

Introduction to Adobe Flex 3

Guide to Navigation Containers

Trademarks

ChikaraDev and the ChikaraDev logo are trademarks of ChikaraDev. Such trademarks may be registered in the United States or in other jurisdictions, including internationally. This manual may include trademarks, service marks, or trade names of Adobe Systems Incorporated, Inc. and other companies. Such trademarks, service marks, or trade names may be registered in the United States or in other jurisdictions, including internationally.

Third-Party Information

This manual contains information such as links to third-party websites that are not under the control of ChikaraDev, and ChikaraDev is not responsible for the content on any linked site. If you access a third-party website mentioned in this manual, then you do so at your own risk. ChikaraDev has provided these links only as a convenience, and the inclusion of the link does not imply that ChikaraDev endorses or accepts any responsibility for the content on those third-party sites.

Copyright © 2008 ChikaraDev. All rights reserved.

The software described in this manual is provided under an agreement with Adobe Systems Incorporated, and such software can only be used in accordance with the terms of the agreement provided by Adobe Systems. Software code described and provided in this manual is provided under an agreement with ChikaraDev. The software can only be used in accordance with the terms of the agreement.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, photocopying, manual, optical, recording, or otherwise, outside the license agreement accompanying these materials, without the prior written permission of ChikaraDev. ChikaraDev claims copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of other rights of ChikaraDev and its subsidiaries.

Information in this manual may change without notice and does not represent a commitment on the part of ChikaraDev.

NOTICE OF LIABILITY

This information in these training materials is distributed on an “AS IS” basis, without warranty of any kind, either express or implied. While every precaution has been taken in the preparation of these materials, neither ChikaraDev nor its licensors shall have any liability to any person or entity with respect to liability, loss, or damage caused or alleged to be caused directly or indirectly by the instructions contained in these materials or by the computer software and hardware products described herein.

First Edition: July 2008 - ChikaraDev - Cupertino, CA 95014 USA

Overview of Navigator Containers in Adobe Flex

Navigator containers present “views” represented as children of those navigator containers. Only one of those views is visible at any given time. This can result in a more efficient use of your application “real estate”.

Moving among the multiple “views” of navigator containers is achieved using either a feature of the navigator container itself (TabNavigator and Accordion), or using a separate control.

Flex provides three navigator containers:

TabNavigator

Accordion

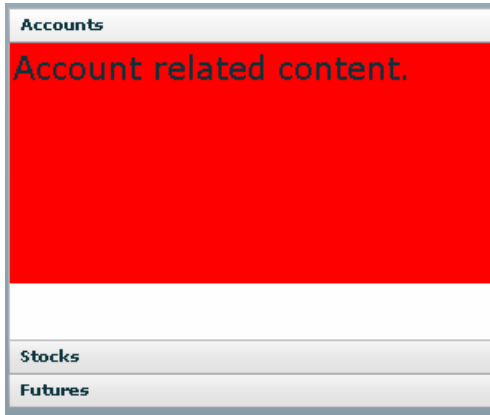
ViewStack

The TabNavigator controls navigation of its views using a set of tabs.



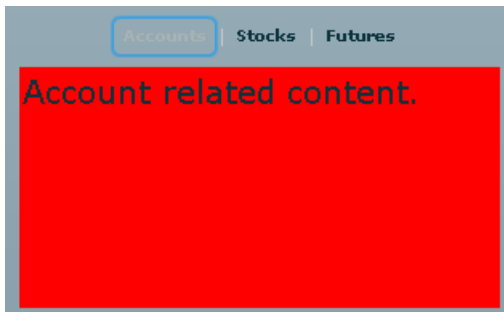
```
<?xml version="1.0"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:TabNavigator borderStyle="solid" >
    <mx:VBox label="Accounts" width="300" height="150" backgroundColor="0xFF0000">
      <mx:Label text="Account related content." fontSize="20"/>
    </mx:VBox>
    <mx:VBox label="Stocks" width="300" height="150" backgroundColor="0x00FF00">
      <mx:Label text="Stock related content." fontSize="20"/>
    </mx:VBox>
    <mx:VBox label="Futures" width="300" height="150" backgroundColor="0x0000FF">
      <mx:Label text="Futures related content." fontSize="20"/>
    </mx:VBox>
  </mx:TabNavigator>
</mx:Application>
```

