

WOLL2WOLL SOFTWARE

Release Notes

FirePower X

Supporting RAD Studio 10.4 Sydney and 10.3 Rio

© Woll2Woll Software Inc.

357 Yukon Way

Livermore, CA 94550

Email: sales@woll2woll.net

<http://www.woll2woll.com>

Table of Contents

WHATS NEW IN FIREPOWER X	1
New TwwTreeView Component	1
Setting up the TwwTreeView to edit data	1
Adding Drag/Drop	2
Assign images to the caption column	2
Enabling multi-selection checkboxes	2
Configuring the type of control to edit with	2
Auto-sizing of Caption and Edit columns	2
Common Methods	2
Adding tree items with code.	2
Combo Enhancements	3
Customizing the dropdown icon	3
Enabling Multi-Selection Checkboxes	3
Select predefined listbox layouts for the dropdown list	3
TwwLayoutGrid modernization	3
TwwDataGrid now supports background images	4
See Revision History for changes in FirePower	4

WHATS NEW IN FIREPOWER X

FirePower X brings a host of new capabilities to existing components, integration with new functionality in RAD Studio 10.4, and a new featured component in the TwwTreeView

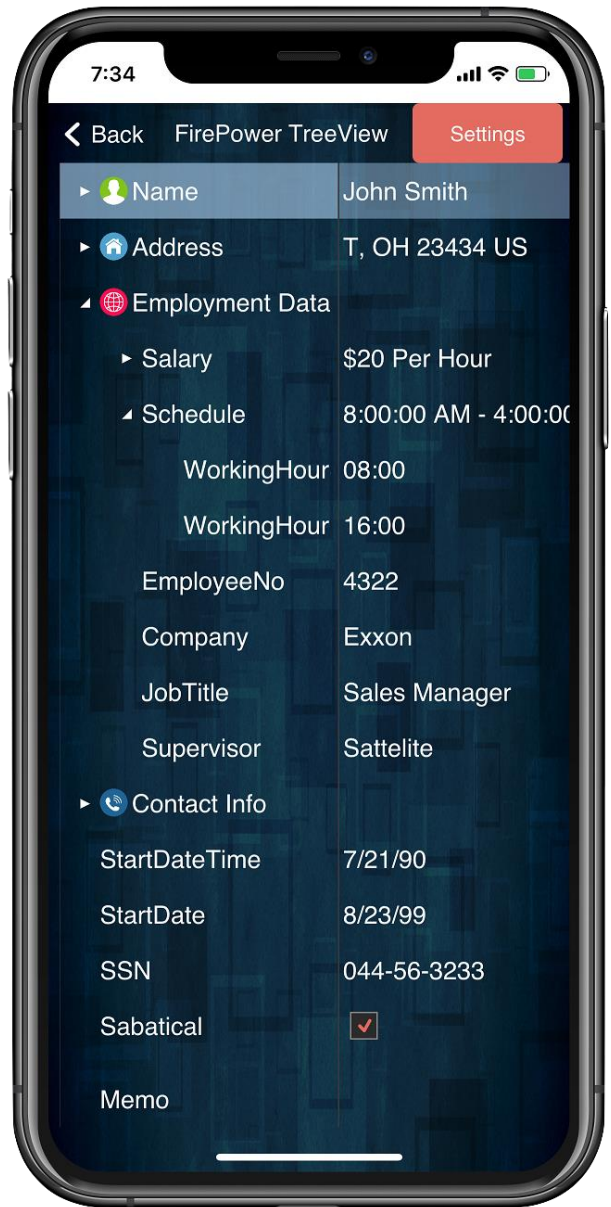
New TwwTreeView Component

Use the flexible new TwwTreeView component to display or edit information in a hierarchical display. The FirePower Tree is built from the ground up to be fast. Load 10000 nodes in a fraction of a second, even on mobile devices. Your nodes can be variable height. Supports checkboxes, images, and automatically uses the 'TreeView' style in your stylebook. You can also use this control to edit hierarchical data (such as the IDE object inspector) with the ability to embed different FirePower controls into each row.

