



Tutorial 2 - Styling a basic horizontal slider



Copyright

© 2009 SAP AG. All rights reserved. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary. These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

2009-11-03

Introduction

This tutorial assumes that you have at least some knowledge of Cascading Style Sheet (CSS), ActionScript 3.0, and MXML, the XML-based markup language introduced by Adobe Flex.

In tutorial 1, you created a basic horizontal slider using a Flex out-of-the-box component. With it, you can add styles and expose them through the property sheet in Xcelsius for the users to modify. Please have a basic horizontal slider created in tutorial 1 ready prior to starting this tutorial.

Styles are handled differently from the custom properties of the component. First, styles do not have get and set functions; `setStyle` and `getStyle` are used instead. Second, you will need to override `styleChanged` and `updateDisplayList` functions to detect changes to the styles and incorporate the new style into the component display. Finally, to include a default CSS definition for the component, I choose to implement `classConstruct` function to define a static initializer.

In this tutorial, for demonstration purposes, only the following styles are mentioned: "Show Track Highlight", "Data Tip Placement", "Title Font Color", "Title Font Family", and "Title Font Size". Please refer to the source code for the rest of the styles.

Note: if you want to create a custom style that is an array of colors, the best way is to create a custom property sheet to handle it. The default property sheet currently does not support styles that are of array type.

Create a basic Flex project and add a new AS component class. I named the class `BasicHorizontalSliderwithStyling`. Repeat the steps in Tutorial 1 to create another basic horizontal slider.

Task 1: Creating custom properties and styles

1. Add the following code above the class definition to define the styles with names, types and other information:

```
[Style (name="dataTipPlacement", defaultVal  
ue="top", enumeration="top,left,right,bottom")]  
[Style (name="showTrackHighlight", defaultVal
```