



# GWT Composite Builder

Issue Number: 1.0  
Issue Date: May 25, 2009  
Author: Jim Campbell

# GWT Composite Builder

## Update History

Authors	Date	Revision	Changes
Jim Campbell	May 25, 2009	1.0	Initial Release

# GWT Composite Builder

## Table of Contents:

<b>Update History</b> .....	<b>2</b>
<b>Overview</b> .....	<b>6</b>
<b>Figure 1</b> .....	<b>6</b>
<b>Acknowledgements</b> .....	<b>7</b>
<b>Tool Structure</b> .....	<b>8</b>
<b>A.1 Heading</b> .....	<b>8</b>
<b>A.2 Menu</b> .....	<b>8</b>
A.2.1 File .....	8
A.2.2 Edit .....	8
A.2.3 View .....	8
A.2.4 Tools .....	8
A.2.5 Help .....	8
<b>B - GWT Widgets</b> .....	<b>9</b>
<b>C - LabWidget</b> .....	<b>9</b>
<b>Splitter</b> .....	<b>9</b>
<b>D - UI Tree</b> .....	<b>10</b>
<b>E - Property Editor</b> .....	<b>10</b>
<b>F - CSS</b> .....	<b>10</b>
<b>COMPOSITE</b> .....	<b>11</b>
<b>Comments / Feedback</b> .....	<b>11</b>
<b>Getting Started</b> .....	<b>12</b>
<b>Figure 2</b> .....	<b>12</b>
<b>1. Decide on the GWT widget to use as the wrapper for the Composite</b> .....	<b>12</b>
<b>2. Design the layout and structure of the contained widgets</b> .....	<b>12</b>
<b>3. Build steps</b> .....	<b>13</b>
3.1 Create LabWidgets .....	13
3.2 Modify LabWidgets Properties .....	14
<b>4. CSS steps</b> .....	<b>16</b>
<b>5. Export / Import XML</b> .....	<b>17</b>
<b>6. Export Java</b> .....	<b>17</b>
<b>7. Implement any interfaces and business logic, as required</b> .....	<b>17</b>
<b>8. Test</b> .....	<b>17</b>
<b>Working with the GWT Composite Builder</b> .....	<b>18</b>
<b>GWT Widget Detail</b> .....	<b>21</b>
AbsolutePanel .....	21
Anchor .....	21
Button .....	21
CaptionPanel .....	21
CheckBox .....	21
Composite .....	21
DateBox .....	21
DeckPanel .....	21
DecoratedStackPanel .....	23
DecoratedTabBar .....	23
DecoratedTabPanel .....	25
DecoratorPanel .....	25
DialogBox .....	25
DisclosurePanel .....	27

# GWT Composite Builder

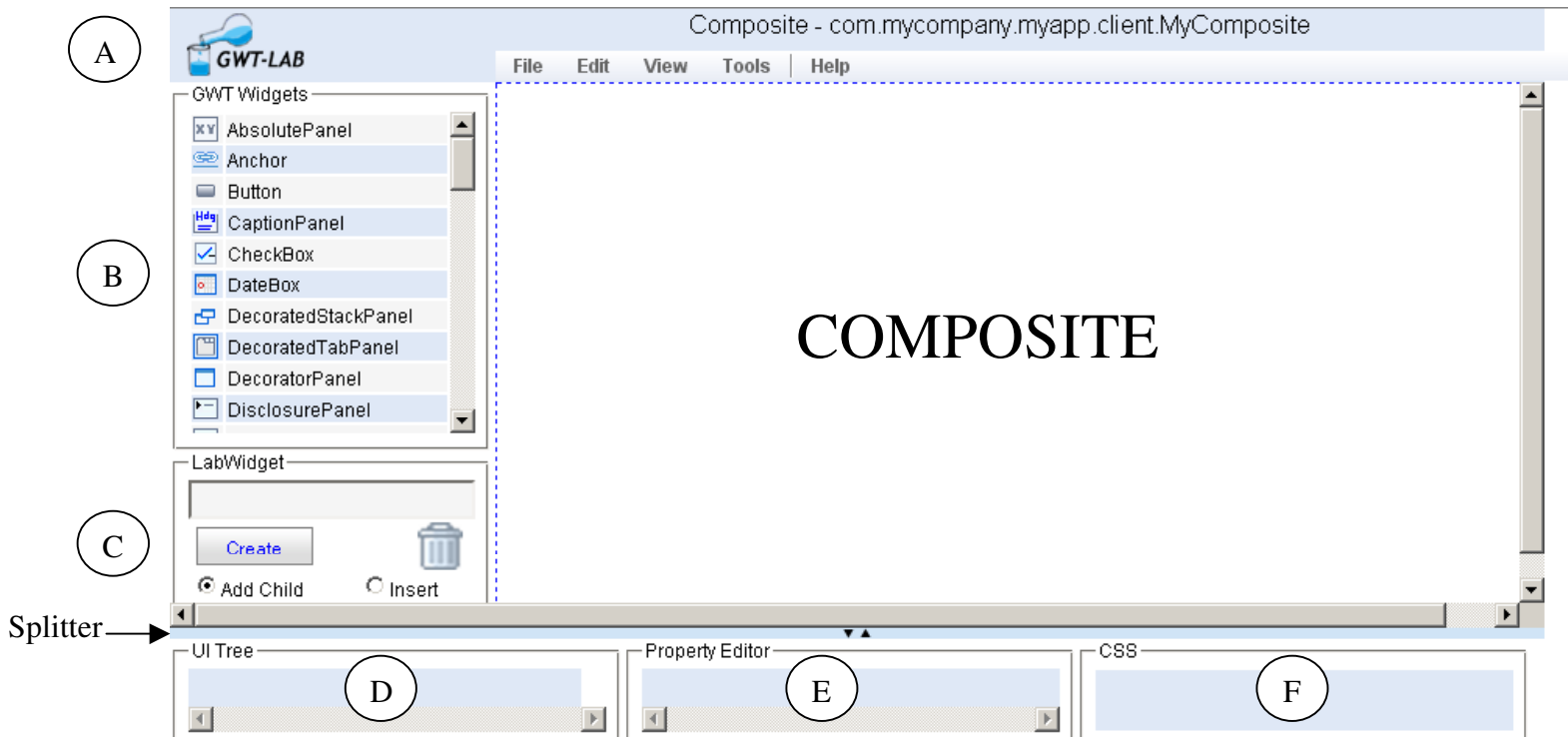
DockPanel .....	27
FileUpload .....	27
FlexRow .....	27
FlexTable .....	27
FlowPanel .....	27
FocusPanel .....	27
FormPanel .....	27
Frame .....	29
Grid .....	29
Hidden .....	29
HorizontalPanel .....	29
HorizontalSplitPanel .....	29
HTML .....	29
HTMLPanel .....	29
Hyperlink .....	30
Image .....	30
InlineHTML .....	30
InlineLabel .....	30
Label .....	30
ListBox .....	30
MenuBar .....	30
MenuBarV .....	32
MenuItem .....	32
MenuItemSeparator .....	32
Panel .....	32
PasswordTextBox .....	32
PopupPanel .....	32
PushButton .....	32
RadioButton .....	32
RichTextArea .....	32
RootPanel .....	32
ScrollPanel .....	32
SimpleCheckBox .....	32
SimplePanel .....	34
StackPanel .....	34
SuggestBox .....	34
TabBar .....	36
TabPanel .....	36
TextArea .....	36
TextBox .....	36
ToggleButton .....	36
Tree .....	36
TreeItem .....	36
VerticalPanel .....	36
VerticalSplitPanel .....	36
<b>GWTWidget Properties .....</b>	<b>38</b>
All .....	38
AbsolutePanel .....	38
Anchor .....	38
Button .....	38
CaptionPanel .....	39
CheckBox .....	39
DateBox .....	39
DecoratedStackPanel .....	39
DecoratedTabPanel .....	39
DecoratorPanel .....	39

