CUSTOM SOFTWARE DEVELOPMENT

AGILE AJAX TOOLKIT COMPONENT FEATURES

WWW.OXAGILE.COM

TABLE OF CONTENTS

1 AGILE AJAX TOOLKIT OVERVIEW

Oxagile Java Agile Ajax Toolkit is a collection of components with beautiful UI and dynamic interface, which allows us to develop any Java web applications using these components.

This toolkit was thoroughly tested, contains no memory leaks, performance problems and is highly customizable. With the use of our Agile Ajax Toolkit components our professional Java developers fulfill any Ajax programming and rich internet application development tasks starting from dynamic website development and ending with complex RIA development and Ajax portal programming.

Agile Ajax Toolkit features:

- Pure GWT 2.0
- IE7,IE8,FIrefox,Chrome,Opera,Safari browser support
- RTL support
- Theming support
- No external Javascript Libraries
- No memory leaks
- Well designed component model
- Fully customizable
- High performance

[View Agile Ajax Toolkit Demo]

2 AGILE AJAX TOOLKIT COMPONENT DESCRIPTION

2.1 BUTTON

A button is a control, which is an interactive component that enables users to communicate with an application.

The border and the background of a button can change according to the state of the button. In the following illustration, the first button shows the default state, and the second button shows a changed border color, which indicates the button is in a focused state. A focused state occurs when a user moves a mouse pointer over the button. The final button shows that the border and background both change if the user clicks the mouse button when the mouse pointer is over the button.

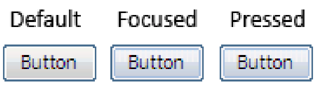


Figure 1. Buttons in default, focused, and pressed states

2.1.1 COMPONENT FEATURES

- Customizable view style;
- Advanced event handling model.

2.1.2 CREATING BUTTONS

When you create a button, you must pass button label as a constructor parameter:

```
Button button = new Button("Button");
```

2.1.3 HANDLING BUTTON EVENTS

You can add the handler using addClickHandler method of the Button class. Handler allows you to control the actions performed when the submit button is clicked.

```
Button button = new Button("Button");
button.addClickHandler(new ClickHandler() {
    public void onClick(ClickEvent event) {
   // Do something
    }
});
```

2.2 CHECKBOX

The CheckBox control enables users to select and clear options in an application's user interface. A CheckBox control can contain content such as text, images, or panels. The user's selection is indicated by a check mark, and when a user clicks a CheckBox, its appearance and state change. The following graphic shows two different states of a CheckBox, *checked* and *unchecked*.

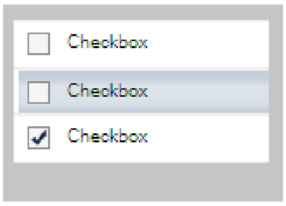


Figure 2. CheckBox in unchecked, focused and checked states

2.2.1 COMPONENT FEATURES

- Customizable view style;
- User can check/uncheck component by clicking component caption;
- Automatic component resizing accordingly parent component;
- Optional content. Text, images or other widgets.

2.2.2 CREATING CHECKBOXES

CheckBox component has three constructors, which allows you create object in different ways.

public CheckBox() – Creates a simple checkbox without caption.

public CheckBox(String text) – This constructor allows you to specify the caption of checkbox you create.

public CheckBox(String text, boolean checked) – Creates component with predefined caption and sets state of the created checkbox to checked.

2.3 RADIOBUTTON

The RadioButton is a control that is usually used as an item in a group of RadioButton controls. However, it is possible to create a single RadioButton. A RadioButton control can contain content such as text, images, or panels. RadioButton controls enable a user to change selections. Unlike a CheckBox, the user cannot clear the selected RadioButton by clicking it again. If the RadioButton is part of a group of RadioButton controls, the group contains functionality to manage toggling selections. The following graphic shows an example of a RadioButton control.

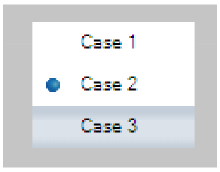


Figure 3. RadioButtons group. Buttons in unselected, selected and focused states

2.3.1 RADIOBUTTON GROUPS

When RadioButton elements are grouped, the buttons are mutually exclusive. A user can select only one item at a time within a RadioButton group. Unlike a CheckBox, when a RadioButton is selected it cannot be cleared by clicking it. The application programmatically clears a selected item when the user selects a new item. Whether a RadioButton is selected is determined by the state of its selected property. You can group RadioButton controls by placing them inside a parent and logically group RadioGroup components by adding them into RadioButtonGroup object.

