

# Sun Java Studio Creator 2

**Mikhail M. Matveev**

Senior Software Engineer  
Sun Microsystems, Inc.

<http://developer.sun.com/jscreator/>

# Goal of Talk

What You Will Gain From This Presentation

Learn about the features and capabilities of Sun Java Studio Creator 2 (Early Access)

# Agenda

## Introduction

Baseline Features

Components Features

Application Model Features

Data Access and DataProviders

Portlet Creation

Summary

# Introducing ...

- I am pleased to introduce you to the immediate availability of:
  - **Sun Java Studio Creator 2 (Early Access)**
  - <http://developer.sun.com/prodtech/javatools/jscreator/>

# Sun Java Studio Creator

A tool for the corporate developer

- Solid IDE platform and foundation – NetBeans
  - Provides infrastructure for building a complete development environment
  - Powerful “standard IDE fare”
- Built with and for the standard Java platform
  - Built in Java from the ground up
  - Built-in JDK 1.5
  - Integrates with Sun Java System Application Server Platform Edition 8.1 and PointBase DB
  - Portable applications for standards based servers
- Focus on visual development of web applications
  - JavaServer Faces technology

# Sun Java Studio Creator

As well as the enterprise developer

- Powerful IDE features:
  - Version control system integration
  - Refactoring
  - Debugging
- Flexible application model
  - Data binding to databases, EJBs, and web services
  - Access encapsulated business logic

# Sun Java Studio Creator

A target platform for component developers

- Component library import facilities:
  - Integrates components into palette
  - Optional source code and JavaDoc integration
  - Automatically included in project on first use
- APIs for defining *design time* behavior of components
  - Property editors and customizers
  - Event model for listening to related changes

# And how does it look?

# Agenda

Introduction

**Baseline Features**

Components Features

Application Model Features

Data Access and DataProviders

Portlet Creation

Summary

# Based on NetBeans 4.1

## Powerful Base Platform

- Common across Sun tools stack
- Common project directory format
- Powerful standard features
  - Version control system integration
  - Code refactoring
  - Multi-view windows
- <http://netbeans.org>

# Agenda

Introduction

Baseline Features

**Components Features**

Application Model Features

Data Access and DataProviders

Portlet Creation

Summary

# JavaServer Faces Technology

A framework for building user interfaces for web applications

- A set of APIs for representing UI components and managing their state
- Custom tag library for expressing a JavaServer Faces interface within a JSP page

## Benefits:

- Ease-of-Use
- Java community standardization
- Device Independence

# Comprehensive UI Component Library

The basic building blocks for web applications

- From the basics:
  - Label, Text Field, Button, Hyperlink, Image ...
- And popular requests:
  - Pop-up Calendar, File Upload, ...
- On to sophisticated layout:
  - Table, Tree, Tab Set, Breadcrumbs, ...
- Up to powerful composite components:
  - Add/Remove List, Orderable List, File Chooser ...

# Common Component Characteristics

## Components that work well together

- Component rendering is **theme sensitive**
  - Theme == Coordinated set of CSS stylesheets, JavaScript files, and icon images
    - Packaged as a single JAR file
  - Default theme and two variations included
  - Select theme at design time
- Components are aware of each other
  - Example: labels automatically display icons for invalid and/or required fields
- Complex behaviors assembled from simple child components

# Some examples

Introducing components  
Let's say "Hello World!"

# Component Library Import

Extending the set of available components

- Creator defines a *component import* archive format for importing component libraries
  - Runtime classes
  - (Optional) design time classes
  - (Optional) Javadocs
  - (Optional) source code (for code completion)
- If design time classes are not present, basic BeanInfo classes can be generated from metadata

# Design Time Behavior

## Improving the design time experience

- Component libraries may provide design time behavior for their components
  - Notification when component is created or deleted
  - Events for changes to other components
  - Customizers, property editors, context menu items, ...
- Design time API submitted to support JSR-273

# Visual Page Navigation Editor

## Set up navigation between pages

Editing Area: Page Navigation

**Page Navigator** lets you view the web pages of a project. By clicking and dragging between pages, you easily define navigation paths for users. Arrange pages according to your preferred work style. You can access Page Navigator from the Projects window or from the pop-up menu in the Editing Area.