# GWT Composite Builder

DisclosurePanel.....	39
DockPanel.....	39
FileUpload.....	39
FlexRow.....	39
FlexTable.....	39
FlowPanel.....	39
FocusPanel.....	39
FormPanel.....	39
Frame.....	39
Grid.....	39
Hidden.....	39
HorizontalPanel.....	40
HorizontalSplitPanel.....	40
HTML.....	40
Image.....	40
InlineHTML.....	40
InlineLabel.....	40
Label.....	40
ListBox.....	40
MenuBar.....	40
MenuBarV.....	40
MenuItem.....	40
MenuItemSeparator.....	40
PasswordTextBox.....	40
PushButton.....	41
RadioButton.....	41
RichTextArea.....	41
ScrollPanel.....	41
SimpleCheckBox.....	41
SimplePanel.....	41
StackPanel.....	41
SuggestBox.....	41
TabPanel.....	41
TextArea.....	41
TextBox.....	41
ToggleButton.....	41
Tree.....	41
TreeItem.....	41
VerticalPanel.....	41
VerticalSplitPanel.....	41

# GWT Composite Builder

## Overview



**Figure 1**

The GWT Composite Builder product simplifies the development of GWT Composite Widgets. When developing a custom widget a Composite is usually preferred to extending another widget because a Composite controls exposing its wrapped widget's methods and properties.

The Composite Builder was developed using Eclipse 3.4 and the Google Web Toolkit (GWT 1.6.4). The Composite Builder runs in a web browser (no Plugins) with JavaScript enabled. The Composite Builder has been tested on IE 7.0 and Firefox 3.0.9.

The 'COMPOSITE' region (surrounded by the dashed blue border) contains the visual representation of the Composite widget being created.

The Composite may be exported/imported to/from XML. The Composite may also be exported to Java, then copied and pasted into the user's favourite IDE, where business logic and event handling may be included.

Import of Java code is not supported.

If you require a product that supports bi-directional code generation then you might want to check out GWT Designer <http://www.instantiations.com/gwt designer/>

# GWT Composite Builder

## Acknowledgements

The GWT Composite Builder uses the gwt-dnd (2.5.6) library for Drag and Drop, which was developed by Fred Sauer and available from <http://allen-sauer.com/gwt>

The GWT Composite Builder uses native JavaScript code, to maintain a CSS Stylesheet, which is based on the code in the book JavaScript: The Definitive Guide, 5th Edition, by David Flanagan (ISBN: 0596101996)

The visual representation of the icons were inspired from:  
<http://download.instantiations.com/DesignerDoc/continuous/latest/docs/html/gwt/index.html>  
<http://www.pinvoke.com/>  
<http://www.famfamfam.com/lab/icons/silk/>

The following books, in no particular order, helped me to understand GWT:

Google Web Toolkit Solutions, More Cool & Useful Stuff  
David Geary with Rob Gordon  
2008 Pearson Education, Inc.  
ISBN 0-13-234481-5

Pro Web 2.0 Application Development with GWT  
Jeff Dwyer  
2008 Apress  
ISBN 1-59050-985-3

GWT in Action  
Easy Ajax with the Google Web Toolkit  
Robert Hanson, Adam Tacy  
2007 Manning Publications  
ISBN 1.933988-23-1

Google Web Toolkit Applications  
Ryan Dewsbury  
2008 Pearson Education, Inc  
ISBN 0-321-50196-9

GWT in Practice  
Robert T. Cooper, Charlie E. Collins  
2008 Manning Publications  
ISBN 1-933988-29-0

Last but not least I thank my wife Beverly for her support and letting me monopolize our computer. Also, thanks to our children Yvonne, Ian, and Graeme for their insights and suggestions.

# GWT Composite Builder

## Tool Structure

The Composite Builder has the following regions, shown in Figure 1 above.

### A.1 Heading

The heading section shows the Composite's Java package and class name.

You provide this information when the Composite Builder is started, or via the File Properties menu.

### A.2 Menu

The Menu section contains:

#### A.2.1 File

Export Java – displays a TextArea containing the Composite's Java code  
(Use copy/paste to save it in a local file)

Export XML – displays a TextArea containing the Composite as XML  
(Use copy/paste to save it in a local file)

Import XML – displays a TextArea to paste 'Exported XML' to rebuild the Composite.

Properties – provide the Composite's Java package and class name

#### A.2.2 Edit

Apply CSS – applies the CSS, entered in the CSS section to the COMPOSITE

Reset CSS – removes the CSS, entered in the CSS section from the COMPOSITE

#### A.2.3 View

Show in DialogBox – displays the COMPOSITE wrapped in a DialogBox

Show in DecoratedPopupPanel – displays the COMPOSITE wrapped in a  
DecoratedPopupPanel

Show in PopupPanel – displays the COMPOSITE wrapped in a PopupPanel

#### A.2.4 Tools

Show HTML Outline – displays a TextArea containing an outline of the Browser's DOM  
for the COMPOSITE. Useful when defining CSS selectors.

Show HTML Detail – displays a TextArea containing the InnerHTML for the  
COMPOSITE. Useful when defining CSS selectors and to see the DOM detail.

Clear UI – removes all LabWidgets and CSS from the COMPOSITE.

#### A.2.5 Help

User's Guide – This document.

About – summary information about the Composite Builder.

# GWT Composite Builder

## B - GWT Widgets

The GWT Widgets section contains the widgets that may be included in the Composite. Scroll through the list and select the type of widget that you wish to create. Clicking on a widget displays a sample name, for the LabWidget, that you may modify. Details for each GWT Widget is described later in this document.

## C - LabWidget

The LabWidget section controls the creation and removal of the Composite's widgets. A LabWidget (Laboratory Widget) wraps a GWT widget and stores its properties.

The LabWidget section contains:

A TextBox for entering the widget's Java coded name.  
The name you assign must be unique for all your LabWidgets in the Composite.

A 'Create' button to create the LabWidget and add it to the UI Tree.

Radio Buttons 'Add Child' and 'Insert' control where widgets are placed in the UI Tree.

Add Child – the LabWidget is added as a child to the selected item in the UI Tree.  
If the UI Tree is empty then the widget becomes the Composite's main, or Root widget.

Insert – the LabWidget is inserted before the item selected in the UI Tree.

To add a LabWidget to the UI Tree:

- . Assign a unique name to the LabWidget
- . Select the correct radio button (Add Child, or Insert)
- . Select its target item in the UI Tree
- . Click Create or press Enter

## Splitter

Drag the Splitter to resize the height of the UI Tree, Property Editor, and CSS sections in the bottom of the VerticalSplitPanel.

# GWT Composite Builder

## D - UI Tree

The UI Tree section is used to maintain the Composite widget's parent / child relationships.

Adding a widget to the UI Tree automatically displays it in the COMPOSITE.

LabWidgets are added to the UI Tree as described in section 3 above.

LabWidgets (and their children) may be rearranged within the UI Tree.

To change the UI Tree structure:

- . Select the correct radio button (Add Child, or Insert)
- . Press 'Shift' and click on the LabWidget to cut/move. (It will become highlighted)
- . Press 'Shift' and click on the paste/target LabWidget to move the highlighted Widget and its children to the target and automatically display the changes in the COMPOSITE.

Dragging a LabWidget to the Trash can (in section 3) removes it and its children from the UI Tree and from the COMPOSITE.

## E - Property Editor

The Property Editor section is used to maintain the UI Tree LabWidget's properties.

Selecting a widget in the UI Tree displays its properties in the Property Editor and any changes are automatically reflected in the COMPOSITE.

The Properties are grouped into four sections. Press a group's arrow to show/hide its properties.