2.3.2 COMPONENT FEATURES

- Customizable view style;
- User can check component by clicking component caption;
- Automatic component resizing accordingly parent component;
- Optional content. Text, images or other widgets.

2.4 MENU COMPONENT

The Menu control presents a list of items that specify commands or options for an application. Typically, clicking a menu item opens a submenu or causes an application to carry out a command.

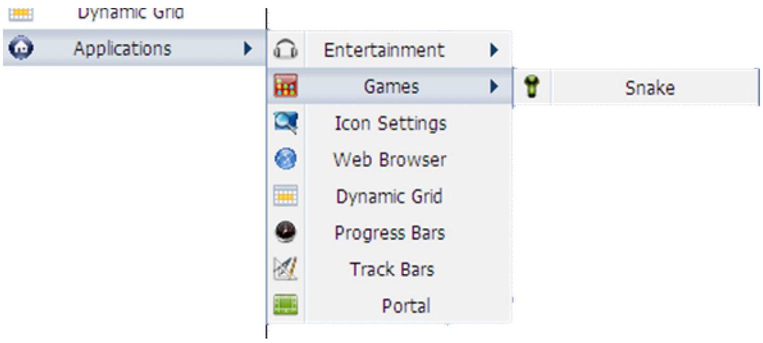


Figure 4. Menu sample

You can add any other widget as a menu item that allows you to create new types of menus, limited only by your imagination. You can also add a popup-menu to any widget. Using various combinations of widgets you can compose something, not conceding menus of desktop application on any modern operating system.

Figure 5. File menu in desktop application style

2.4.1 COMPONENT FEATURES

- Customizable view style;
- Optional content. Text, images or other widgets;
- Menu component can be assigned to any widget;
- Optional menu structure;
- Arbitrary level of nesting. You can create submenus anywhere you want.

2.5 HINT COMPONENT

In case you need to display some helpful information when user move mouse over control, you can use the hint component. This component watches for browser mouse events and displays specified content when mouse cursor moves over the control. You can fill hint component with anything you want. Text, images or complex content – all this can be displayed.

Figure 6. An example of simple hint

Hint component is smart enough to display himself in depend of screen position of parent component. For example, when you assigned hint to some component and that component is too close to right screen border, the hint will be displayed a way that does not go beyond the screen, even if default position of the hint window is bottom and right from mouse cursor. In particularly complicated cases you can create hint object from pure html code like

```
<font color="red">This </font>
<font color="green">is HTML</font>
<font color="blue"> hint</font>
```

Here we have the beautiful complex hint to demonstrate possibilities of our component.

Figure 7. Complex hint example

2.5.1 COMPONENT FEATURES

- Customizable view style;
- Optional content. Text, images or other widgets;
- Hint component can be assigned to any widget.

2.6 PROGRESSBAR

ProgressBar controls are used whenever an operation takes more than a short period of time. The ProgressBar indicates to users that the operation is happening and that the application is not hung.

Traditionally, progress bar include a text label, which describe component's or application's state. In some cases, label can display percent of process done or estimated time if process completion. The second ingredient of a classical progress bar is a color bar shows the ratio between completed part of the process and total workload.

The following graphic shows a different types ProgressBar component.

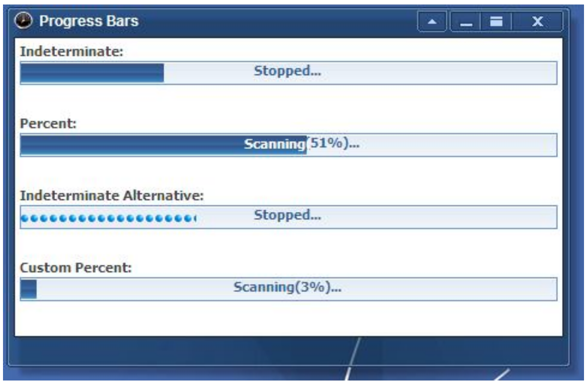


Figure 8. Various progress bars

As you can see on Fig. 8, ProgressBars components can be different. Depending of their purpose you can create indeterminate or percent progress bar with various representation.

Also you can choose alternative text and progress indication direction (from right to left) in case your application needs internationalization to the corresponding languages.

