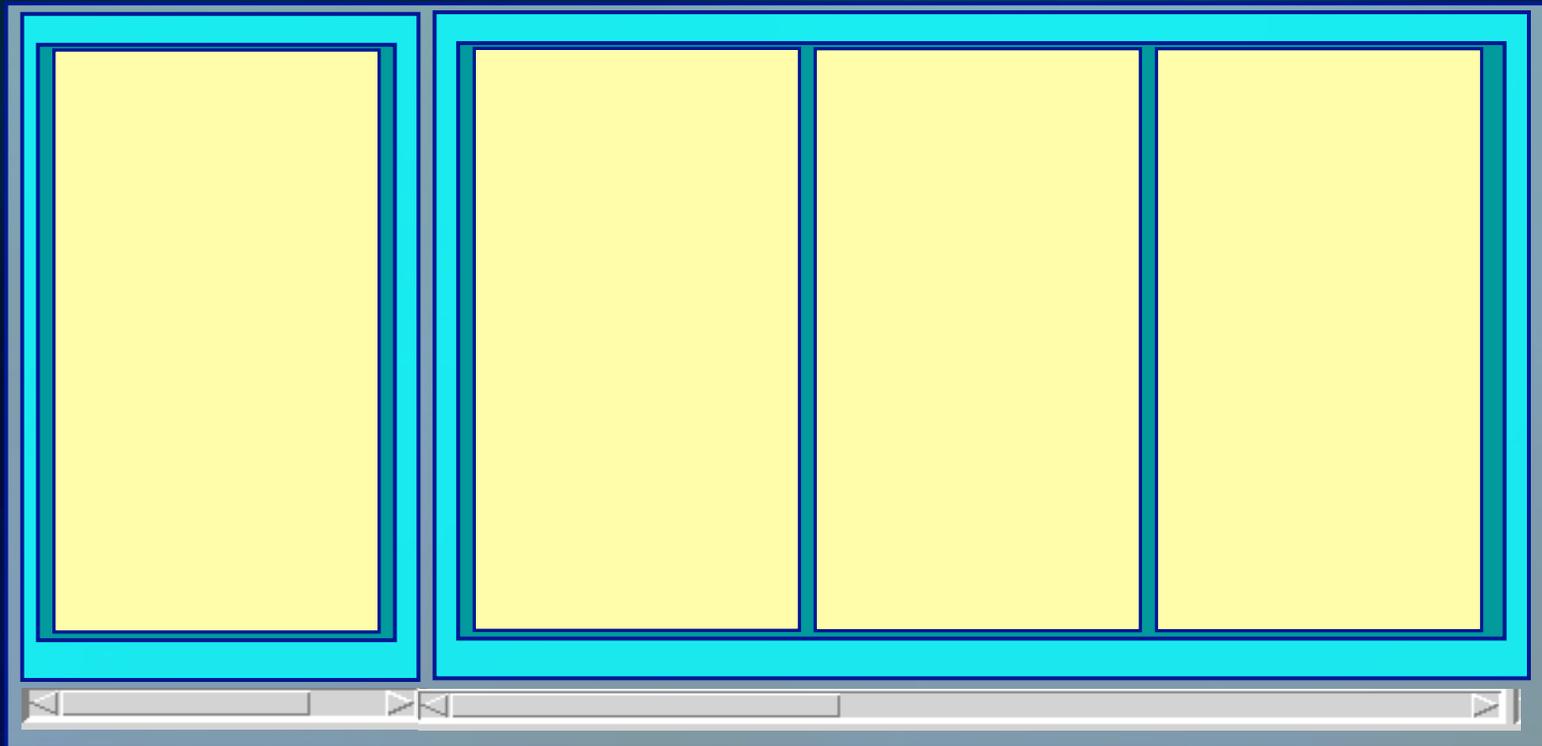
Frozen Tree Column

- 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