General – common to all widgets, excluding the Style properties

Style – CSS styling properties

Specific – properties that are specific to the type of widget

Cell – properties when a widget is contained in a FlexTable, via a FlexRow

## F - CSS

The CSS section is used to enter CSS rules for the Composite.

To add or maintain CSS:

- . If required, assign the CSS style to a widget using the Property Editor
- . Enter the CSS selector and its properties in the CSS section  
For example, to add a border to the style 'mystyle'

```
.mystyle { border: 1px solid blue; }
```

Select 'Apply CSS', from the 'Edit' menu, to apply the rules to the COMPOSITE.

Use copy/paste to save the CSS rules in your application's css file.

# GWT Composite Builder

## **COMPOSITE**

The COMPOSITE section shows the visual representation of the Composite.

The appearance of the Composite is automatically changed when  
its structure is maintained in the UI Tree  
its widget's properties are changed in the Property Editor  
c) its CSS rules are applied or removed via the Edit menu

## **Comments / Feedback**

Please send your feedback to me at [jimcampsw@gmail.com](mailto:jimcampsw@gmail.com)

## Getting Started

This section describes the steps for creating the 'Contact Information' Composite, shown below.

1. Decide on the type of GWT widget to use as the wrapper for the Composite
2. Design the layout and structure of the contained widgets
3. Build the Composite
4. Assign the CSS Styles
5. Export XML and save in local file
6. Export Java and copy to favourite IDE
7. Modify Java code to implement required interfaces and business logic
8. Test

Figure 2

The screenshot shows a web form titled 'Personal Information' within a tabbed interface. The 'Personal' tab is active, showing input fields for 'My Name', 'Gender' (with radio buttons for 'Male' and 'Female'), 'Birthday', and 'I live in' (a dropdown menu currently showing 'Brazil' and 'Canada'). Below the form are 'New' and 'Update' buttons.

### 1. Decide on the GWT widget to use as the wrapper for the Composite

A DecoratedTabPanel widget has been chosen as the main widget for our Composite.

### 2. Design the layout and structure of the contained widgets

The layout and structure for the 'Personal' information tab is shown above

# GWT Composite Builder

## 3. Build steps

For this example, we will first add all of the components to the UI Tree then modify their properties in the Property Editor.

During the Build steps we will create the LabWidgets and assign their Text values through the Property Editor. In the next section 'CSS steps' we focus on their visual appearance.

### 3.1 Create LabWidgets

#### 3.1.1 Select 'DecoratedTabPanel' in the list of GWT Widgets

Enter its name (For example, myDecoratedTabPanel) in the LabWidget TextBox

Ensure that the 'Add Child' radio button is selected

Click the 'Create' button or press 'Enter' to add the LabWidget into the UI Tree

The 'myDecoratedTabPanel' should now be selected in the UI Tree and in the Property Editor.

The COMPOSITE section will become visible once a child widget is added.

#### 3.1.2 Select 'CaptionPanel' in the list of GWT Widgets

Enter its name (For example, personalCaptionPanel)

Click the 'Create' button or press 'Enter' to add the 'personalCaptionPanel' as a child of 'myDecoratedTabPanel' in the UI Tree

The COMPOSITE section should now display the 'personalCaptionPanel' as the first tab

#### 3.1.3 Select 'CaptionPanel' in the list of GWT Widgets, if not already selected

Enter its name (For example, otherCaptionPanel)

Click the 'Create' button or press 'Enter' to add the 'otherCaptionPanel' as a child of 'myDecoratedTabPanel' in the UI Tree

The COMPOSITE section should now display the 'otherCaptionPanel' as the second tab

#### 3.1.4 Add widgets to the 'personalCaptionPanel' Select 'personalCaptionPanel' in the UI Tree

(The Property Editor shows the selected widget in its heading)

To layout the widgets in the personalCaptionPanel we will use a FlexTable

Select 'FlexTable' in the list of GWT Widgets

Enter its name (For example, personalFlexTable)

Click the 'Create' button or press 'Enter' to add the 'personalFlexTable' as a child of 'personalCaptionPanel' in the UI Tree.

#### 3.1.5 Add widgets to the 'personalFlexTable' Select 'personalFlexTable' in the UI Tree

To simplify construction, the Composite Builder uses a 'FlexRow' to hold a row of widgets in a FlexTable. A 'FlexRow' is only used internally within the Composite Builder.

Select 'FlexRow' in the list of GWT Widgets

Enter its name (For example, nameFlexRow)

Click the 'Create' button or press 'Enter' to add the 'nameFlexRow' as a child of 'personalFlexTable' in the UI Tree

## GWT Composite Builder

**3.1.6 Similarly, add FlexRows for ‘genderFlexRow’, ‘birthdayFlexRow’, ‘locationFlexRow’, and ‘buttonFlexRow’ as children of ‘personalFlexTable’ in the UI Tree.**

**3.1.7 In a similar manner add widgets to the FlexRows**

To ‘nameFlexRow’ add a Label ‘nameLabel’, TextBox ‘firstNameTextBox’ and ‘lastNameTextBox’

To ‘genderFlexRow’ add a Label ‘genderLabel’, and RadioButtons ‘genderMaleRadioButton’ and ‘genderFemaleRadioButton’

To ‘birthdayFlexRow’ add a Label ‘birthdayLabel’ and a DateBox ‘birthdayDateBox’

To ‘locationFlexRow’ add a Label ‘locationLabel’ and a ListBox ‘locationListBox’

To ‘buttonFlexRow’ add Button ‘newButton’ and ‘updateButton’

Click on the first tab in the COMPOSITE to show its widgets.

**3.2 Modify LabWidgets Properties**

Let’s modify the COMPOSITE by changing some of its properties in the Property Editor.

If you need to rearrange widgets in the UI Tree refer to section D – UI Tree, above.

**3.2.1 Changes to ‘personalCaptionPanel’**

Select the ‘personalCaptionPanel’ widget in the UI Tree

Expand the Property Editor’s ‘Specific’ panel by clicking on its heading

Enter ‘Personal’ in the ‘TabText’ field

Enter ‘Personal Information’ in the ‘CaptionText’ field

**3.2.2 Changes to ‘personalFlexTable’**

Select the ‘personalFlexTable’ widget in the UI Tree

Expand the Property Editor’s ‘General’ panel by clicking on its heading

Enter ‘100%’ in the ‘Width’ field

Expand the Property Editor’s ‘Cell’ panel by clicking on its heading

Enter ‘5’ in the ‘CellPadding’ field

**3.2.3 Changes to the widgets in ‘nameFlexRow’**

Select the ‘nameLabel’ widget in the UI Tree

Expand the Property Editor’s ‘Specific’ panel by clicking on its heading

Enter ‘My Name:’ in the ‘Text’ field

## GWT Composite Builder

### **3.2.4 Changes to the widgets in 'genderFlexRow'**

Select the 'genderLabel' widget in the UI Tree

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'Gender:' in the 'Text' field

### **3.2.5 Select the 'genderMaleRadioButton' widget in the UI Tree**

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'Male' in the 'Text' field

### **3.2.6 Select the 'genderFemaleRadioButton' widget in the UI Tree**

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'Female' in the 'Text' field

### **3.2.7 Changes to the widgets in 'birthdayFlexRow'**

Select the 'birthdayLabel' widget in the UI Tree

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'Birthday:' in the 'Text' field

### **3.2.8 Changes to the widgets in 'locationFlexRow'**

Select the 'locationLabel' widget in the UI Tree

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'I live in:' in the 'Text' field

### **3.2.9 Select the 'locationListBox' widget in the UI Tree**

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'Brazil, Canada, United States' in the 'AddItem' field

### **3.2.10 Changes to the widgets in 'buttonFlexRow'**

Select the 'newButton' widget in the UI Tree

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'New' in the 'Text' field

Expand the Property Editor's 'Cell' panel by clicking on its heading

Select 'ALIGN\_RIGHT' from the 'CellHorizontalAlignment' dropdown

### **3.2.11 Select the 'updateButton' widget in the UI Tree**

Expand the Property Editor's 'Specific' panel by clicking on its heading

Enter 'Update' in the 'Text' field

Changes to the second Tab (otherCaptionPanel) are left as an exercise for the reader.

