

## Freeze Column Example - TreeGrids

TreeGrid EXTJS 4.2 Big Data Example							
Region	Total	Male	Female	<5yrs	5-9	10-14	15-19
▼ Maine	1,274,923	620,309	654,614	70,726	83,022	92,252	89,485
▶ Androscoggin County	103,793	50,385	53,408	6,122	6,898	7,414	7,463
▶ Aroostook County	73,938	36,095	37,843	3,730	4,459	5,107	5,439
▼ Cumberland County	265,612	128,589	137,023	15,443	17,507	18,513	17,576
Baldwin town	1,290	632	658	77	91	86	82
Bridgton town	4,883	2,445	2,438	250	258	361	499
Brunswick town	21,172	10,210	10,962	1,313	1,394	1,400	1,819
Cape Elizabeth town	9,068	4,331	4,737	470	700	806	573
Casco town	3,469	1,714	1,755	206	238	275	224
Expand Collapse 2							

### a. Demonstrates

Horizontal scrolling for Tree Grids with synchronized vertical scrolling and expand/collapse functionality.  
Each column set references the same TreeStore, but with different column configurations  
Multiple freeze and horizontal scrolling column sets can be implemented

### b. Program Flow

Controller: CommonController.js

View: CensusFreezeCont.js

Data:

- TouchTreeGrid.store.CensusFreezeStore (storeId = censusFreezeStore)
- Model: TouchTreeGrid.model.Census
- TreeStore (./data/censusmaine2000TREE.json)
- Data loaded via launch() method within CommonController.js:

```
commonController.launch() =>
commonController.loadCensusFreezeStore =>
commonController.loadStore =>
commonController.postLoadProcess =>
commonController.loadColumnsCensusFreeze()
```

- censusController.loadColumnsCensusFreeze() loads columns[] array defined in pixels. Note pixels are preferred over em's in this case so that the column headers line up with the grid columns. This is necessary when font sizes are different between the two. CensusFreezeCont.js has initialize() method that reduces the column width of the initial freeze column for phone portrait orientations to allow more scrolling width for the remaining columns.
- **IMPORTANT:** When specifying column widths in pixels or em's for TreeGrids you **MUST** set following configs to avoid TPL compile error within doRefreshList() method:

**categIndentPct = '0'**  
**colNumberToTruncateForIndents = 0**

c. Notable Configs

- CensusFreezeCont.js is outermost container for all scrolling grids. Unlimited combinations of freeze columns and horizontal scrolling column sets supported.
  - **itemId = 'censusFreezeCont'**
  - **Layout = 'hbox'**
- **Freeze column**
  - xtype = 'touchtreegrid'
  - itemId = 'censusfreezeX'
  - listItemId = 'censusfreezelistX'
  - **linkedGridsParentItemId = 'censusFreezeCont'**  
(ultimate parent container itemId for all scrolling grids)
  - **linkedGridsArr = [{itemId: 'censusfreeze' }]**  
(itemId of other scrolling grid)
  - Columns[] array defines single dataIndex = 'CATEG' column (width = '200px').
  - **categIndentPct = '0' (REQUIRED FOR PIXEL/EM WIDTHS !!)**
  - **colNumberToTruncateForIndents = 0 (REQUIRED FOR PIXEL/EM WIDTHS !!)**
  - arrowPctWidth = '6' (need to choose percent that allows arrow to remain visible for narrower panel width)
  - Footer related configs
    - includeFooterLevels = true  
(we want to include expand/collapse and footer levels in freeze column only)
    - includeFooter = true (default and not required)
  - minWidth/width = '220px' (includes 20px right padding)
  - style = 'border-right: 2px solid gray;' (line border between panels)
  - **Phone-specific configuration**
  - CensusFreezeCont.js initialize() method contains logic to reduce width of freeze column for Phone Portrait configurations to allow more scrolling width for scrolling columns
- **Horizontal Scrolling columns**
  - Horizontal scrolling columns require a parent container with following configs:
    - i. Flex = 1 (optional, but at least one set of scrolling grids should have flex=1)
    - ii. Layout = 'hbox'
    - iii. Scrollable = { direction: 'horizontal', directionLock: true }
  - Child container to this parent container is linked instance of TouchTreeGrid
  - xtype = 'touchtreegrid'
  - itemId = 'censusfreeze'
  - listItemId = 'censusfreezelist'
  - **flex = 1** (required here)

