Echo2 Web Development Framework

Rakesh Vidyadharan
rakesh@sptci.com

May 14, 2007

Echo2

Echo2 is:

- Architecturally similar to AWT/Swing.

Echo2

Echo2 is:

- Architecturally similar to AWT/Swing.
- Event driven API.

Echo2

Echo2 is:

- Architecturally similar to AWT/Swing.
- Event driven API.
- Component based API.

Echo2

Echo2 is:

- Architecturally similar to AWT/Swing.
- Event driven API.
- Component based API.
- AJAX client-server interactions.

Echo2

Echo2 is:

- Architecturally similar to AWT/Swing.
- Event driven API.
- Component based API.
- AJAX client-server interactions.
- MPL/LGPL licence.

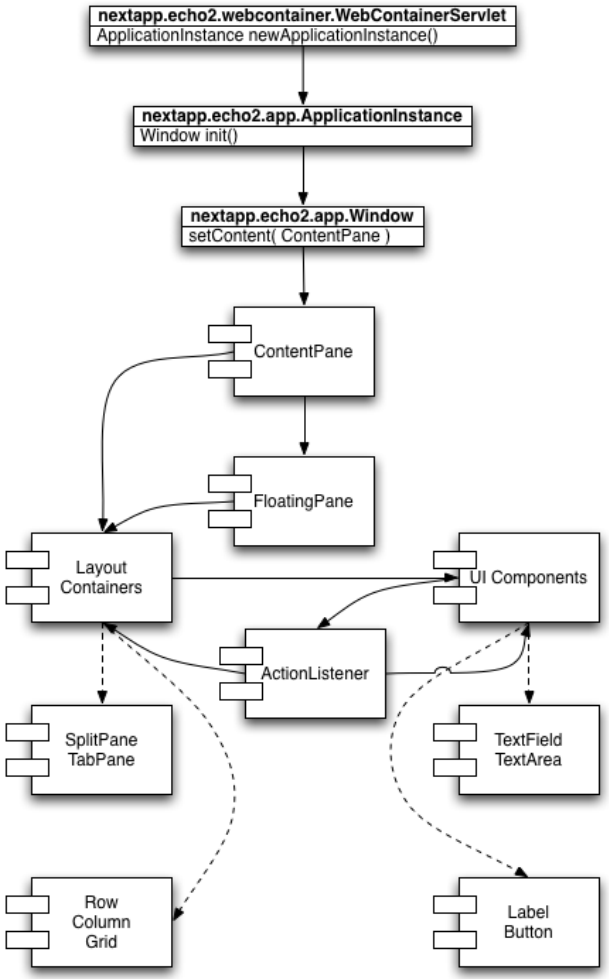
Echo2

Echo2 is:

- Architecturally similar to AWT/Swing.
- Event driven API.
- Component based API.
- AJAX client-server interactions.
- MPL/LGPL licence.

```
Button button = new Button( "Save" );
button.setActionCommand( "save" );
button.addActionListener(
  new ActionListener()
  {
    public void actionPerformed( ActionEvent e )
    { ... }
  });
```

Anatomy of Application



Application framework

Parts of an application.

Servlet A servlet instance that is used to initialise the ApplicationInstance.

Application framework

Parts of an application.

Servlet A servlet instance that is used to initialise the ApplicationInstance.

ApplicationInstance The application instance that is used to display a Window.

Application framework

Parts of an application.

Servlet A servlet instance that is used to initialise the ApplicationInstance.

ApplicationInstance The application instance that is used to display a Window.

ContentPane The ContentPane that is used to display the UI Components.

Application framework

Parts of an application.

Servlet A servlet instance that is used to initialise the ApplicationInstance.

ApplicationInstance The application instance that is used to display a Window.

ContentPane The ContentPane that is used to display the UI Components.

Components The Component objects that are used to build the UI.

Application framework

Parts of an application.

Servlet A servlet instance that is used to initialise the ApplicationInstance.

ApplicationInstance The application instance that is used to display a Window.

ContentPane The ContentPane that is used to display the UI Components.

Components The Component objects that are used to build the UI.

ActionListeners The event listeners that are triggered by user interaction with the UI Components displayed in the Window.

Layout Components

WindowPane A floating window component.

Layout Components

WindowPane A floating window component.

SplitPane Used to split the ContentPane into separate areas.

Layout Components

WindowPane A floating window component.

SplitPane Used to split the ContentPane into separate areas.

Column A container component used to hold child components stacked vertically.

Row A container component used to hold child components stacked horizontally.

Grid A container component used to hold child components in tabular format.