## 4. CSS steps

### 4.1 Changes to 'myDecoratedTabPanel'

Select the 'myDecoratedTabPanel' widget in the UI Tree  
Expand the Property Editor's 'Style' panel by clicking on its heading  
Enter 'contact' in the 'StyleName\_ADD' field  
(Note: using 'StyleName\_SET' would replace the default styling)

### 4.2 Changes to 'personalFlexTable'

Select the 'personalFlexTable' widget in the UI Tree  
Expand the Property Editor's 'Style' panel by clicking on its heading  
Enter 'personalFlex' in the 'StyleName\_SET' field

### 4.3 Select Tools → Show HTML Outline from the menu

Show the outline of the COMPOSITE and see the CSS class names that you assigned

### 4.4 Enter the following CSS in the CSS section (the bottom right section of the Composite Builder)

```
.contact {  
width: 500px;  
}
```

```
.gwt-TabPanelBottom {  
background-color: #D8E8F8;  
}
```

```
.personalFlex .gwt-Label {  
width: 100px;  
text-align: right;  
}
```

```
.personalFlex .gwt-DateBox,  
.personalFlex .gwt-TextBox,  
.personalFlex .gwt-ListBox {  
width: 150px;  
}
```

### 4.5 Select Edit → Apply CSS from the menu

To apply and see the CSS changes

Your Composite should now look similar to the target 'Contact Information' Composite.

# GWT Composite Builder

## 5. Export / Import XML

### 5.1 select File → Export XML from the menu

A DialogBox will display the COMPOSITE's XML

### 5.2 Copy and Paste the XML in a local file

The XML may then be Imported via the File → Import XML menu to make subsequent changes.

### 5.3 Try to Export then Import the XML for the COMPOSITE

## 6. Export Java

### 6.1 select File → Export Java from the menu

A DialogBox will display the COMPOSITE's Java code.

### 6.2 Copy and Paste the Java code to your favourite IDE

## 7. Implement any interfaces and business logic, as required

Import of Java code into the Composite Builder is not supported.

## 8. Test

Integrate your modified Java code into your GWT application's source code.

Copy / paste the CSS rules to your application's css file.

Test as normal.

# GWT Composite Builder

## Working with the GWT Composite Builder

The following tips/observations may help you when building your composite.

Disclosure Panel - Specifying the Height of the parent container of Disclosure Panels, causes a gap between the Disclosure Panels.

During development it may be helpful to add a border to a container to see its boundary.

For example,

add a style 'outline' to a component via the Property Editor

enter the CSS below in the CSS section

Apply CSS from the Edit Menu.

Note: Style names entered in the Property Editor have no leading period (.), while style names entered in the CSS section do have a leading period (.).

```
.outline {  
  border: 1px solid grey;  
}
```

The border may not be shown until the container contains at least 1 widget.

Vertical placement of components may be achieved by using a FlowPanel and specifying a width of 100% for its components, as an alternative to using a Vertical Panel.

Setting the height of a VerticalPanel may cause space between vertical components to fill the height.

ListBox – set the height property in the Property Editor, to see ListBox items.

To add multiple items to a ListBox enter the items delimited by comma, in the ListBox's AddItem property.

To enter multiple suggestions for a SuggestBox enter the suggestions, delimited by comma, in the SuggestBox's Suggest property.

## GWT Composite Builder

Anchor	The following target names are reserved and have special meanings.
_blank	The user agent should load the designated document in a new, unnamed window.
_self	The user agent should load the document in the same frame as the element that refers to this target.
_parent	The user agent should load the document into the immediate FRAMESET parent of the current frame. This value is equivalent to _self if the current frame has no parent.
_top	The user agent should load the document into the full, original window (thus canceling all other frames).

For example, entering 'http://www.google.ca' and '\_blank' in the Property Editor's 'Href', and 'Target' properties, respectively will open up a new window for Google.

Only TreeItems may be children of a Tree.

A MenuItem (with a Command) or a MenuSeparator may only be added to a MenuBar (or MenuBarV). Adding a MenuBar (or MenuBarV) to a MenuItem adds a submenu.

Two modes are provided to populate a Grid:

- a) The default quick mode fills the grid cells with row and column strings (Rn, Cm). Entering values for RowCount or ColumnCount, via the Property Editor, resizes and populates the cell contents.
- b) The second mode is used to fill a column with widgets. Adding a widget to a grid fills a column of corresponding widgets.
  - row 0 – contains the column Headers, as Labels
  - row 1 – subsequent rows contain widgets of the added type

The Composite Builder restricts the widgets that may be added to a Grid to:

Anchor, Button, CheckBox, DateBox, Hidden, Image, Label, ListBox, PasswordTextBox, PushButton, RadioButton, SimpleCheckBox, TextArea, TextBox, and ToggleButton.

To add widgets to a FlexTable, first add a FlexRow(s) to the FlexTable then add widgets to the FlexRow(s).

## GWT Composite Builder

The COMPOSITE may be shown in a DialogBox, DecoratedPopupPanel, or PopupPanel via the View menu. Java is not generated for these Popup Panels, but may be easily added to the Java that is generated for the COMPOSITE.

For example, the Java to include the Composite named MyComposite, in a DialogBox.

```
final MyComposite widget = new MyComposite();
final DialogBox popup = new DialogBox(true); // auto-hide
popup.setText("MyComposite Dialog Heading");

popup.addCloseHandler(new CloseHandler<PopupPanel>() {
    public void onClose(CloseEvent<PopupPanel> event) {
        // do something
        popup.hide();
    }
});

popup.setWidget(widget);

popup.show();
```

To create a Horizontal Ruler, create an HTML widget and set its HTML to `<hr/>`

The first LabWidget created becomes the Root of the UI Tree and it must remain the Root, unless it is dragged to the trash, which will remove it and all the LabWidgets from the UI Tree.

If required, you could modify the exported XML to include a new Root then import the modified XML. (Modifying the XML is beyond the scope of this document)

If the Root widget is not displaying its expected style then you may have overridden its default styling. Check the Tools → HTML Outline output for the class assigned to the Root widget.

For example,

If the Root widget is a DecoratorPanel and it is not showing a decorated border then check that its HTML is similar to:

```
<TABLE class="gwt-DecoratorPanel">
```

If not then use the Property Editor to set the expected css class names.

If you change the appearance of the UI Composite directly, rather than through the Property Editor, then the Export/Import of XML may not be what you expected, as only changes applied via the Property Editor are included. For example, if you select a RadioButton in the UI Composite and do not change the 'Checked' property to true in the Property Editor then it would be unselected if you were to Export / Import XML.

# GWT Composite Builder

## GWT Widget Detail

The following list contains the GWT Widgets supported by the GWT Composite Builder. The descriptions are from the GWT API.