- **linkedGridsParentItemId** = 'censusFreezeCont'  
(ultimate parent container itemId for all scrolling grids)
- **linkedGridsArr** = [{itemId: 'censusfreezeX' }]  
(itemId of other scrolling grid)
- Columns[] array defines all scrolling columns (except frozen CATEG column and is loaded within loadColumnsCensusFreeze() method of CensusController.js. All widths defined in pixels (vs. em's) so that detail columns line up with header columns (which have different font-size).
- categIndentPct = '0' **(REQUIRED FOR PIXEL/EM WIDTHS !!)**
- colNumberToTruncateForIndents = 0 **(REQUIRED FOR PIXEL/EM WIDTHS !!)**
- arrowPctWidth = '6' (need to choose percent that allows arrow to remain visible for narrower panel width)
- Footer related configs
  - includeFooter = true (default)
  - hideExpandCollapseBtns = true  
(we want footer bar for consistency across both scrolling grids, but we don't want expand/collapse and collapse-to button repeated on second panel)
  - includeFooterLevels = false
- minWidth/width = '220px' (includes 20px right padding)
- style = 'border-right: 2px solid gray;' (line border between panels)

b. CSS – overrides default CSS defined in TouchTreeGrid.css via '.x-touchtreegrid-list-censusmaine' selector. Refer to treegriddemo.css:

- .x-touchtreegrid-list-censusmaine .touchtreegrid-list-categ0 {...}
- .x-touchtreegrid-list-censusmaine .touchtreegrid-list-categ1 {...}
- .x-touchtreegrid-list-censusmaine .touchtreegrid-list-content{...}
- .x-touchtreegrid-list-censusmaine .touchtreegrid-header {...}

## Freeze Column Example – Simple List with Grouped Headers

(Simulates EXTJS 4.2 KitchenSink Big Data grid example)

TreeGrid EXTJS 4.2 Big Data Example							
Id	Name	Notice Period	Email address	Absences			Rate
				Illness	Holidays	Holiday Allowance	
Accounting							
513524	Abe Avins	2 weeks	<a href="mailto:abe_avins@sentcha.com">abe_avins@sentcha.com</a>	4	7	27	2
254015	Dave Avins	1 month	<a href="mailto:dave_avins@sentcha.com">dave_avins@sentcha.com</a>	3	5	28	3
520444	Jamie Avins	3 months	<a href="mailto:jamie_avins@sentcha.com">jamie_avins@sentcha.com</a>	0	7	26	5
041425	Jamie Avins	3 months	<a href="mailto:jamie_avins@sentcha.com">jamie_avins@sentcha.com</a>	9	4	30	4
365413	Abe Conran	2 weeks	<a href="mailto:abe_conran@sentcha.com">abe_conran@sentcha.com</a>	2	1	36	4
431730	Abe Conran	2 weeks	<a href="mailto:abe_conran@sentcha.com">abe_conran@sentcha.com</a>	0	7	34	1
632526	Nicolas Conran	2 weeks	<a href="mailto:nicolas_conran@sentcha.com">nicolas_conran@sentcha.com</a>	7	4	32	5
107471	Tommy Conran	1 month	<a href="mailto:tommy_conran@sentcha.com">tommy_conran@sentcha.com</a>	7	2	36	1
702373	Tommy Conran	2 weeks	<a href="mailto:tommy_conran@sentcha.com">tommy_conran@sentcha.com</a>	9	3	22	3
455535	Tommy Conran	3 months	<a href="mailto:tommy_conran@sentcha.com">tommy_conran@sentcha.com</a>	1	4	29	4
203626	Abe Davis	2 weeks	<a href="mailto:abe_davis@sentcha.com">abe_davis@sentcha.com</a>	0	5	26	5
344354	Abe Davis	2 weeks	<a href="mailto:abe_davis@sentcha.com">abe_davis@sentcha.com</a>	8	9	32	1
776720	Adam Davis	3 months	<a href="mailto:adam_davis@sentcha.com">adam_davis@sentcha.com</a>	3	1	24	4
771406	David Davis	3 months	<a href="mailto:david_davis@sentcha.com">david_davis@sentcha.com</a>	10	10	33	5
361321	Jamie Davis	3 months	<a href="mailto:jamie_davis@sentcha.com">jamie_davis@sentcha.com</a>	8	3	38	2
276453	Jay Davis	1 month	<a href="mailto:jay_davis@sentcha.com">jay_davis@sentcha.com</a>	2	8	24	5
144031	Nicolas Davis	3 months	<a href="mailto:nicolas_davis@sentcha.com">nicolas_davis@sentcha.com</a>	8	7	35	5

### a. Demonstrates

Horizontal scrolling for Simple Lists with grouped Grids with synchronized vertical scrolling and expand/collapse functionality.

This particular example simulates EXTJS 4.2 Kitchen Sink Big Data grid example.

headerTplOverride config utilized to create custom grouped column headers.

Each column set references the same Store, but with different column configurations

Multiple freeze and horizontal scrolling column sets can be implemented

### b. Program Flow

View: bigDataList.js

Data:

- TouchTreeGrid.store.BigDataStore (storeId = BigDataStore)
- Model: TouchTreeGrid.model.BigDataModel
- Store (./data/BigData.json) – dummied data provided in Sencha EXTJS 4.2.1 download (\\ext-4.2.1.883\\examples\\kitchensink\\app\\data\\BigData.js)
- Data loaded automatically via Store.

