Sun Java Studio Creator

Ken Paulsen
Staff Engineer
Sun Microsystems, Incorporated
(Slides by: Craig R. McClanahan)

Agenda

- Background
  - Developer characteristics
  - Corporate developers
  - Sun Java Studio Creator
- Application Model
  - Overview – Basic components
  - Under the covers – Additional components
  - Using data in applications
  - Page navigation

Agenda

- Other Features
  - Component Library Import
  - Creator Design Time APIs
- Summary

Agenda

- Background
  - Developer characteristics
  - Corporate developers
  - Sun Java Studio Creator
- Application Model
  - Overview – Basic components
  - Under the covers – Additional components
  - Using data in applications
  - Page navigation and value bindings

The Appeal of Java

- Java has traditionally appealed to technology-savvy developers
  - Object orientation
  - Type safety
  - Sophisticated, powerful APIs
- But it has often been perceived as “too hard to use” by a significant audience

Corporate Developers

- May not be your typical developers
  - May be domain experts
  - Might use VB, 4GLs, or scripting languages
- The applications they create:
  - Are typically <u>consumers</u> of data and services
  - Provide rich UI to what may be a complex underlying system

Corporate Developers

- The bar with most Java IDEs is just too high
  - Geared toward techologists / enterprise developers
    - Value = flexibility and richness
  - Corporate developers have different requirements
    - Value = simplicity and understandability
- Prefer a complete solution

Tools for Corporate Developers

Different development styles

- More “up front” coding
  - Design UI, modify attributes, plug in data, services, and logic
  - Heavy re-use and component drag and drop
- Simplified coding
  - Does not mean “no code”
  - Results should be understandable, obvious
  - “Truth in code”
  - Cause and effect should be evident

Tools for Corporate Developers

Different requirements

- Platform “power”
  - Must be available, but not required to be productive
  - Understanding complexity should not be required
- Resulting applications
  - Should be standards based
  - Minimal or no required runtime

Sun Java Studio Creator

A Tool For The Corporate Developer

- Solid IDE platform – NetBeans
  - Infrastructure for building a complete development environment
  - Powerful “standard IDE fare” features
- Built with and for Java standards
  - Integrated with SJSAS Platform Edition
  - Features standard Java APIs
  - Runtime applications are portable

DEMO

A Quick Orientation

Agenda

- Background
  - Developer characteristics
  - Corporate developers
  - Sun Java Studio Creator
- Application Model
  - Overview – Basic components
  - Under the covers – Additional components
  - Using data in applications
  - Page navigation and value bindings

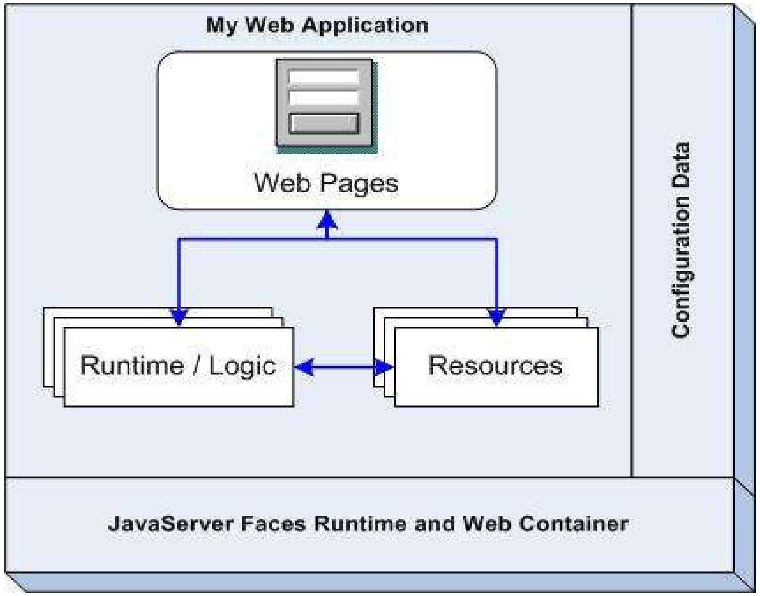
The Application Model

High Level View

- Applications consist of the following primary elements:
  - A set of web pages (and Java source)
  - Configuration (metadata) information
  - Additional logic, services, and runtime code
  - Extra resources as necessary (for example):
    - Images
    - Localizing bundles

The Application Model

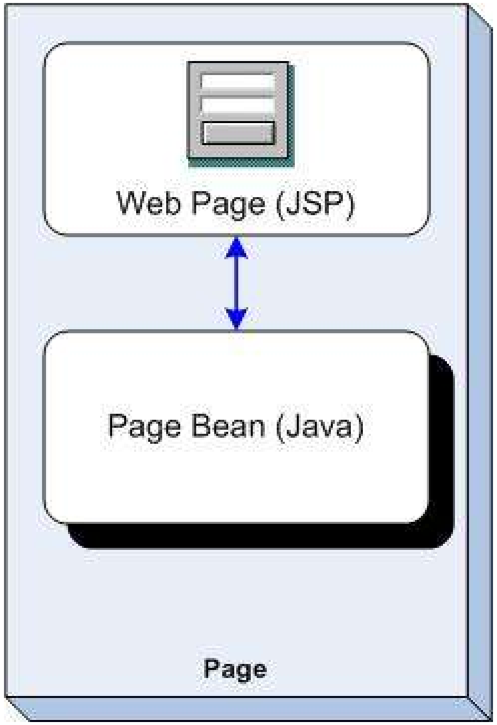
High Level View



The Application Model

Page Level View

- Each logical page consists of:
  - JSP Page
    - JSF components
    - Additional markup
  - Corresponding Page Bean
    - Component instances
    - Event handlers
    - Page properties
    - Methods