Type	Description / CSS	Comment
AbsolutePanel	<p>An absolute panel positions all of its children absolutely, allowing them to overlap.</p> <p>Note that this panel will not automatically resize itself to allow enough room for its absolutely positioned children. It must be explicitly sized in order to make room for them.</p> <p>CSS – no default</p>	
Anchor	<p>A widget that represents a simple &lt;a&gt; element.</p> <p>.gwt-Anchor { }</p>	
Button	<p>A standard push-button widget</p> <p>.gwt-Button { }</p>	
CaptionPanel	<p>A panel that wraps its contents in a border with a caption that appears in the upper left corner of the border. This is an implementation of the fieldset HTML element.</p> <p>CSS – no default</p>	
CheckBox	<p>A standard check box widget. This class also serves as a base class for RadioButton.</p> <p>.gwt-CheckBox the outer element .gwt-CheckBox-disabled applied when Checkbox is disabled</p>	
Composite	<p>A type of widget that can wrap another widget, hiding the wrapped widget's methods. When added to a panel, a composite behaves exactly as if the widget it wraps had been added.</p>	<p>Not included in the GWT Widget list. The top widget in the UI Tree becomes the main component of the Composite being created.</p>
DateBox	<p>DateBox with Date picker.</p> <p>.gwt-DateBox default style name .dateBoxPopup Applied to the popup around the DatePicker .dateBoxFormatError Default style for when the date box has bad input. Applied by DateBox.DefaultFormat when the text does not represent a date that can be parsed</p>	<p>Added in GWT 1.6</p>
DeckPanel	<p>A panel that displays all of its child widgets in a 'deck', where only one can be visible at a time. TabPanel uses it.</p> <p>CSS – no default</p>	<p>Not included in the GWT Widget list. Used by its TabPanel</p>
DecoratedPopupPanel	<p>A PopupPanel that wraps its content in a 3x3 grid, which</p>	<p>Via View Menu</p>

## GWT Composite Builder

Type	Description / CSS	Comment
	<p>allows users to add rounded</p> <p>.gwt-DecoratedPopupPanel { the outside of the popup }</p> <p>.gwt-DecoratedPopupPanel .popupContent { the wrapper around the content }</p> <p>.gwt-DecoratedPopupPanel .popupTopLeft { the top left cell }</p> <p>.gwt-DecoratedPopupPanel .popupTopLeftInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupTopCenter { the top center cell }</p> <p>.gwt-DecoratedPopupPanel .popupTopCenterInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupTopRight { the top right cell }</p> <p>.gwt-DecoratedPopupPanel .popupTopRightInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupMiddleLeft { the middle left cell }</p> <p>.gwt-DecoratedPopupPanel .popupMiddleLeftInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupMiddleCenter { the middle center cell }</p> <p>.gwt-DecoratedPopupPanel .popupMiddleCenterInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupMiddleRight { the middle right cell }</p> <p>.gwt-DecoratedPopupPanel .popupMiddleRightInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupBottomLeft { the bottom left cell }</p> <p>.gwt-DecoratedPopupPanel .popupBottomLeftInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupBottomCenter { the bottom center cell }</p> <p>.gwt-DecoratedPopupPanel .popupBottomCenterInner { the inner element of the cell }</p> <p>.gwt-DecoratedPopupPanel .popupBottomRight { the bottom right cell }</p> <p>.gwt-DecoratedPopupPanel .popupBottomRightInner { the inner element of the cell }</p> <p>corners.</p>	

## GWT Composite Builder

Type	Description / CSS	Comment
DecoratedStackPanel	<p>A StackPanel that wraps each item in a 2x3 grid (six box), which allows users to add rounded corners.</p> <pre>.gwt-DecoratedStackPanel { the panel itself } .gwt-DecoratedStackPanel .gwt-StackPanellItem { unselected items } .gwt-DecoratedStackPanel .gwt-StackPanellItem-selected { selected items } .gwt-DecoratedStackPanel .gwt-StackPanelContent { the wrapper around the contents of the item } .gwt-DecoratedStackPanel .stackItemTopLeft { top left corner of the item} .gwt-DecoratedStackPanel .stackItemTopLeftInner { the inner element of the cell} .gwt-DecoratedStackPanel .stackItemTopCenter { top center of the item} .gwt-DecoratedStackPanel .stackItemTopCenterInner { the inner element of the cell} .gwt-DecoratedStackPanel .stackItemTopRight { top right corner of the item} .gwt-DecoratedStackPanel .stackItemTopRightInner { the inner element of the cell} .gwt-DecoratedStackPanel .stackItemMiddleLeft { left side of the item } .gwt-DecoratedStackPanel .stackItemMiddleLeftInner { the inner element of the cell} .gwt-DecoratedStackPanel .stackItemMiddleCenter { center of the item, where the item text resides } .gwt-DecoratedStackPanel .stackItemMiddleCenterInner { the inner element of the cell} .gwt-DecoratedStackPanel .stackItemMiddleRight { right side of the item } .gwt-DecoratedStackPanel .stackItemMiddleRightInner { the inner element of the cell}</pre>	
DecoratedTabBar	<p>A TabBar that wraps each tab in a 2x3 grid (six box), which allows users to add rounded corners.</p> <pre>.gwt-DecoratedTabBar { the tab bar itself } .gwt-DecoratedTabBar .gwt-TabBarFirst { the left edge of the bar } .gwt-DecoratedTabBar .gwt-TabBarRest { the right edge of the bar } .gwt-DecoratedTabBar .gwt-TabBarItem { unselected tabs } .gwt-DecoratedTabBar .gwt-TabBarItem-wrapper { table cell around tab } .gwt-DecoratedTabBar .gwt-TabBarItem-selected { additional style for selected tabs } .gwt-DecoratedTabBar .gwt-TabBarItem-wrapper-selected { table cell around selected tab } .gwt-DecoratedTabBar .tabTopLeft { top left corner of the tab} .gwt-DecoratedTabBar .tabTopLeftInner { the inner element of the cell} .gwt-DecoratedTabBar .tabTopCenter { top center of the tab} .gwt-DecoratedTabBar .tabTopCenterInner { the inner</pre>	Not included in the GWT Widget list. Used by its TabPanel.

## GWT Composite Builder

Type	Description / CSS	Comment
	<p>element of the cell}</p> <p>.gwt-DecoratedTabBar .tabTopRight { top right corner of the tab}</p> <p>.gwt-DecoratedTabBar .tabTopRightInner { the inner element of the cell}</p> <p>.gwt-DecoratedTabBar .tabMiddleLeft { left side of the tab }</p> <p>.gwt-DecoratedTabBar .tabMiddleLeftInner { the inner element of the cell}</p> <p>.gwt-DecoratedTabBar .tabMiddleCenter { center of the tab, where the tab text or widget resides }</p> <p>.gwt-DecoratedTabBar .tabMiddleCenterInner { the inner element of the cell}</p> <p>.gwt-DecoratedTabBar .tabMiddleRight { right side of the tab }</p> <p>.gwt-DecoratedTabBar .tabMiddleRightInner { the inner element of the cell}</p>	

## GWT Composite Builder

Type	Description / CSS	Comment
DecoratedTabPanel	<p>A TabPanel that uses a DecoratedTabBar with rounded corners.</p> <pre>.gwt-DecoratedTabPanel { the tab panel itself } .gwt-TabPanelBottom { the bottom section of the tab panel (the deck containing the widget) }</pre>	
DecoratorPanel	<p>A SimplePanel that wraps its contents in stylized boxes, which can be used to add rounded corners to a Widget.</p> <pre>.gwt-DecoratorPanel { the panel } .gwt-DecoratorPanel .top { the top row } .gwt-DecoratorPanel .topLeft { the top left cell } .gwt-DecoratorPanel .topLeftInner { the inner element of the cell } .gwt-DecoratorPanel .topCenter { the top center cell } .gwt-DecoratorPanel .topCenterInner { the inner element of the cell } .gwt-DecoratorPanel .topRight { the top right cell } .gwt-DecoratorPanel .topRightInner { the inner element of the cell } .gwt-DecoratorPanel .middle { the middle row } .gwt-DecoratorPanel .middleLeft { the middle left cell } .gwt-DecoratorPanel .middleLeftInner { the inner element of the cell } .gwt-DecoratorPanel .middleCenter { the middle center cell } .gwt-DecoratorPanel .middleCenterInner { the inner element of the cell } .gwt-DecoratorPanel .middleRight { the middle right cell } .gwt-DecoratorPanel .middleRightInner { the inner element of the cell } .gwt-DecoratorPanel .bottom { the bottom row } .gwt-DecoratorPanel .bottomLeft { the bottom left cell } .gwt-DecoratorPanel .bottomLeftInner { the inner element of the cell } .gwt-DecoratorPanel .bottomCenter { the bottom center cell } .gwt-DecoratorPanel .bottomCenterInner { the inner element of the cell } .gwt-DecoratorPanel .bottomRight { the bottom right cell } .gwt-DecoratorPanel .bottomRightInner { the inner element of the cell }</pre>	
DialogBox	<p>A form of popup that has a caption area at the top and can be dragged by the user.</p> <pre>.gwt-DialogBox { the outside of the dialog } .gwt-DialogBox .Caption { the caption } .gwt-DialogBox .dialogContent { the wrapper around the content } .gwt-DialogBox .dialogTopLeft { the top left cell } .gwt-DialogBox .dialogTopLeftInner { the inner element of the cell } .gwt-DialogBox .dialogTopCenter { the top center cell, where the caption is located } .gwt-DialogBox .dialogTopCenterInner { the inner element of</pre>	Via View Menu