### c. Notable Configs

- bigDataList.js is outermost container for all scrolling grids. Unlimited combinations of freeze columns and horizontal scrolling column sets supported.
  - itemId = 'bigDataList'
  - Layout = 'hbox'
- Freeze column
  - xtype = 'touchtreegrid'
  - itemId = 'bigDataX'
  - listItemId = 'bigDataListX'

- **linkedGridsParentItemId** = 'bigDataList'  
(ultimate parent container itemId for all scrolling grids)
  - **linkedGridsArr** = [{itemId: 'bigDataList' }]  
(itemId of other scrolling grid)
  - **Columns[]** array:
 

```
columns: [
  {
    header: 'Id',
    dataIndex: 'employeeNo',
    width: '70px',
    style: 'text-align: center;',
    sortable: true
  },
  {
    header: 'Name',
    dataIndex: 'surname',
    width: '150px',
    style: 'text-align: left; padding-left: 8px;',
    renderer: 'this.fullName(values.forename, values.surname)',
    sortable: true
  }
]
```
  - header: {maxHeight: '59px', minHeight: '59px'}
  - itemHeight: 24
  - variableHeights: false
  - infinite: true
  - simpleList: true
  - columnSorting: true
  - additionalListConfigs: {grouped: true}
  - renderers: {fullName: function(forename, surname)
 {return forename + ' ' + surname;}}
  - headerTplOverride: (refer below)
  - minWidth/width = '220px' (includes 20px right padding)
  - style = 'border-right: 2px solid gray;' (line border between panels)
- **Horizontal Scrolling columns**  
Horizontal scrolling columns require a parent container with following configs:
    - Flex = 1 (optional, but at least one set of scrolling grids should have flex=1)
    - Layout = 'hbox'
    - Scrollable = {direction: 'horizontal', directionLock: true}

Child container to this parent container is linked instance of TouchTreeGrid

- xtype = 'touchtreegrid'
- itemId = 'bigData'
- listItemId = 'bigDataList'
- **flex = 1** (required here)
- **linkedGridsParentItemId** = 'bigDataList'  
(ultimate parent container itemId for all scrolling grids)
- **linkedGridsArr** = [{itemId: 'bigDataX' }]  
(itemId of other scrolling grid)
- **Columns[]** array
 

```
[
  {
    header: 'Date of Birth',
    dataIndex: 'dob',
    width: '120px',
```

```

        style: 'text-align: left;',
        renderer: 'Ext.Date.format(values.dob, "m/d/Y")',
        sortable: true
    },
    {
        header: 'Join Date',
        dataIndex: 'joinDate',
        width: '120px',
        style: 'text-align: left;',
        renderer: 'Ext.Date.format(values.joinDate, "m/d/Y")',
        sortable: true
    },
    {
        header: 'Notice Period',
        dataIndex: 'noticePeriod',
        width: '120px',
        style: 'text-align: left;',
        sortable: true
    },
    {
        header: 'Email address',
        dataIndex: 'email',
        width: '200px',
        style: 'text-align: left;',
        renderer: 'this.emailFormat(values.email)',
        sortable: true
    },
    {
        header: 'Illness',
        dataIndex: 'sickDays',
        width: '100px',
        style: 'text-align: left;',
        sortable: false
    },
    {
        header: 'Holidays',
        dataIndex: 'holidayDays',
        width: '100px',
        style: 'text-align: left;',
        sortable: false
    },
    {
        header: 'Holiday Allowance',
        dataIndex: 'holidayAllowance',
        width: '150px',
        style: 'text-align: left;',
        sortable: false
    },
    {
        header: 'Rating',
        dataIndex: 'rating',
        width: '70px',
        style: 'text-align: left;',
        sortable: true
    },
    {
        header: 'Salary',
        dataIndex: 'salary',
        width: '110px',
        style: 'text-align: right; padding-right: 8px;',
        sortable: true,
        renderer: "this.formatNumbers(values.salary, 0, '$")"
    }
]

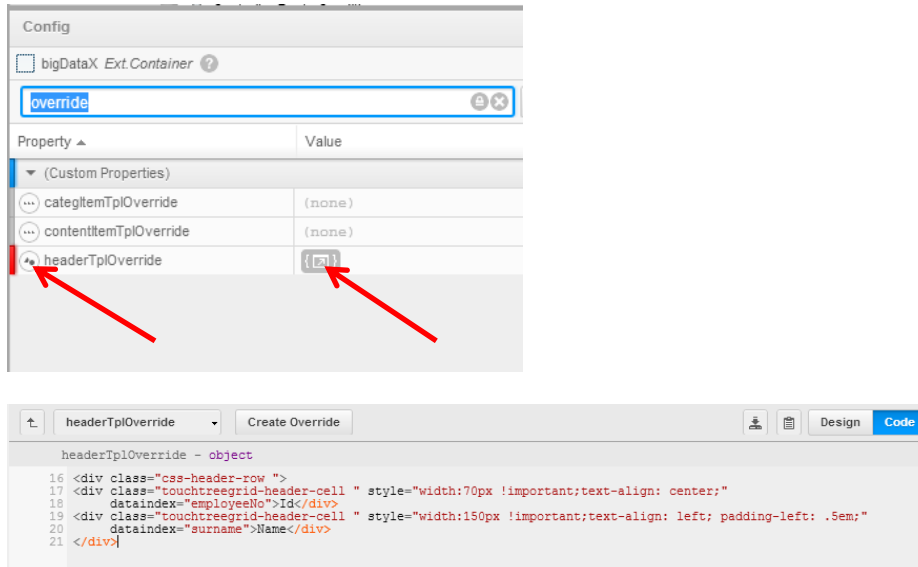
```