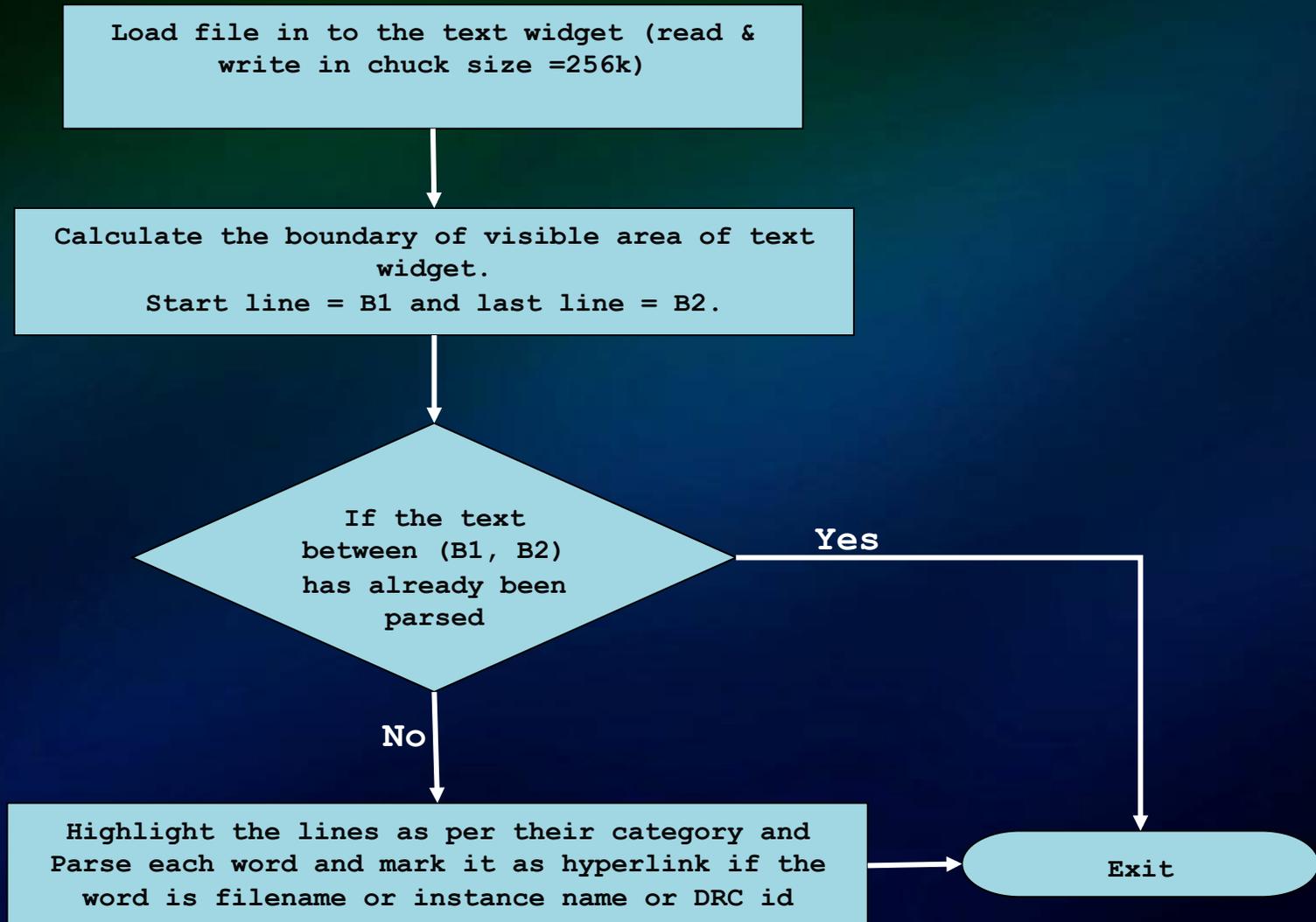
Enhanced Text Widget

- 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.