

# Glossary of Terms

## Table of contents

1 Glossary: Controls.....	2
2 Glossary: NetUI.....	2

## 1. Glossary: Controls

### **Control**

Controls are designed to make it easier to integrate complex resources into your Java application. Controls consist of two Java files: a Control implementation file, and a Control interface file.

### **Control Bean**

A Control Bean is a file that is generated while running the Controls annotation processing. The Control Bean is the glue between a control interface and a control implementation. When using a Control from a Control client, either the Control's interface or the Control Bean may be used to refer to declare a Control. The Control Bean provides access to methods supporting adding / removing event listeners and getting / setting Control properties.

### **System Control**

System controls are those that are low-level abstractions atop typical system resources such as EJB, JMS, JDBC, and web services.

## 2. Glossary: NetUI

### **page flow**

A page flow consists of one [controller](#) class and one or more pages, all associated with a single directory path. A web project can contain many page flows.

### **page flow controller**

A controller class is a Java class that defines actions, exception handlers, state, etc. related to a [page flow](#). Configuration information is defined through annotations on methods, fields, and the class itself.

### **shared flow**

A shared flow is a class that defines actions, exception handlers and state which can be shared by [page flow controllers](#).

### **data binding**

Data binding used to bind UI widgets to data in the web-tier environment. The data objects can be located in various places throughout the web application:

- in the [page flow controller class](#)
- in container provided JSP implicit objects
- in NetUI framework-provided JSP implicit objects

More details about data binding can be found [here](#).

### **form bean**

Typically, a form bean is a server-side representation of the data in an HTML <form> tag. A form bean follows ordinary Java Bean syntax: each form bean is a class consisting of any number of properties, each property having a setter and a getter method associated with it. Form bean classes normally implement `java.io.Serializable` in order to facilitate persistence of form bean instances. The sample form bean below has two properties; each one has a getter and setter method associated with it.

```
public class ProfileFormBean
    implements java.io.Serializable {

    private int age;
    private String name;

    public void setAge(int age) {
        this.age = age;
    }

    public int getAge() {
        return this.age;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }
}
```

Form beans are most often used to pick up data that is submitted from a <netui:form> tag. Once the POST data has been mapped to a form bean instance, the form bean is passed to a Page Flow action method. Note that form beans can be either standalone Java files or inner classes of a [page flow controller](#).

### NetUI JSP Tag Library

The JSP library for NetUI web applications. The <netui> tag library uses JSP 2.0 and NetUI databinding expressions to bind UI to data provided by the [page flow controller](#) and other web application data/resources.

Java, J2EE, and JCP are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

© 2004, Apache Software Foundation