DEMO

JSP Pages and Page Beans

The Application Model

Under The Covers – Additional Elements

- Prebuilt JavaBeans:
  - PageBean – Per-request properties, components, event handlers
  - SessionBean – Per-user properties required across requests or pages
  - ApplicationBean – Application wide properties (such as cached contents for drop down lists)

The Application Model

Under The Covers – Additional Elements

- PageBean Additional Contents:
  - Event handlers (actions, value changes)
  - Converters
  - Validators
- Data Source Components:
  - JDBC Rowsets
  - Web Services
  - Enterprise JavaBeans

Using Data In Applications

JDBC Rowsets

- JDBC Rowsets are components:
  - Non-visual, so represented in the “tray”
  - Have properties that can be customized
  - Properties persisted in constructor of the containing page bean
- Visual binding of rowsets to UI components
- Visual editing of SELECT queries

DEMO

Binding Rowsets To UI Components #1

Rich Design Time Experience

- As we have seen in the demos so far, Creator provides rich design time facilities
  - WSYWIG rendering of components
  - Configuration of components on drop
  - Visual binding to properties or services
  - Context sensitive operations
- Let's see more of this in action

DEMO

Binding Rowsets To UI Components #2

Using Data In Applications

Web Services

- SOAP-Based Web Services are:
  - Defined by WSDL resource, which is ...
  - Located via a URL or a local file
- When you drop a web service onto a page in your application:
  - Creator generates client stubs
  - Adds appropriate libraries to your application
  - Provides Java API to access web service from event handlers

DEMO

Accessing Web Services (requires net access to deploy)

Using Data In Applications

Enterprise JavaBeans (Preview Feature)

- You may access stateless session beans that are available on your application server
- When you drop an EJB onto your page
  - Creator builds a client that encapsulates the EJB lookup and call
  - Adds appropriate libraries to your application
  - Provides Java API to access EJBs from event handlers

DEMO

Accessing Enterprise JavaBeans (Preview Feature)

Page Navigation

Easy Configuration for Multiple Page Applications

- JavaServer Faces supports two approaches to navigation:
  - Static – Predetermined path to second page
  - Dynamic – Calculate where to go next
- Navigation is based on:
  - Current page being submitted
  - Which submit button was pressed
  - Logical “outcome” (or static value) returned

Value Binding Expressions

Connecting components to data

- JavaServer Faces uses value binding expressions to connect components to underlying model data
  - #{customer.address.city}
- Based on expression language from JSTL 1.0 and JSP 2.0
- Extended to support dynamic bean creation via managed beans facility

Method Binding Expressions

Connecting components to event handlers

- Specialized expressions bind components to event handlers
  - #{Page1.submit_action}
  - public void submit_action() { ... }
- Allows all the event handlers for a page to be combined in one class
- Can be used for actions, validators, and value change listeners

DEMO

Using Page Navigation, Value Bindings, and Method Bindings

Agenda

- Other Features
  - Component Library Import
  - Creator Design Time APIs
- Summary

Component Library Import

Going Beyond The Standard Components

- Creator supports a mechanism to import third party libraries of JavaServer Faces components
- “Component Import” archive contains:
  - Runtime classes (to include in a webapp)
  - Design time classes (to be used by Creator)
  - Metadata (to register components with JSF)

Component Library Import

Going Beyond The Standard Components

- At import time,
  - Each library gets its own palette section
  - BeanInfo classes generated (if necessary)
- At design time,
  - Using third party components is the same drag-and-drop process as for standard components
  - Required runtime libraries added to webapp
- At run time,
  - JSF seamlessly manages all of the components

Creator Design Time APIs

Support for Third Party Components (Preview Feature)

- Creator supports a rich design time experience for standard components:
  - Event handlers for drop and link actions
  - Persists configuration in JSP and/or Java code
  - Custom property editors and customizers
  - Custom context menu items
- Internally, provided via “Creator Design Time APIs

Creator Design Time APIs

Support for Third Party Components (Preview Feature)

- We are publishing these APIs (as a Preview Feature) for use by third party component developers
- User's Guide for these APIs available on the Creator web site

DEMO

Component Import and Design Time APIs (Preview Feature)

Agenda

- Other Features
  - Component Library Import
  - Creator Design Time APIs
- Summary

Sun Java Studio Creator

A Tool For The Corporate Developer

- Rich design time user experience
- Drag-and-drop visual programming
- Persists all decisions in code:
  - JSP pages
  - Java sources
  - Configuration files
- Simple, intuitive application model

Sun Java Studio Creator

A Tool For The Corporate Developer

- Available as part of the first year of a Sun Developer Network subscription
  - Rich supporting web site content
  - Active user forum and support channels
  - Attractive price ($99/year list)
  - http://developer.sun.com/jscreator/
- Also available at Amazon with “Java Studio Creator Field Guide”

Sun Java Studio Creator

Ken Paulsen (ken.paulsen@sun.com)
Staff Engineer
Sun Microsystems, Incorporated
(Slides by: Craig R. McClanahan)