Setting up the TwwTreeView to edit data

To setup a TwwTreeView for editing data from a table, set the following properties

1. *ColumnEdit.Visible*. If you do not wish to show the editing column, set *ColumnEdit.Visible* to false.
2. Assign *DataSource* to the binding source that contains your data (such as a *TBindingSourceDB*). If you don't already have one on your form, you can right-click your dataset and select *Add BindSource*
3. Then to assign your treeitems to edit your data, dbl-click the component from the IDE to bring up the items editor where you can add your fields.
4. See the demo form *TreeViewWalkthru* that is part of our maindemo to see a complete example of using the tree to edit a record of your dataset.
5. Optional: Please note that this demo uses the *CustomFormat* property of the *TwwTreeViewItem* to automatically concatenate the strings of multiple fields to display in the tree when grouping related items together. The beauty of this is that you can see your data at design time, and do not need to create calculated fields in your dataset to compute these



values. You should also set the `ReadOnly` to true so that focus does not go into this row/item. For an example, see the items in the tree captioned 'Name' and 'Address' from our demo mentioned above. Please note that you do not need to do this step if you are content with not seeing a summary of the child items data.

Adding Drag/Drop

If you wish to allow drag/drop between items on desktop platforms, then set `AllowDrag` to true

Assign images to the caption column

To assign images to the caption part of the tree, assign your `images` property in your `TnwTreeView` followed by assigning the `ImageName` property of your `TnwTreeViewItem`.

Enabling multi-selection checkboxes

To enable checkboxes in the caption column, set `ShowCheckboxes` to true. This will enable checkboxes for all columns. To individually configure checkboxes for each item, assign the `TnwTreeViewItem.CheckboxType` property. To check with your code if a item is checked, then use the `IsCaptionChecked` property of the `TnwTreeViewItem`.

Configuring the type of control to edit with

To configure the control type for a tree item, use the `ControlType` and `ControlAttributes` properties of the `TnwTreeViewItem`

Auto-sizing of Caption and Edit columns

Use `AutoSizeColumns` to configure the tree to always fully utilize available space and adjust the editing column's width based on the width of the tree.

Common Methods

To expand or collapse the entire tree use the `TnwTreeView's ExpandAll` and `CollapseAll` methods.

Adding tree items with code.

To add items with code use the following syntax. This will add 100 nodes with 5 children in each. Make sure that you enclose your tree changes within `BeginUpdate/EndUpdate` or you may encounter unexpected painting issues.

```
var item, childitem: TnwTreeViewItem;  
  i,j: integer;  
begin  
  wwTreeView1.BeginUpdate;  
  for i:= 1 to 100 do  
    begin
```

```

item:= wwTreeView1.Items.Add;
item.Text:= inttostr(i);
item.EditText:= inttostr(i);
for j:= 1 to 5 do begin
    childitem:= item.Items.Add;
    childitem.Text:= inttostr(j);
    childitem.EditText:= inttostr(j);
end
end;
wwTreeView1.EndUpdate;

```

Combo Enhancements

New enhancements in this version support customization of the drop-down icons and support for multi-selections from the dropdown list using the `TwwAdvComboEdit`

Customizing the dropdown icon

Use the `ButtonImages` and `ButtonImageName` property to override the icon used for the dropdown button.