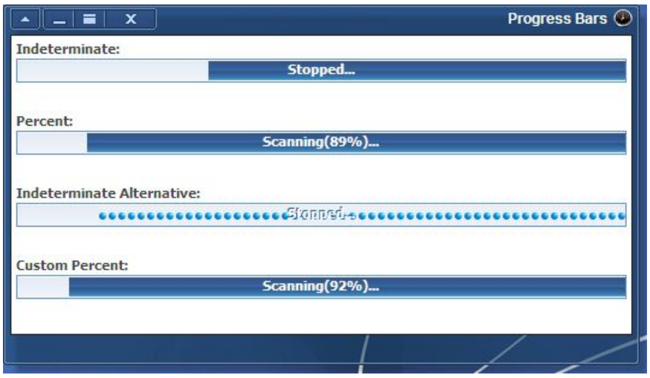


Figure 9. RTL orientation of the progress bars

2.6.1 COMPONENT FEATURES

- Customizable view style;
- Customizable progress indicator;
- Supporting of right to left writing languages.

2.7 TABBAR

Sometimes you get into situation when your application form full of information and controls. In this case you need to divide form into several certain forms containing fewer items. On the other hand all modern trends of desktop application development include the widespread use of different tabs. If you take a look on Microsoft Office, Internet Explorer, Mozilla Firefox or Google Chrome, you will see tabs in each of these products. Tabs are easy-to-use and can really increase your productivity when using application.

Sometimes images louder than text, so see yourself:

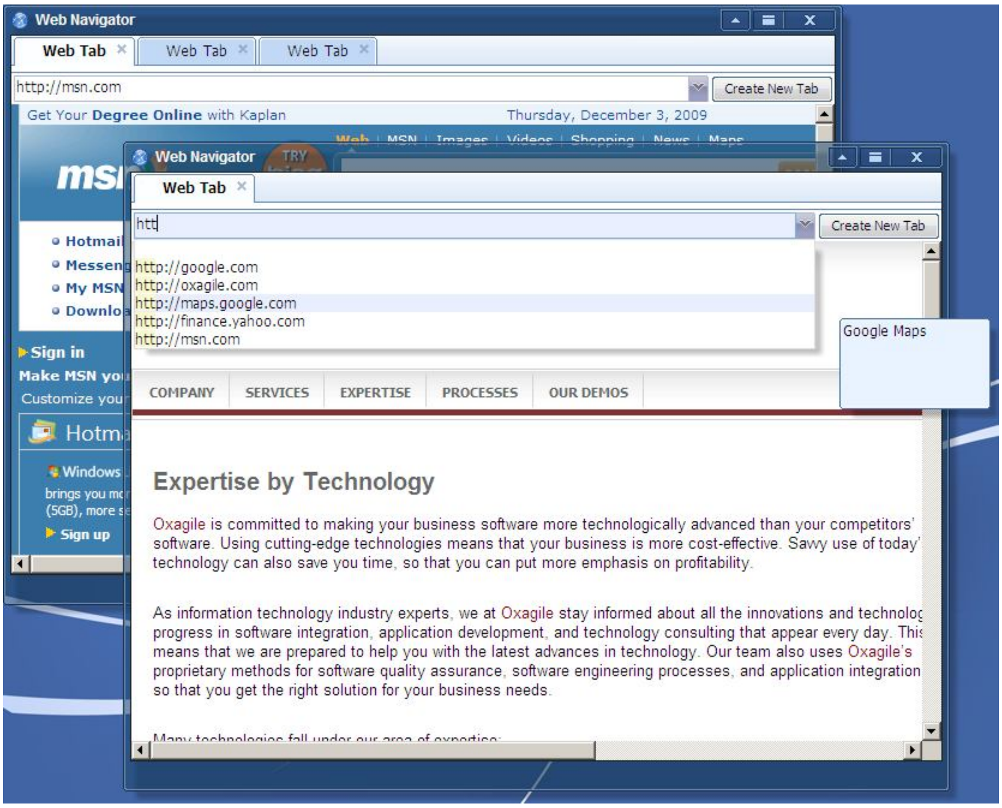


Figure 10. Tab component using in web browser Toolkit application

2.7.1 COMPONENT FEATURES

- Customizable view style;
- Automatic scroll items appears when tabs fills entire parent element width.