## GWT Composite Builder

Type	Description / CSS	Comment
	the cell } .gwt-DialogBox .dialogTopRight { the top right cell } .gwt-DialogBox .dialogTopRightInner { the inner element of the cell } .gwt-DialogBox .dialogMiddleLeft { the middle left cell } .gwt-DialogBox .dialogMiddleLeftInner { the inner element of the cell } .gwt-DialogBox .dialogMiddleCenter { the middle center cell, where the content is located } .gwt-DialogBox .dialogMiddleCenterInner { the inner element of the cell } .gwt-DialogBox .dialogMiddleRight { the middle right cell } .gwt-DialogBox .dialogMiddleRightInner { the inner element of the cell } .gwt-DialogBox .dialogBottomLeft { the bottom left cell } .gwt-DialogBox .dialogBottomLeftInner { the inner element of the cell } .gwt-DialogBox .dialogBottomCenter { the bottom center cell } .gwt-DialogBox .dialogBottomCenterInner { the inner element of the cell } .gwt-DialogBox .dialogBottomRight { the bottom right cell } .gwt-DialogBox .dialogBottomRightInner { the inner element of the cell }	

## GWT Composite Builder

Type	Description / CSS	Comment
DisclosurePanel	<p>A widget that consists of a header and a content panel that discloses the content when a user clicks on the header.</p> <pre>.gwt-DisclosurePanel { the panel's primary style } .gwt-DisclosurePanel-open { dependent style set when panel is open } .gwt-DisclosurePanel-closed { dependent style set when panel is closed }</pre> <p>The header and content sections can be easily selected using css with a child selector:  <pre>.gwt-DisclosurePanel-open .header { ... }</pre></p>	
DockPanel	<p>A panel that lays its child widgets out "docked" at its outer edges, and allows its last widget to take up the remaining space in its center.</p> <p>CSS – no default</p>	
FileUpload	<p>A widget that wraps the HTML &lt;input type='file'&gt; element. This widget must be used with FormPanel if it is to be submitted to a server.</p> <pre>.gwt-FileUpload{}</pre>	
FlexRow	<p>A custom widget to group a row of FlexTable widgets.</p> <p>CSS – not applicable</p>	A structural widget used only in the UI Tree to simplify building a FlexTable.
FlexTable	<p>A flexible table that creates cells on demand. It can be jagged (that is, each row can contain a different number of cells) and individual cells can be set to span multiple rows or columns.</p> <p>CSS – no default</p>	
FlowPanel	<p>A panel that formats its child widgets using the default HTML layout behavior.</p>	
FocusPanel	<p>A simple panel that makes its contents focusable, and adds the ability to catch mouse and keyboard events.</p> <p>CSS – no default</p>	
FormPanel	<p>A panel that wraps its contents in an HTML &lt;FORM&gt; element. The following widgets (those that implement HasName) will be submitted to the server if they are contained within this panel:</p> <ul style="list-style-type: none"> <li>* TextBox</li> <li>* PasswordTextBox</li> <li>* RadioButton</li> <li>* SimpleRadioButton</li> <li>* CheckBox</li> <li>* SimpleCheckBox</li> <li>* TextArea</li> <li>* ListBox</li> <li>* FileUpload</li> <li>* Hidden</li> </ul>	

# GWT Composite Builder

<b>Type</b>	<b>Description / CSS</b>	<b>Comment</b>
	CSS – no default	

## GWT Composite Builder

Type	Description / CSS	Comment
Frame	<p>A widget that wraps an IFRAME element, which can contain an arbitrary web site. If you are using History, any browser history items generated by the Frame will interleave with your application's history.</p> <p>.gwt-Frame { }</p>	<p>[WARN] Confirmation was required to visit untrusted URL: 'http://www.google.com/'</p> <p>[WARN] reason: http://www.google.com / is not in the whitelist</p> <p>[WARN] To fix: add regex matching URL to -whitelist command line argument</p> <p>[WARN] Example: -whitelist="^http:[/][/]www[.]google[.]com"</p> <p>[WARN] To reject automatically: add regex matching URL to -blacklist command line argument</p> <p>[WARN] Example: -blacklist="^http:[/][/]www[.]google[.]com"</p>
Grid	<p>A rectangular grid that can contain text, html, or a child Widget within its cells. It must be resized explicitly to the desired number of rows and columns.</p> <p>CSS – no default</p>	
Hidden	<p>Represents a hidden field in an HTML form.</p> <p>CSS – not applicable</p>	
HorizontalPanel	<p>A panel that lays all of its widgets out in a single horizontal column.</p> <p>CSS – no default</p>	
HorizontalSplitPanel	<p>A panel that arranges two widgets in a single horizontal row and allows the user to interactively change the proportion of the width dedicated to each of the two widgets. Widgets contained within a HorizontalSplitPanel will be automatically decorated with scrollbars when necessary.</p> <p>.gwt-HorizontalSplitPanel { the panel itself }</p> <p>.gwt-HorizontalSplitPanel hsplitter { the splitter }</p>	
HTML	<p>A widget that can contain arbitrary HTML. This widget uses a &lt;div&gt; element, causing it to be displayed with block layout.</p> <p>.gwt-HTML { }</p>	
HTMLPanel	<p>A panel that contains HTML, and which can attach child widgets to identified elements within that HTML.</p> <p>CSS – no default</p>	Future

## GWT Composite Builder

Type	Description / CSS	Comment
Hyperlink	<p>A widget that serves as an "internal" hyperlink. That is, it is a link to another state of the running application. When clicked, it will create a new history frame using History.newItem (java.lang.String), but without reloading the page.</p> <p><code>.gwt-Hyperlink { }</code></p>	Future
Image	<p>A widget that displays the image at a given URL. The image can be in 'unclipped' mode (the default) or 'clipped' mode.</p> <p><code>.gwt-Image { }</code></p>	
InlineHTML	<p>A widget that can contain arbitrary HTML. This widget uses a &lt;span&gt; element, causing it to be displayed with inline layout.</p> <p><code>.gwt-InlineHTML { }</code></p>	
InlineLabel	<p>A widget that contains arbitrary text not interpreted as HTML. This widget uses a &lt;span&gt; element, causing it to be displayed with inline layout.</p> <p><code>.gwt-InlineLabel { }</code></p>	
Label	<p>A widget that contains arbitrary text not interpreted as HTML. This widget uses a &lt;div&gt; element, causing it to be displayed with block layout.</p> <p><code>.gwt-Label { }</code></p>	
ListBox	<p>A widget that presents a list of choices to the user, either as a list box or as a drop-down list.</p> <p><code>.gwt-ListBox { }</code></p>	<p>Defaults to multiple select because the setMultipleSelect method is deprecated in GWT 1.6. For single select modify the constructor in the generated Java code. To add multiple items enter them in the AddItem property delimited by comma</p>
MenuBar	<p>A standard horizontal menu bar widget. A menu bar can contain any number of menu items, each of which can either fire a Command or open a cascaded menu bar.</p> <p><code>.gwt-MenuBar</code> the menu bar itself <code>.gwt-MenuBar-horizontal</code> dependent style applied to horizontal menu bars <code>.gwt-MenuBar-vertical</code> dependent style applied to vertical menu bars <code>.gwt-MenuBar .gwt-MenuItem</code> menu items <code>.gwt-MenuBar .gwt-MenuItem-selected</code> selected menu items <code>.gwt-MenuBar .gwt-MenuItemSeparator</code> section breaks between menu items <code>.gwt-MenuBar .gwt-MenuItemSeparator .menuSeparatorInner</code></p>	

