Data Binding and Java™ Platform, Micro Edition (Java™ ME)

Martin Brehovsky, David Kaspar, Michal Skvor

Software Developers
Sun Microsystems, Inc.

TS-5626

Session Goal

Data binding and Java Platform, Micro Edition (Java ME platform)

Learn how easy is to develop interactive applications connected to local or remote data by using NetBeans™ Data Binding Micro Edition (ME) library.

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Introduction

Data binding

- “Data Binding” is a general mechanism to bind data to user interface elements
- Abstraction provides uniform access to various data sources
- Allows:
  - Separation of data/model and view/controller
  - Cleaner design
  - Faster application development
- Many data binding frameworks available for Java Platform, Enterprise Edition (Java EE) and Java Platform, Standard Edition (Java SE), but not for Java ME platform

Introduction

Expression Language

- A mechanism to easily navigate complex data structures
- Parsed and evaluated at runtime—expressions can be changed without recompiling application sources
- Originally appeared as a part of JavaServer Pages™ (JSP™) 2.0 specification (Java Specification Request (JSR) 154)
- Starting to appear in other frameworks—for example Beans Binding (JSR 295)

Introduction

Expression Language—The syntax

- Simple and rich syntax
  - Variables and properties access
  - Arrays and maps access
  - Function calls
  - Arithmetic and compare operations
- Supported types
  - Java platform primitive types
  - Number, BigInteger, …
  - String, Comparable, Object

Introduction

Expression Language—Samples

- Variable and properties access
  ```
  ${user.lastname + " " + user.firstname}
  ```
- Method calls
  ```
  ${sms:sendMessage(user.phoneNumber,"Hello")}
  ```
- Comparison
  ```
  ${user.age >= 21 ? "beer" : "soda" }
  ```
- Array and map access
  ```
  ${users[0 + 1].lastname}, ${users["nickname"].lastname}
  ```
- Dynamic evaluation
  ```
  a=#{5}, b=${a*a}, a=#{10} // eval(b) is 100
  ```
- Static evaluation
  ```
  a=#{5}, b=#{a*a}, a=#{10} // eval(b) is 25
  ```

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Data Binding and Java ME Platform

Dealing with Java ME platform

- Limited hardware resources
  - Limited size of memory available to application
  - Limited download size of the application package
  - Limited processing power
- No reflection or component model
- Necessity to work with existing UI components, but allow integration with other frameworks