Enabling Multi-Selection Checkboxes

Set `ShowCheckbox` to true. This will enable checkboxes in the dropdown list. To access which items are selected, you can refer to the `Item's Selected` property. For instance this code shows each selection that is checked

```

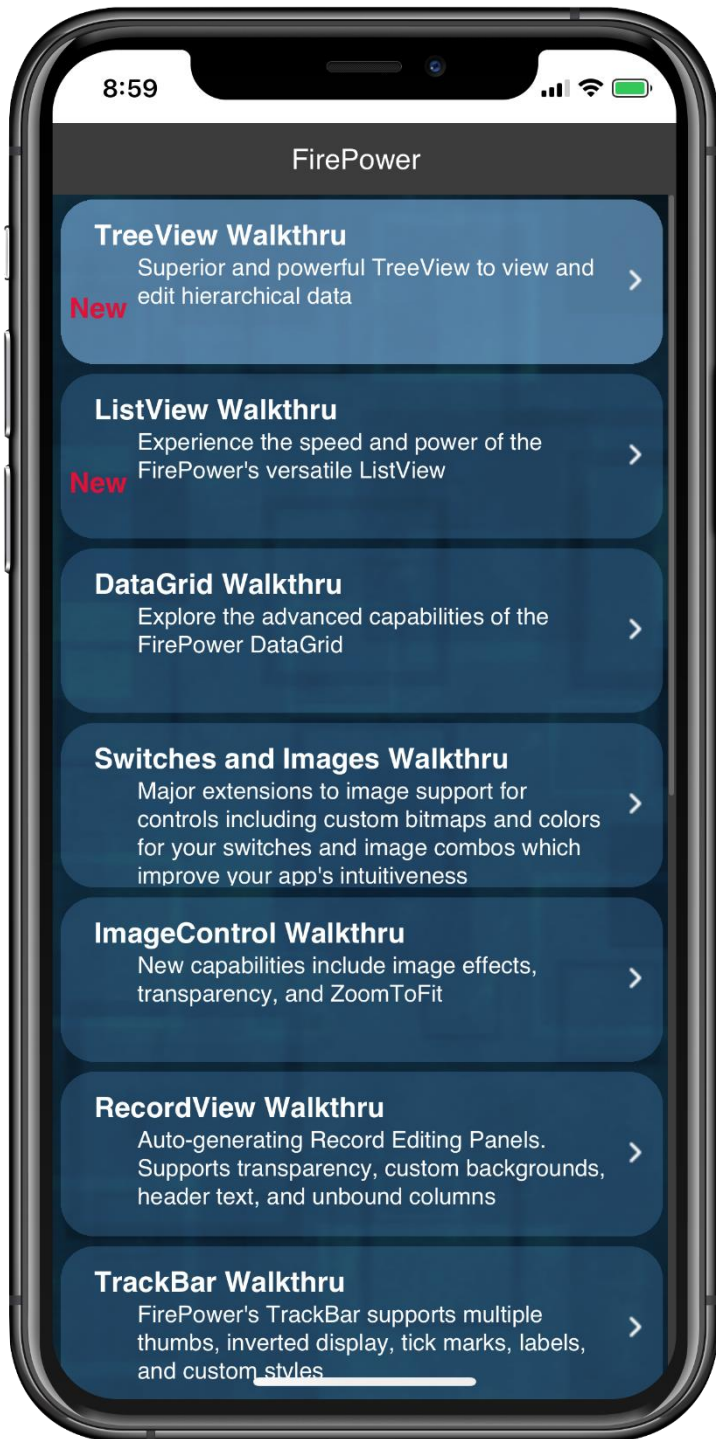
for i:= 0 to wwadvcomboedit1.Items.Count-1 do
begin
    if wwadvcomboedit1.Items[i].selected then
        showmessage(wwadvcomboedit1.Items[i].StoredText);
end;

```

Select predefined listbox layouts for the dropdown list

Combos dropdown list can have multiple layouts using the new property `ListboxLayoutName`. You can also select this at design time by right-clicking the component and selecting *Select Item Layout*

TwwLayoutGrid modernization



Support rounded rectangles and spacing for items using the new *OverrideStyleSettings.Frame* property. Assign *OverrideStyleSettings.FFrame* to change the frame properties a record is presented by in the *LayoutGrid*. Normally this defaults to a simple rectangle, or just lines separating the records. By assigning these properties, you can change to a rounded rectangle and also change the colors. To enable custom framing, you minimally must set the *Frame.Enabled* to true and the *Color* property to a non-null value. You may also want to configure the frame's *ColorOpacityPercent* and the *SelectionColor* and *SelectionColorOpacityPercent* properties. Note that when framing is enabled, the lines are no longer drawn. You can review the *mainunit.pas* file in the *maindemo* application to see an example of framing enabled

TwwDataGrid now supports background images

Use the new *ImageControl* property to select a background image for the grid. You can use the *ImageControl*'s properties to have ultimate flexibility on the background image including brightness, drawing style, stretching, tiling and zooming.

See Revision History for changes in FirePower

<https://www.woll2woll.com/firepower-revisions>