2.8 TRACKBAR

The TrackBar control (also sometimes called a "slider" control) is used for navigating through a large amount of information or for visually adjusting a numeric setting. The TrackBar control has two parts: the thumb, also known as a slider, and the scale on which the thumb moves. The thumb is the part that can be adjusted. Its position corresponds to the some value. The trackbar moves in increments that you specify and can be aligned horizontally or vertically. For example, you might use the track bar to control the current position media file playback or change volume level.

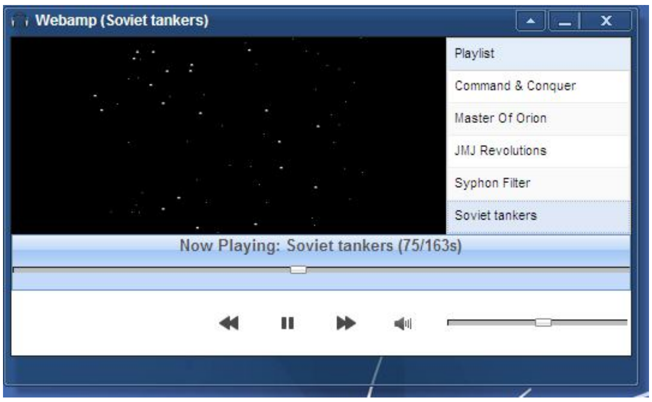


Figure 11 .Track bar controls in web-based media player

Also you can use track bars in navigation purposes: to navigate content not fitted in window or to specify placement of some interface object. Track bars can be horizontally or vertically oriented that helps you to create toolkit-based applications with rich functionality.

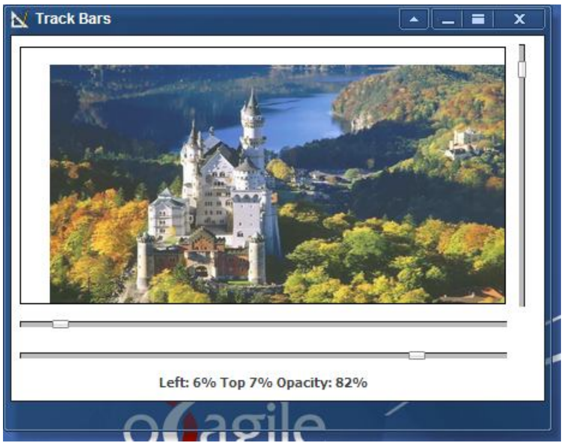


Figure 12. Track bars demo application

2.8.1 COMPONENT FEATURES

- Customizable view style;
- Horizontal and vertical element orientation;
- Smooth thumb movement.

2.9 SINGLE SELECTOR COMPONENT

The Single Selector Component is a user interface item that presents users with a list of options. The contents can be shown and hidden as the control expands and collapses. In its default state, the list is collapsed, displaying only one choice. The user clicks a button to see the complete list of options.

The following graphic shows the appearance of a selector component in different states.

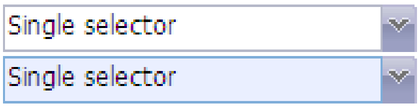


Figure 13. Single selector component in default and focused state

Selector component can filter assigned data. When data filtering you can see matches highlighted. Matching is doing by "include" clause. Component has great performance so you can process thousands of items in real-time.

Selector is based on true MVC architecture and can contain any type of objects in its low-level layer so you can select business object directly. Each item in single selector can contain optional hint component which will be displayed when mouse cursor positioned over item in drop-down list.

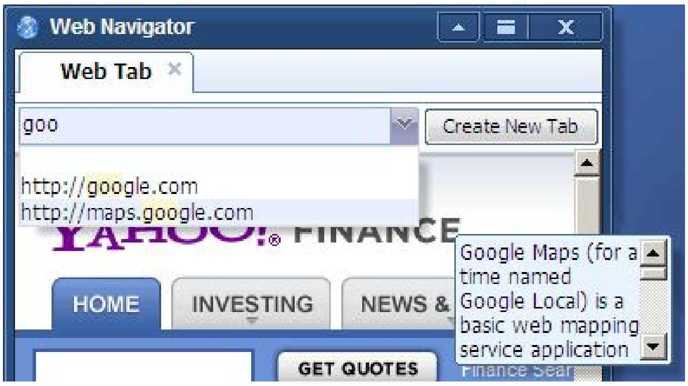


Figure 14. Selector in action

2.9.1 COMPONENT FEATURES

- Customizable view style;
- Filtering option;
- Great performance;
- Item hint element;
- True MVC model.