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

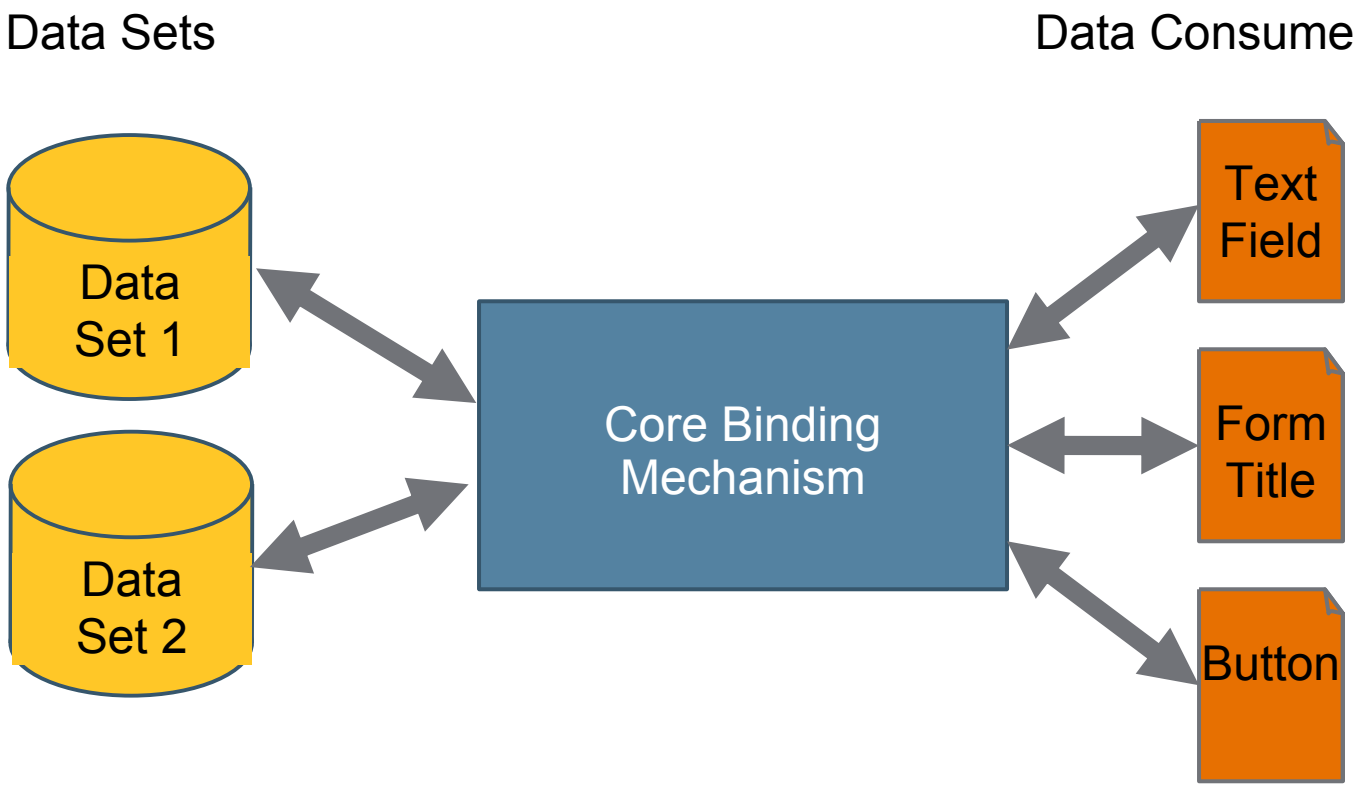
NetBeans Data Binding ME Library

Overview

- Data binding framework for Java ME platform
  - Supports CLDC, might be used for CDC and Java SE platform
- Open source project
  - Hosted on http://databindingme.dev.java.net/
- Uses Expression Language-like syntax as the binding “glue”
- Focuses on simplicity
  - Only a single line of code necessary to do the bind
- Provides API for creating data part
- Provides API and implementation for UI part

NetBeans Data Binding ME Library

Block scheme



NetBeans Data Binding ME Library

The architecture

- Three logic blocks
- Core Binding Mechanism
  - **Expression Language parser**
  - Binding and updating support mechanisms
- Data sets
  - APIs to be implemented by boundable data, e.g., PIM
- Data consumers
  - APIs to be implemented by consumers, e.g., LCDUI bindings

NetBeans Data Binding ME Library

Examples

```
// register data set
DataSet ds = new MyRemoteDatabaseDS("db.netbeans.org",
                                    "username", "password");
DataBinder.registerDataSet(ds ,"database");

// one-time value access
String username =
(String)DataBinder.readValue("database.username");

// create consumer
StringItem item = new StringItem("Username", null);

// bind the data set with consumer
DataBinder.bind("database.users[index].name",
new StringItemBindingProvider(), item,
StringItemBidningProvider.FEATURE_TEXT);
```

NetBeans Data Binding ME Library

Delivering changes in data sets to consumers

- Framework supports automatic listening on changes in data sets
  - No extra code required
  - EL defines a place in data structure (not just the value)
  - DataBinder listens on changes of data sets
  - DataBinder notifies DataBindingProvider
  - DataBindingProvider updates the consumer
- Two-way bindings
  - Consumer can push data back to data set e.g., updating an input value from TextField in connected data set

DEMO

Data Binding in Action

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Binding to Data/Services

What makes data a data source

- Data or services can be either local or remote
- Implementation needs to implement interface `org.netbeans.microedition.databinding.DataSet`
- Important methods are:
  - `getValue(name)`
  - `setValue(name,value)`
- When a value is changed, call: `DataBinder.fireDataSetChanged(dataSet, dataItemName)`
- For hierarchical data use DataSet as a value
- For indexable data use IndexableDataSet

DEMO

Binding to Local and Remote Data

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Binding UI Components

Extensible API for data consumers

- Data consumer API is extensible
- Framework provides implementation for LCDUI
- Can be extended to other UI frameworks
- Can be extended even to non-UI frameworks

Binding UI Components

Example—Providing bindings to SVG graphics

- SVG graphics is a XML-based vector graphics format
- DOM APIs can be used to manipulate the content
- Changes to the document automatically appear on the screen
- Data binding is done using element IDs
- Any attribute/text node can be bound to data set

DEMO

Binding to SVG Graphics

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Tools Support

Beyond the NetBeans Data Binding ME Library

- NetBeans Mobility Pack includes the library
- End-to-end tools can be used to automatically generate source code for data sets
- Support for web services using JSR 172
- Support for any remote data using Mobile Web Client

DEMO

Connecting to a Remote Service Using NetBeans Mobility Pack

Future Plans

- Extend the library to offer implementations of data set providers for various local services, e.g., PIM or RMS
- Visual support for using data bindings in Mobility Pack's Visual Mobile Designer

Agenda

Introduction

Data Binding and Java ME Platform

NetBeans Data Binding ME Library

Binding to Data/Services

Binding UI Components

Tools Support

Summary and Q&A

Summary

- NetBeans Data Binding ME Library provides a generic data binding facility to Java ME platform
- Data binding can be used to connect to both local and remote data and services
- Data binding can be used with LCDUI as well as other frameworks
- NetBeans Mobility Pack comes with tools support for generating data binding-enabled code

For More Information

- Links
  - http://databindingme.dev.java.net
  - http://mobility.netbeans.org
  - http://www.netbeans.org
- Other 2007 JavaOne^{SM} Conference Sessions
  - TS-9452—Visual Development and Deployment of Advanced Mobile Applications
  - TS-5628—Developing Flashy Mobile Applications, Using SVG and JSR 226
- Look for us in tools area and mobility area of the Sun booth

Q&A

Martin Brehovsky, David Kaspar, Michal Skvor

Data Binding and Java™ Platform, Micro Edition (Java™ ME)

Martin Brehovsky, David Kaspar, Michal Skvor

Software Developers
Sun Microsystems, Inc.

TS-5626