UI Components

Button Used to trigger action events similar to AWT/Swing.

Label Used to display text.

TextField/TextArea Used to display single or multi-line editable text. PasswordField is available to capture masked input.

SelectField/CheckBox/ListBox Used to display multiple selectable values to user. Backed by their own data model.

Table Used to display tabular data. Backed by data model.

Third Party Components

Some components provided by EchoPointNG:

MenuBar/Menu/MenuItem Used to build menu systems.

DateChooser/DateField Used to display calendar to user.

DirectHtml Used to display raw HTML.

RichTextArea Text area with HTML editor capability.

Tree Used to display objects in a tree.

AutolookupTextField A TextField that supports JavaScript and AJAX lookups to match text input into field.

Third Party Components

Some components provided by EchoPointNG:

MenuBar/Menu/MenuItem Used to build menu systems.

DateChooser/DateField Used to display calendar to user.

DirectHtml Used to display raw HTML.

RichTextArea Text area with HTML editor capability.

Tree Used to display objects in a tree.

AutolookupTextField A TextField that supports JavaScript and AJAX lookups to match text input into field.

SortableTable Enhanced Table implementation for displaying sortable tabular data.

PageableSortableTable Enhanced Table implementation for displaying pageable and sortable tabular data.

Third Party Components

Some components provided by EchoPointNG:

MenuBar/Menu/MenuItem Used to build menu systems.

DateChooser/DateField Used to display calendar to user.

DirectHtml Used to display raw HTML.

RichTextArea Text area with HTML editor capability.

Tree Used to display objects in a tree.

AutolookupTextField A TextField that supports JavaScript and AJAX lookups to match text input into field.

SortableTable Enhanced Table implementation for displaying sortable tabular data.

PageableSortableTable Enhanced Table implementation for displaying pageable and sortable tabular data.

Look & Feel

Styles are used to control look and feel for components:

Style Attributes

Color	Foreground and background color properties for components.
Font	Font types and sizes to be applied to components.
Border	Borders to be applied to components.
Size & Postition	Size and position of components.
Image	Fill images for component background.

Look & Feel

Echo2 offers three techniques for applying styles to components.

Style Techniques

Component Style properties can be directly specified for each component.

MutableStyle Custom objects used to capture style properties and applied directly to target component.

StyleSheet A class used to maintain a mapping of style names to style classes. Used to facilitate easy sharing of style objects. A custom StyleSheet loader that initialises itself from an XML file is also provided.

Component styling

```
Button button = new Button( "Save" );
button.setFont( new Font(
  Font.HELVETICA, Font.PLAIN,
  new Extent( 12 ) );
button.setForeground( Color.BLACK );
button.setTextAlignment( Alignment.ALIGN_CENTER );
```

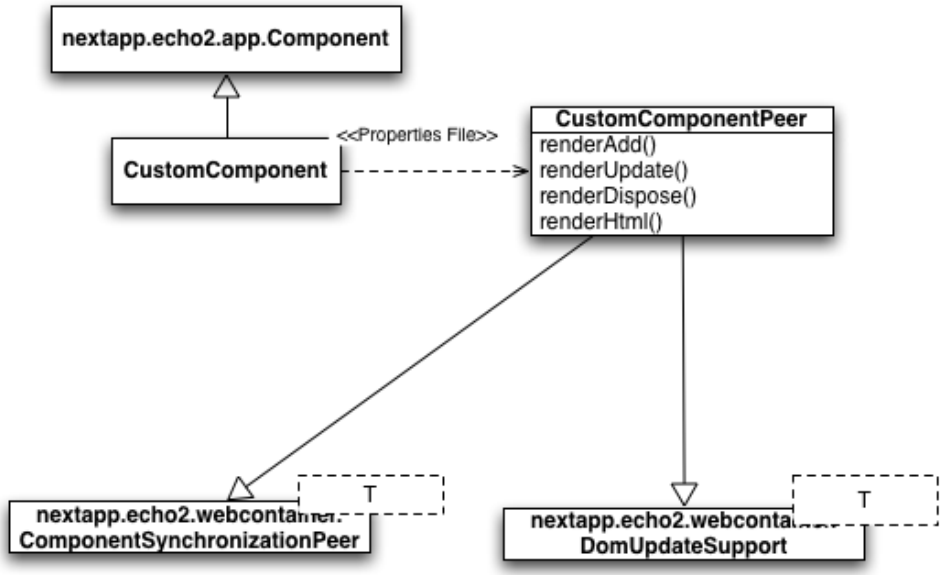
Style Class

```
Button button = new Button( "Save" );
button.setStyle( new ButtonStyle() );

public class ButtonStyle extends MutableStyle
{
   ...
   setProperty( Component.PROPERTY_FONT,
     new Font( Font.HELVETICA, Font.PLAIN,
     new Extent( 12 ) );
   setProperty(
     Component.PROPERTY_FOREGROUND, Color.BLACK );
   setProperty(
     Component.PROPERTY_ALIGNMENT, Alignment.CENTER
}
```

