

```

        {
            return _target;
        }

        public function set target(value:SkinnableComponent):void
        {
            _target = value;

            if (_target.skin)
                _target.skin.addEventListener(FlexEvent.UPDATE_COMPLETE,
                    skin_updateCompleteHandler, false, 0, true);
        }

        override protected function updateDisplayList(unscaledWidth:Number,
            unscaledHeight:Number):void
        {
            this.setActualSize(target.width, target.height);
            super.updateDisplayList(target.width, target.height);
        }

        private function skin_updateCompleteHandler(event:Event):void
        {
            invalidateDisplayList();
        }

    ]]>
</fx:Script>

<s:Rect id="focusRect" top="-1" left="-1" right="-1" bottom="-1">
    <s:fill>
        <s:SolidColor id="bgFill" color="0xC5C551" />
    </s:fill>
</s:Rect>
</s:Group>

```

Setting focus when a window appears

Whenever a new screen opens in a Flex application, developers should ensure that focus is set on the first interactive element or the element that the user would use to first interact with the content. If this initial focus is not set, screen readers may announce nothing at all when the screen changes. Setting the focus on the correct element in the screen ensures that all content is read in the intended order from that location forward. Otherwise, the screen reader may *guess* where the focus is and the user may miss important content. In general, focus should be set to the component with the lowest `tabIndex` property that has `tabEnabled` set to `true`.

```

// when the title windows appears set focus to the first field
txtUserName.setFocus();

```

Return focus to open applications

When keyboard users leave a Flex application to use another application and then return, the focus should be where it was last left, specifically on the last focused element. By default, Flash Player resets focus to the element with the lowest `tabIndex` property value that is enabled every time the user returns to the Flex application. Assistive technology users will need to tab back to the component that had the focus each time they return to the application. If the Flex application is frequently used with other applications to copy, paste,