# Virtual Forms – Creator 2 feature

Dealing with multiple input scenarios in a single page

- Consider a scenario where you have input and/or command components on both portions of a master-detail display
- You only want validation rules and updates fired for the portion actually submitted
- Options:
  - Multiple individual form components
    - Constrained to submit only child components
  - JavaServer Faces “immediate” property
  - Virtual forms
    - Feature of the Sun Web UI Components shipped with

# Virtual Forms

Dealing with multiple input scenarios in a single page

- A *virtual form* is a set of input components that participate in that form, and a set of input or action components that submit that form
- Input fields may participate in more than one virtual form
- Upon form submit, *only* the input components that participate in the submitted virtual form should be processed
- Configuration of virtual forms accessible programmatically

# Agenda

Introduction

Baseline Features

Components Features

**Application Model Features**

Portlet Creation

Summary

# Application Lifecycle Events

Reducing the complexity of application development

- JavaServer Faces application developers must typically be familiar with the standard *request processing lifecycle* for each request
- A simpler programming model would be if all coding were responses to events
  - Hollywood Principle: “don't call us, we'll call you”
- Runtime environment Creator provides creates exactly this approach
  - Uses only standard JavaServer Faces APIs, so portable to any runtime environment

# Application Lifecycle Events

Reducing the complexity of application development

- Creator associates a “page bean” with each JSP page representing a view
  - Components bound to concrete properties
  - User interface event handlers
- Creator 2 feature is support for *application* events:
  - `init()` -- When page bean is first created
  - `preprocess()` -- Before form submit is processed
  - `prerender()` -- Before current page is rendered
  - `destroy()` -- Clean up call (if `init()` was called)

# Agenda

Introduction

Baseline Features

Components Features

Application Model Features

**Data Access and DataProviders**

Portlet Creation

Summary

# Work with databases

## Creator's API for databases

- Based of JDBC rowsets
- Bundled PointBase DBMS
- Support of Oracle, SQL Server, Sybase, DB2 systems
- Support of other DBMS – JAR with driver is necessary
  - MySQL example
- Visual SQL editor

# Work with EJBs and web services

## Creator's API for EJBs and web services

- Easy interface of usage EJBs and web services
- Support of session EJBs
- Support of entity EJBs is planned
- Visual web service designer is planned

# Data Provider APIs

## Binding components to arbitrary data sources

- Initial version of Creator made binding to database tables easy
- Creator 2 Early Access introduces *Data Provider* APIs for binding to arbitrary data sources
- *DataProvider* represents access to an individual data object made up of *fields*:
  - JavaBean (field == property)
  - Map (field == key)
  - Row in a RowSet (field == column)

# Data access examples

# Agenda

Introduction

Baseline Features

Components Features

Application Model Features

Data Access and DataProviders

**Portlet Creation**

Summary

# New Application Project Type in Creator 2

## Graphically creating JSR-168 compatible Portlets

- Original version of Creator focused on visual creation of servlet based web applications
- Creator 2 Early Access adds the ability to create portlets compatible with the JSR-168 Specification
- Exactly the same design time experience
- Development testing done with a test portlet container inside the app server
- “Export WAR” for deployment to Portal Server

# Travel Center Sample

# Agenda

Introduction

Baseline Features

Components Features

Application Model Features

Data Access and DataProviders

Portlet Creation

**Summary**

# Summary

- Sun Java Studio Creator focuses on easy visual creation of web applications and portlets
- A new user interface component library makes building sophisticated applications easier
- New application model features make coding the server side application logic easier
- Final release of Sun Java Studio Creator 2 is planned to occur later this calendar year
- Sun Java Studio Creator 2 costs only \$99

# Waiting for Creator 3

- Sun Java Studio Creator 3 is planned to be released next year
- Broaden Client coverage (Richer and Mobile Clients)
- Improve developer productivity
- Integration with Java Studio Enterprise Tools

# Conclusion

- It's never been easier to develop Java web applications!

# Q&A

# Sun Java Studio Creator 2

**Mikhail L. Matveev**

Senior Software Engineer  
Sun Microsystems, Inc.

Special thanks to

**Craig R. McClanahan**

<http://developer.sun.com/jscreator/>

[java.sun.com/javaone/sf](http://java.sun.com/javaone/sf)

2005 JavaOne Conference