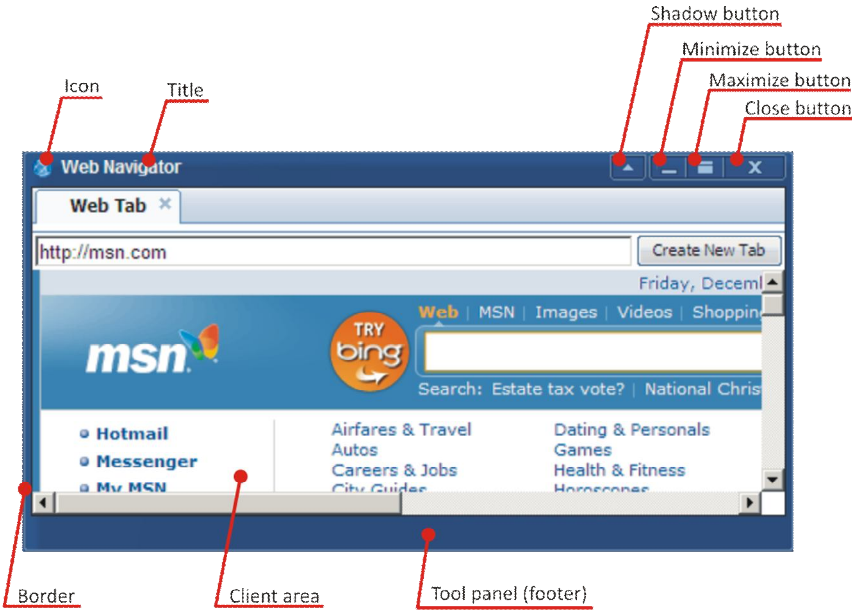
2.10 WINDOW COMPONENT

Window is one of fundamental components. In most cases your application will contain one or more windows. Windows are always rectangular. They are placed above and below each other along an imaginary line that runs perpendicular to the screen. Each window has a unique position in the z-order. Windows that appear first in the z-order are considered to be in front of, or on top of, windows that appear later in the z-order. A window's position in the z-order affects its appearance; a window might obscure another window partially or totally, depending on its location, size, and position in the z-order.

You can drag windows anywhere you want inside browser window. You can specify show or not window content during the drag process.

Windows can be resizable. In this case user can change window size using familiar way. Not resizable windows also known as dialog windows can be only closed or minimized.

Generally user can do with windows the same things he usual do with desktop windows. Toolkit Window component allows you to do one additional thing. You can “shadow” window, in this case window minimizes to its caption, freeing up desktop space for more important data.



As you can see, toolkit application window is fully analogical to window of the desktop application so your end-users do not need to re-accustom when they will migrate to web application.

As a nice extra you get additional are at the bottom of window which you can use at your discretion.

Windows supports transparency effect so you can see underlying things throw windows frames.