## GWT Composite Builder

Type	Description / CSS	Comment
	inner component of section separators .gwt-MenuBarPopup .menuPopupTopLeft the top left cell .gwt-MenuBarPopup .menuPopupTopLeftInner the inner element of the cell .gwt-MenuBarPopup .menuPopupTopCenter the top center cell .gwt-MenuBarPopup .menuPopupTopCenterInner the inner element of the cell .gwt-MenuBarPopup .menuPopupTopRight the top right cell .gwt-MenuBarPopup .menuPopupTopRightInner the inner element of the cell .gwt-MenuBarPopup .menuPopupMiddleLeft the middle left cell .gwt-MenuBarPopup .menuPopupMiddleLeftInner the inner element of the cell .gwt-MenuBarPopup .menuPopupMiddleCenter the middle center cell .gwt-MenuBarPopup .menuPopupMiddleCenterInner the inner element of the cell .gwt-MenuBarPopup .menuPopupMiddleRight the middle right cell .gwt-MenuBarPopup .menuPopupMiddleRightInner the inner element of the cell .gwt-MenuBarPopup .menuPopupBottomLeft the bottom left cell .gwt-MenuBarPopup .menuPopupBottomLeftInner the inner element of the cell .gwt-MenuBarPopup .menuPopupBottomCenter the bottom center cell .gwt-MenuBarPopup .menuPopupBottomCenterInner the inner element of the cell .gwt-MenuBarPopup .menuPopupBottomRight the bottom right cell .gwt-MenuBarPopup .menuPopupBottomRightInner the inner element of the cell	

## GWT Composite Builder

Type	Description / CSS	Comment
MenuBarV	A standard vertical menu bar widget. A menu bar can contain any number of menu items, each of which can either fire a Command or open a cascaded menu bar.  CSS - see MenuBar	Same as MenuBar but Vertical. Required because GWT does not have a property to change MenuBar orientation (must be specified in the constructor)
MenuItem	A widget that can be placed in a MenuBar  CSS – see MenuBar	
MenuItemSeparator	A separator that can be placed in a MenuBar.  CSS – see MenuBar	
Panel	Abstract base class for all panels, which are widgets that can contain other widgets.	Not included in the GWT Widget list.
PasswordTextBox	A text box that visually masks its input to prevent eavesdropping.  .gwt-PasswordTextBox { primary style } .gwt-PasswordTextBox-readonly { dependent style set when the password text box is read-only }	
PopupPanel	A panel that can "pop up" over other widgets. It overlays the browser's client area (and any previously created popups).  A PopupPanel should not generally be added to other panels; rather, it should be shown and hidden using the show () and hide () methods.  .gwt-PopupPanel the outside of the popup .gwt-PopupPanel .popupContent the wrapper around the content	Via View Menu
PushButton	A normal push button with custom styling.  .gwt-PushButton-up/down/up-hovering/down-hovering/up-disabled/down-disabled { .html-face }	
RadioButton	A mutually exclusive selection radio button widget.  .gwt-RadioButton { }	
RichTextArea	A rich text editor that allows complex styling and formatting.  .gwt-RichTextArea { }	For examples: See the GWT Showcase, or GWT Applications by Ryan Dewsbury
RootPanel	The panel to which all other widgets must ultimately be added. RootPanels are never created directly. Rather, they are accessed via get ().	Not included in the GWT Widget list.
ScrollPanel	A simple panel that wraps its contents in a scrollable area.  CSS – no default	
SimpleCheckBox	A simple checkbox widget, with no label.	

# GWT Composite Builder

Type	Description / CSS	Comment
	<pre>.gwt-SimpleCheckBox { } .gwt-SimpleCheckBox-disabled { Applied when checkbox is disabled }</pre>	

## GWT Composite Builder

Type	Description / CSS	Comment
SimplePanel	<p>Base class for panels that contain only one widget.</p> <p>CSS – no default</p>	
StackPanel	<p>A panel that stacks its children vertically, displaying only one at a time, with a header for each child which the user can click to display.</p> <p>.gwt-StackPanel { the panel itself }</p> <p>.gwt-StackPanel .gwt-StackPanelItem { unselected items }</p> <p>.gwt-StackPanel .gwt-StackPanelItem-selected { selected items }</p> <p>.gwt-StackPanel .gwt-StackPanelContent { the wrapper around the contents of the item }</p>	
SuggestBox	<p>A SuggestBox is a text box or text area, which displays a pre-configured set of selections that match the user's input.</p> <p>.gwt-SuggestBox the suggest box itself</p> <p>.gwt-SuggestBoxPopup the suggestion popup</p> <p>.gwt-SuggestBoxPopup .item an unselected suggestion</p> <p>.gwt-SuggestBoxPopup .item-selected a selected suggestion</p> <p>.gwt-SuggestBoxPopup .suggestPopupTopLeft the top left cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupTopLeftInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupTopCenter the top center cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupTopCenterInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupTopRight the top right cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupTopRightInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupMiddleLeft the middle left cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupMiddleLeftInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupMiddleCenter the middle center cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupMiddleCenterInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupMiddleRight the middle right cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupMiddleRightInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupBottomLeft the bottom left cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupBottomLeftInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupBottomCenter the bottom center cell</p>	<p>To add multiple suggestions enter them in the Suggest property delimited by comma</p>

# GWT Composite Builder

<b>Type</b>	<b>Description / CSS</b>	<b>Comment</b>
	<p>.gwt-SuggestBoxPopup .suggestPopupBottomCenterInner the inner element of the cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupBottomRight the bottom right cell</p> <p>.gwt-SuggestBoxPopup .suggestPopupBottomRightInner the inner element of the cell</p>	

## GWT Composite Builder

Type	Description / CSS	Comment
TabBar	<p>A horizontal bar of folder-style tabs most commonly used as part of a TabPanel.</p> <pre>.gwt-TabBar { the tab bar itself } .gwt-TabBar .gwt-TabBarFirst { the left edge of the bar } .gwt-TabBar .gwt-TabBarFirst-wrapper { table cell around the left edge } .gwt-TabBar .gwt-TabBarRest { the right edge of the bar } .gwt-TabBar .gwt-TabBarRest-wrapper { table cell around the right edge } .gwt-TabBar .gwt-TabBarItem { unselected tabs } .gwt-TabBar .gwt-TabBarItem-wrapper { table cell around tab } .gwt-TabBar .gwt-TabBarItem-selected { additional style for selected</pre>	<p>Not included in the GWT Widget list. Used by the TabPanel</p>
TabPanel	<p>A panel that represents a tabbed set of pages, each of which contains another widget.</p> <pre>.gwt-TabPanel { the tab panel itself } .gwt-TabPanelBottom { the bottom section of the tab panel (the deck containing the widget) }</pre>	<p>Add a TabPanel inside another panel that has width set, then set TabPanel width.</p>
TextArea	<p>A text box that allows multiple lines of text.</p> <pre>.gwt-TextArea { primary style } .gwt-TextArea-readonly { dependent style set when the text area is read-only }</pre>	
TextBox	<p>A standard single-line text box.</p> <pre>.gwt-TextBox { primary style } .gwt-TextBox-readonly { dependent style set when the text box is read-only }</pre>	
ToggleButton	<p>A ToggleButton is a stylish stateful button, which allows the user to toggle between up and down states.</p> <pre>.gwt-ToggleButton-up/down/up-hovering/down-hovering/up-disabled/down-disabled { .html-face }</pre>	
Tree	<p>A standard hierarchical tree widget. The tree contains a hierarchy of TreeItems that the user can open, close, and select.</p> <pre>.gwt-Tree   the tree itself .gwt-Tree .gwt-TreeItem   a tree item .gwt-Tree .gwt-TreeItem-selected   a selected tree item</pre>	
TreeItem	<p>An item that can be contained within a Tree.</p> <p>CSS – see Tree</p>	
VerticalPanel	<p>A panel that lays all of its widgets out in a single vertical column.</p> <p>CSS – no default</p>	
VerticalSplitPanel	<p>A panel that arranges two widgets in a single vertical column</p>	