- **renderers:** {emailFormat: function(email)  
 {return '<a href="mailto:' + email + '">' + email + '</a>';}}
- minWidth/width = '1200px' (includes right padding)
- style = 'border-right: 2px solid gray;' (line border between panels)
- other configs same as itemId = 'bigData'

### c. headerTplOverride Config

This config used to customize column headers including grouped headers.

**TIP:** Architect allows long formatted text to be input into text string configs by changing type to “Object” for easier maintenance. Architect-generated code will be a single line of code however.



Header TPL for both Freeze and Scrolling columns:

```
/* Freeze Columns (itemId = 'bigDataX') */
<div class="css-header-row">
<div class="touchtreegrid-header-cell" style="width:70px !important;text-align: center;" dataIndex="employeeNo">Id</div>
<div class="touchtreegrid-header-cell" style="width:150px !important;text-align: left; padding-left: .5em;"
dataindex="surname">Name</div>
</div>

/* Horiz Scrolling Columns (itemId = 'bigData'):
- Column header grouping implemented using HTML table
- Included dataIndex attribute for column sorting (also requires columnSorting=true config in linked instance of
TouchTreeGrid.
(column sorting not currently supported for sub-grouped headers so dataIndex attribute not included)
*/
<div class="css-header-row">
<div class="touchtreegrid-header-cell" style="width:120px !important;text-align: left;" dataIndex="dob">Date of Birth</div>
<div class="touchtreegrid-header-cell" style="width:120px !important;text-align: left;" dataIndex="joinDate">Join Date</div>
<div class="touchtreegrid-header-cell" style="width:120px !important;text-align: left;" dataIndex="noticePeriod">Notice
Period</div>
<div class="touchtreegrid-header-cell" style="width:200px !important;text-align: left;" dataIndex="email">Email address</div>

<table cellpadding="0" cellspacing="0">
<tr class="touchtreegrid-header-cell" style="line-height: 29px; border-bottom: 1px solid #c0c0c0;
border-right: 1px solid #c0c0c0; width:350px !important;text-align: center;"><td colspan="3">Absences</td></tr>
<tr>
<td class="touchtreegrid-header-cell" style="line-height: 30px; width:100px !important;text-align: left;">Illness</td>
<td class="touchtreegrid-header-cell" style="line-height: 30px; width:100px !important;text-align: left;">Holidays</td>
<td class="touchtreegrid-header-cell" style="line-height: 30px; width:150px !important;text-align: left;">Holiday Allowance</td>
</tr>
</table>

<div class="touchtreegrid-header-cell" style="width:70px !important;text-align: left;" dataIndex="rating">Rating</div>
<div class="touchtreegrid-header-cell" style="width:110px !important;text-align: right;padding-right: 8px;"
dataindex="salary">Salary</div>
</div>
```

### d. CSS – overrides default CSS defined in TouchTreeGrid.css via '.x-touchtreegrid-list-bigData' selector. Refer to treegriddemo.css:

- .x-touchtreegrid-list-bigData .touchtreegrid-header {...}
- .x-touchtreegrid-list-bigData .touchtreegrid-header-cell {...}

- .x-touchtreegrid-list-bigData .touchtreegrid-simplelist-cell{...}
- .x-touchtreegrid-list-bigData .x-touchtreegrid-item {...}
- .x-touchtreegrid-list-bigData .x-touchtreegrid-item:nth-child(even) {...}  
/\* for alternate shading \*/
- .x-touchtreegrid-list-bigData .x-grid-sort-desc {...}
- .x-touchtreegrid-list-bigData .x-grid-sort-asc {...}
- .x-touchtreegrid-list-bigData .x-list-header {...} /\* Grouped header rows \*/