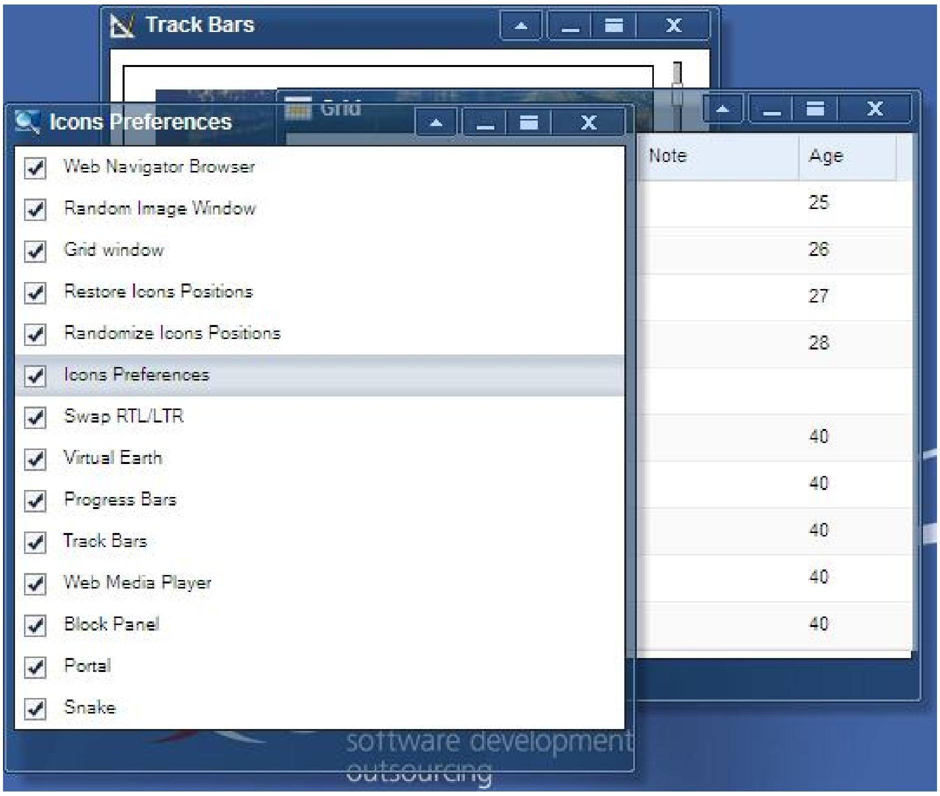


Figure 15. Windows examples

2.10.1 TIPS AND TRICKS

Of course you can use that windows in usual way, but if you want some additional functionality – here it is.

You can simply create custom controls in "non-client" area of the window. It can be any widget you want. For example, it possible to add button to the bottom window frame or some analog of Microsoft Windows Status Bar component, containing arbitrary controls.

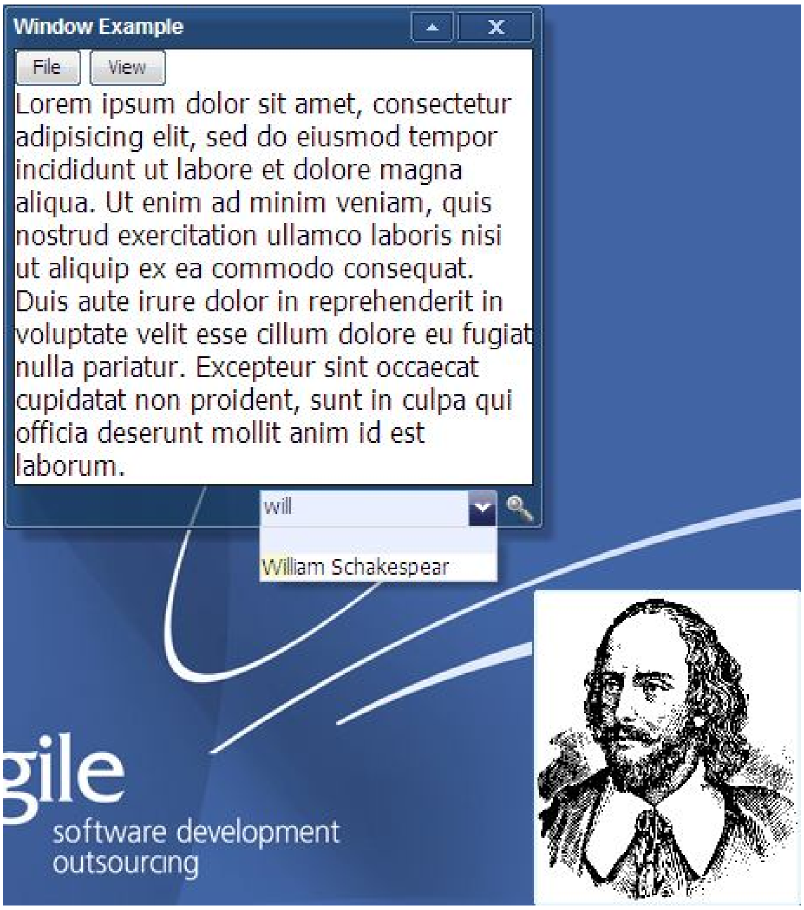


Figure 16. Window example with a tool in the bottom area

If you going to promote your application worldwide, you certainly need the support of right to left write languages.

Figure 17. Right-to-left window orientation

2.11 DRAG-N-DROP PORTAL

Drag portal is a new-generation way of widgets organization. This component is fully-customizable in run-time. You can configure general view of component during development time and allow it to freely evolve during execution time. You can add independent columns to organize top-level entities and fill that columns with custom widgets represents other business objects need to be grouped. After portal will be displayed user can drag separate widgets to any place he wants or swap entire columns. You can add simple widgets or complex, composite widgets. All components are moving smoothly.

You can allow or reject user to move some widgets from one place to another or only reject one defined target place. Also you can minimize widgets to save space on screen.