# GWT Composite Builder

Type	Description / CSS	Comment
	<p>and allows the user to interactively change the proportion of the height dedicated to each of the two widgets. Widgets contained within a VerticalSplitterPanel will be automatically decorated with scrollbars when necessary.</p> <pre>.gwt-VerticalSplitPanel { the panel itself } .gwt-VerticalSplitPanel vsplitter { the splitter }</pre>	

# GWT Composite Builder

## GWTWidget Properties

The Properties supported by the GWT Widgets are listed below.  
Where possible the Property name corresponds to the GWT Property.

GWTWidget	Property	Description	Sample
All	GWT_Type	Read only. The type of widget	VerticalPanel
	Id	The element's identifier	myunique
	Width	The object's width in CSS units	100%
	Height	The object's height in CSS units	200px
	Title	The tool-tip	Some info
	Visible	Is the object visible	true / false
	StyleName_ADD	Add a secondary style	mystyle
	StyleName_SET	Clear all styles and set to the given style	gwt-Button
	StyleDependentName	A suffix added to the primary style	gwt-TabBarItem-selected
	StylePrimaryName	The object's primary style	gwt-TabBarItem
	StackText	Stack heading when a widget is a child of StackPanel or DecoratedStackPanel	My Heading
	TabText	Tab heading when a widget is a child of TabPanel or DecoratedTabPanel	My Tab
	ColumnHeading	The Column Heading when used in a Grid	MyButtonHdg
AbsolutePanel	no extra		
Anchor	Direction	a direction that is right-to-left (RTL), left-to-right (LTR)	LTR
	Name	the name that will be associated with the widget when its form is submitted	myname
	Focus	can receive keyboard focus	true
	HorizontalAlignment	indicates that a widget can be aligned horizontally	ALIGN_CENTER
	HTML	the object's text can be set either as HTML or as text	<H1>Header</H1>
	Text	an object contains text	Hello
	WordWrap	indicates word-wrapping is enabled	true
	Href	the anchor's href (the url to which it links)	http://www.google.ca/
	Target	the frame in which navigation will occur when the link is selected	_blank
Button	Focus		
	HTML		
	Text		

## GWT Composite Builder

<b>GWTWidget</b>	<b>Property</b>	<b>Description</b>	<b>Sample</b>
CaptionPanel	CaptionText	the Text caption for the panel	
CheckBox	Focus		
	HTML		
	Name		
	Text		
	Checked	the CheckBox state	true
	Enabled	whether this widget is enabled	true
DateBox	no extra		
DecoratedStackPanel	no extra		
DecoratedTabPanel	no extra		
DecoratorPanel	no extra		
DisclosurePanel	HeaderText	the Text header for the panel	My Header
DockPanel	HorizontalAlignment	indicates that a widget can be aligned horizontally	ALIGN_CENTER
	VerticalAlignment	indicates that a widget can be aligned vertically	ALIGN_MIDDLE
	DockAlign_XXXX	Read only. Contains the LabWidget name at the location. Set when add a LabWidget to the DockPanel	myLabWidgetName
FileUpload	no extra		
FlexRow	no extra	Not a GWT widget. Used for structure only.	
FlexTable	BorderWidth	the width of the table's border, in pixels	1 (for 1px)
	CellPadding	the amount of padding to be added around all cells, in pixels	5 (for 5px)
	CellSpacing	the amount of spacing to be added around all cells, in pixels	2 (for 2px)
FlowPanel	no extra		
FocusPanel	no extra		
FormPanel	no extra		
Frame	Url	the URL of the frame's resource	<a href="http://www.google.ca/">http://www.google.ca/</a>
Grid	BorderWidth		
	CellPadding		
	CellSpacing		
	ColumnCount	the number of columns in this grid	8
	RowCount	the number of rows in this grid	10
Hidden	DefaultValue	the default value of the hidden field	mydefaultvalue
	Id	the id of the hidden field	uniqueid
	Value	the value of the hidden field	myvalue

## GWT Composite Builder

GWTWidget	Property	Description	Sample
HorizontalPanel	HorizontalAlignment		
	VerticalAlignment		
HorizontalSplitPanel	HorizontalSplit_LEFT	Read only. The LabWidget in the left side of the panel. Set when add a LabWidget to the Panel	myNavPanel
	HorizontalSplit_RIGHT	Read only. The LabWidget in the right side of the panel. Set when add a LabWidget to the Panel	myContentPanel
HTML	Direction		
	HorizontalAlignment		
	HTML		
	Text		
	WordWrap		
Image	Url		
InlineHTML	Direction		
	HorizontalAlignment		
	HTML		
	Text		
	WordWrap		
InlineLabel	Direction		
	HorizontalAlignment		
	Text		
	WordWrap		
Label	Direction		
	HorizontalAlignment		
	Text		
	WordWrap		
ListBox	Focus		
	Name		
	AddItem	The item name(s) to be added to the ListBox. Delimited by comma	apple, orange, pear
	SelectIndex	the currently selected index	0 (selects the first item)
	VisibleItemCount	the number of items that are visible. setting this value to 1 turns it into a drop-down list	3
MenuBar	AutoOpen	indicates whether this menu bar's child menus will open when the mouse is moved over it	true
	AnimationEnabled	indicates whether animations are Enabled	false
MenuBarV	same as MenuBar		
MenuItem	HTML		
	Text		
MenuItemSeparator	no extra		
PasswordTextBox	Direction		
	Focus		
	Name		
	Text		

## GWT Composite Builder

GWTWidget	Property	Description	Sample
	MaxLength		
	VisibleLength		
PushButton	Focus		
	HTML		
	Text		
RadioButton	Focus		
	HTML		
	Name		
	Text		
	Checked		
	Enabled		
RichTextArea	Focus		
	HTML		
	Text		
ScrollPanel	no extra		
SimpleCheckBox	Focus		
	Name		
	Checked		
	Enabled		
SimplePanel	no extra		
StackPanel	no extra		
SuggestBox	Focus		
	Text		
	Suggestion	The suggestion(s) to be added to the SuggestBox. Delimited by comma	car, cat, career, coat
TabPanel	no extra		
TextArea	Focus		
	Name		
	Text		
TextBox	Direction		
	Focus		
	Name		
	Text		
	MaxLength	the maximum allowable length of the text box	50
	VisibleLength	the number of visible characters in the text box	30
ToggleButton	Focus		
	HTML		
	Text		
Tree	Focus		
Treeltem	HTML		
	Text		
VerticalPanel	HorizontalAlignment		
	VerticalAlignment		
VerticalSplitPanel	VerticalSplit_TOP	Read only. The LabWidget in the top of the panel	myTopPanel
	VerticalSplit_BOTTOM	Read only. The LabWidget in the bottom of the panel	myBottomPanel