StyleSheet

```
Application.getActive().setStyleSheet(
  new StyleSheet() );
Button button = new Button( "Save" );
button.setStyleName( "General.Button" );

public class StyleSheet extends MutableStyleSheet
{
  ...
  ButtonStyle bs = new ButtonStyle();
  addStyle( Button.class, null, bs );
  addStyle( Button.class, "General.Button", bs );
}
```

Custom Components



CustomComponent Sub-class of Echo2 Component class.

CustomComponentPeer Rendering peer implementation.

CustomComponent.js Client side message processor to notify server of client interactions with components.

Sample Applications

SPTWebMail java.net project to provide webmail access.

Password Simple application to manage UNIX system account passwords.

SPTCMS Work in progress to create a simple Wiki/CMS.

PRS Performance Review System to be released on java.net

RAD A framework for Rapid Application Development through code generators, automatic view initialisation and automatic bi-direction binding.

Advantages

Advantages of using Echo2 and related extras:

API familiarity The API should be familiar to most Java developers.

Advantages

Advantages of using Echo2 and related extras:

API familiarity The API should be familiar to most Java developers.

Quick development Java developers can build dynamic web applications in very short time.

Advantages

Advantages of using Echo2 and related extras:

API familiarity The API should be familiar to most Java developers.

Quick development Java developers can build dynamic web applications in very short time.

Multiple interfaces Through the use of products such as BeanView one can automatically generate a Swing or Echo2 application around data model.

Advantages

Advantages of using Echo2 and related extras:

API familiarity The API should be familiar to most Java developers.

Quick development Java developers can build dynamic web applications in very short time.

Multiple interfaces Through the use of products such as BeanView one can automatically generate a Swing or Echo2 application around data model.

IDE plugin EchoStudio Eclipse plugin available for easy interface development.

Disadvantages

Disadvantages of using Echo2 and related extras:

Developers needed Applications can be maintained and updated only by developers.

Disadvantages

Disadvantages of using Echo2 and related extras:

Developers needed Applications can be maintained and updated only by developers.

Heavy runtime Echo2 server acts as a windowing server on the application server. Memory footprint for maintaining multiple user sessions is high.

Disadvantages

Disadvantages of using Echo2 and related extras:

Developers needed Applications can be maintained and updated only by developers.

Heavy runtime Echo2 server acts as a windowing server on the application server. Memory footprint for maintaining multiple user sessions is high.

Custom component The current model for developing custom components is a little involved.

Disadvantages

Disadvantages of using Echo2 and related extras:

Developers needed Applications can be maintained and updated only by developers.

Heavy runtime Echo2 server acts as a windowing server on the application server. Memory footprint for maintaining multiple user sessions is high.

Custom component The current model for developing custom components is a little involved.

Drag & Drop Rudimentary D&D support through extras library.

Disadvantages

Disadvantages of using Echo2 and related extras:

Developers needed Applications can be maintained and updated only by developers.

Heavy runtime Echo2 server acts as a windowing server on the application server. Memory footprint for maintaining multiple user sessions is high.

Custom component The current model for developing custom components is a little involved.

Drag & Drop Rudimentary D&D support through extras library.

Session required Each user gets a HTTP Session bound. There is no way to develop an application that does not require a session.

Disadvantages

Disadvantages of using Echo2 and related extras:

Developers needed Applications can be maintained and updated only by developers.

Heavy runtime Echo2 server acts as a windowing server on the application server. Memory footprint for maintaining multiple user sessions is high.

Custom component The current model for developing custom components is a little involved.

Drag & Drop Rudimentary D&D support through extras library.

Session required Each user gets a HTTP Session bound. There is no way to develop an application that does not require a session.

No deep linking You cannot bookmark or go directly to a screen within an application.

Echo2 vs GWT

Comparison with another Java based web development framework.

Feature	Echo2	GWT
Java API based	Yes	Yes
Java API	Full	Limited
Easy to modify JavaScript	No	Yes
AJAX abstraction	Yes	No
Server runtime	Heavy	Light
Client runtime	Light	Heavy
Development	Light	Heavy
Company	Small	Huge