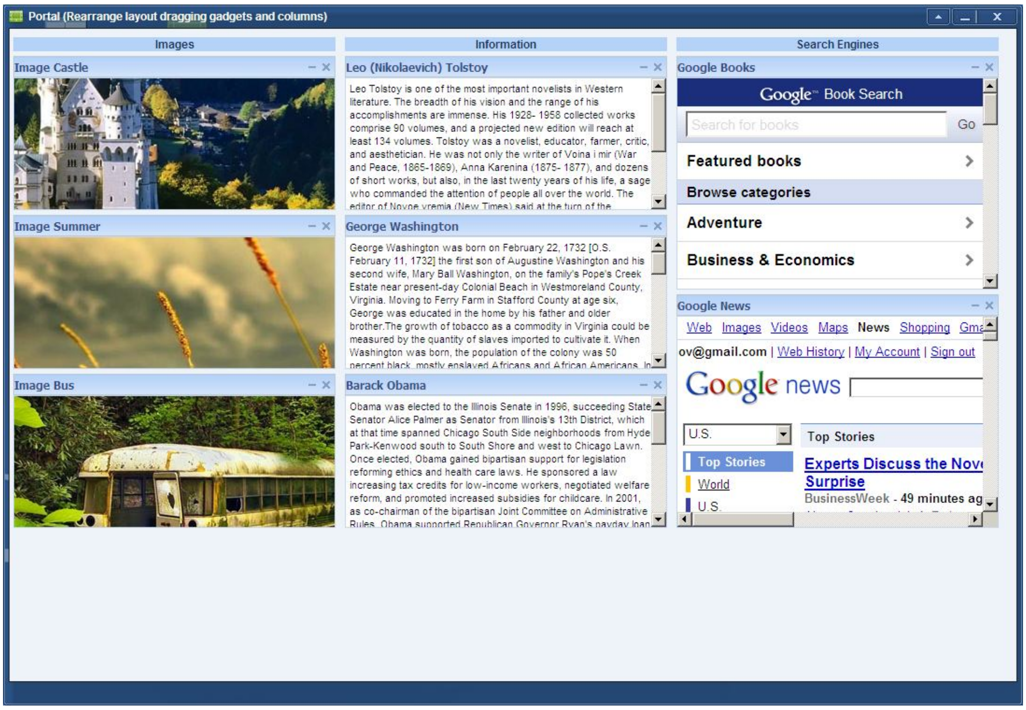


Figure 18. Drag portal window

When you move any element, portal dynamically adds placeholder for newly added element regardless of whether you drag one component or column entirely.