The Accordion controls navigation of its views using header bars clicked to access each view.



```
<?xml version="1.0"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:Accordion id="accordion1" height="250">
    <mx:VBox label="Accounts" width="300" height="150" backgroundColor="0xFF0000">
      <mx:Label text="Account related content." fontSize="20"/>
    </mx:VBox>
    <mx:VBox label="Stocks" width="300" height="150" backgroundColor="0x00FF00">
      <mx:Label text="Stock related content." fontSize="20"/>
    </mx:VBox>
    <mx:VBox label="Futures" width="300" height="150" backgroundColor="0x0000FF">
      <mx:Label text="Futures related content." fontSize="20"/>
    </mx:VBox>
  </mx:Accordion>
</mx:Application>
```

The ViewStack can control navigation of its views using a LinkBar, ButtonBar, individual buttons, a menu, etc., and offers the most versatile option for controlling access to each view. The following example uses a LinkBar.



```

<?xml version="1.0"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:LinkBar dataProvider="{myViewStack}"/>
  <mx:ViewStack id="myViewStack" height="250">
    <mx:VBox label="Accounts" width="300" height="150" backgroundColor="0xFF0000">
      <mx:Label text="Account related content." fontSize="20"/>
    </mx:VBox>
    <mx:VBox label="Stocks" width="300" height="150" backgroundColor="0x00FF00">
      <mx:Label text="Stock related content." fontSize="20"/>
    </mx:VBox>
    <mx:VBox label="Futures" width="300" height="150" backgroundColor="0x0000FF">
      <mx:Label text="Futures related content." fontSize="20"/>
    </mx:VBox>
  </mx:ViewStack>
</mx:Application>

```

Note: The direct children of a navigator container must be containers, either layout or other navigator containers. You cannot directly nest a control within a navigator; controls must be children of a child container of the navigator container.

Navigator Container Text and Icons

The text displayed in the TabNavigator and Accordion navigator containers is controlled by the “label” property of its container children. An optional image displayed in the TabNavigator tabs and Accordion headers is controlled by the “icon” property and is not simply the path to an image, but the “class” for an embedded image, and here is one way to specify the icon image:

```

<mx:VBox label="Futures" width="300" height="150" backgroundColor="0x0000FF"
  icon="@Embed('blue30_30.jpg')">

```

The text displayed in the component controlling navigation among the views of the ViewStack container also comes from the “label” property of the ViewStack container children, but is specified by the dataProvider property of the controlling component, and the value of the dataProvider is the id of the ViewStack:

```

<mx:LinkBar dataProvider="{myViewStack}"/>

```

The ViewStack child container “icon” property can also be used to display an image in the controlling component.

Navigator Container creationPolicy Property

When a navigator container is initially displayed, the child container with an index of 0 appears, and by default none of the components in the other child views has been created yet. The components of those other navigator container views are created when that view is initially displayed due to user interaction or through ActionScript controlling the currently displayed view.

This can be a problem in situations where other areas of the UI reference components in non-initial views of navigator containers, or when they are referenced in ActionScript code. The result is a “null reference” error because the component being referenced does not yet exist.

The navigator container “**creationPolicy**” property, which by default is set to “**auto**”, can be explicitly set to “**all**” to force all components in all views of a navigator container to be created when the container is initialized, but this is considered indicative of bad design, as the creationPolicy property allows you to make efficient use of memory by only initializing components when they are needed. It can also affect performance, as it can slow down initialization as the components of multiple complex views are initialized.

If components in non-initial views must be referenced, they should be initialized manually, typically using global variables referenced in the non-initial view.