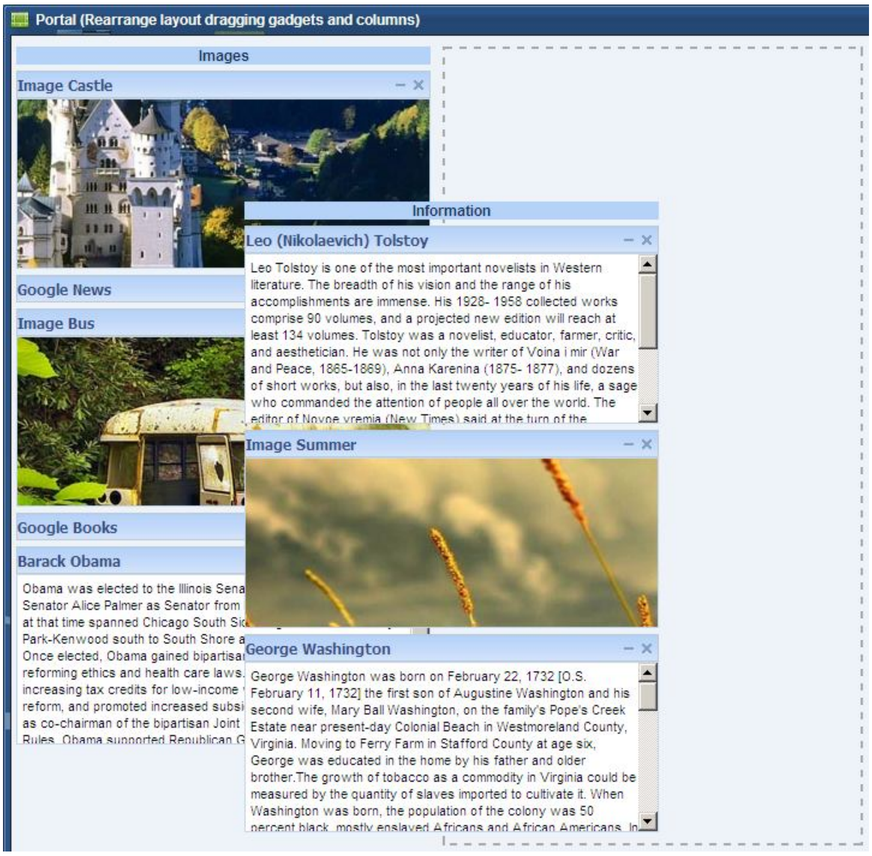


Figure 19. Moving whole column

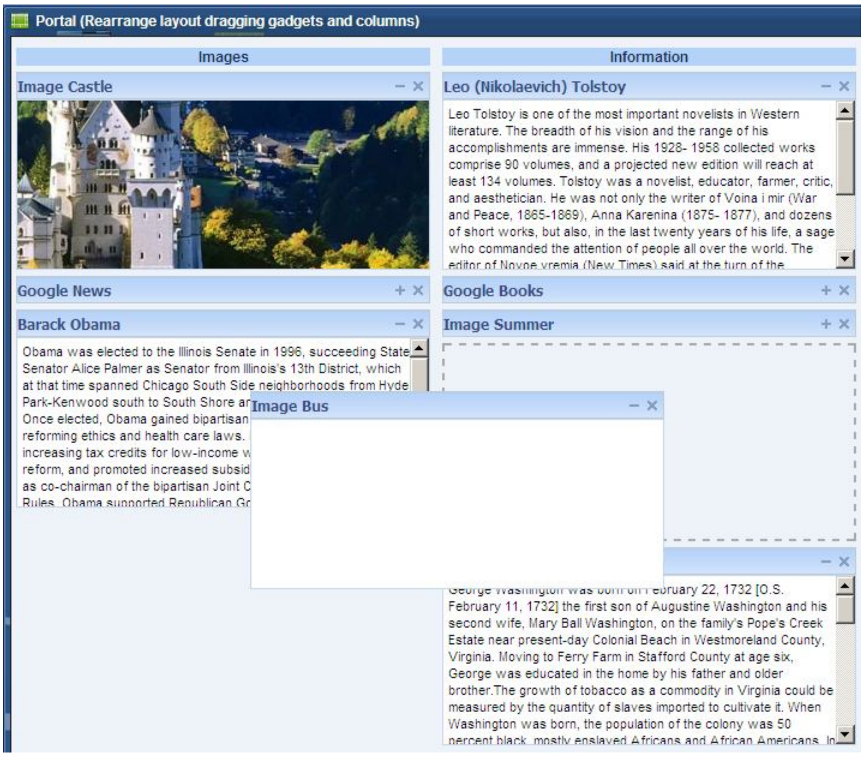


Figure 20. Moving single component

2.12 GRID COMPONENT

For table data representation we have a Grid component. With help of this component you can easily organize viewing of large amount of data or creating complicated custom forms.

Grids can contain any widgets in its cells and has an advanced event handling model. Any part of a grid can be an event source. You can add handlers for events generated by grid column headers or grid cells. You can drag entire columns from one place to another to reorganize their order or click on headers for change sort order of the grid data. Also you can specify minimum and maximum width for each column.

Figure 21. Grid sample

2.12.1 COMPONENT FEATURES

- Customizable view style;
- Resizable columns;
- Custom cell content;
- Moveable columns.

2.13 TASKBAR

Taskbar is kind of component that not usually used in web applications, but of course you can find it very useful when you encounter it in desktop operation systems like Microsoft Windows or Ubuntu. Task bar helps you to group and easily manage your running application. Since Toolkit applications closer to desktop applications, we thought that you will need something what can grant similar functions.

Figure 22. TaskBar component

As you can see on Fig. 21 TaskBar has wide set of controls inside.

Here we have similar to users "Start" button for access to the main menu of your system containing shortcuts to basic applications. Next to the start menu goes "quick launch panel" to provide user fast access to the most commonly used applications. Main part of the taskbar is a task area also known as task area. Here placed all thumbnails of applications now running. User can switch between applications by clicking on their corresponding representation in task area. Then goes buttons for changing whole taskbar position (top or bottom of the screen) and system clock. After clock widget we have another useful button which allows you to hide entire taskbar beyond the bounds of the screen or gain it back.