

# Struts Spring Plugin

## 1. Overview

This project was originally created to integrate the [Spring Framework's](#) Inversion of Control (IoC) into [Struts 1.1+](#). As of Spring's 1.0.1 release, the [ContextLoaderPlugin](#) was added to Spring that has this same functionality. Therefore, this project's plugin is no longer recommended for use. We continue to maintain this site for documentation and an example app.

The integration uses Spring to create and populate Struts actions, using IoC to resolve dependencies. It does **NOT** use a custom request processor and therefore can be more easily used with existing web applications and/or other Struts extensions.

To demonstrate the project, we modified the struts-example webapp that is distributed with Struts 1.1 to use Spring to resolve all UserDatabase dependencies automatically. Comments and suggestions are appreciated.

## 2. Features

- Intuitive Inversion of Control (IoC)
- Requires little or no direct references to Spring in the Struts webapp
- Can be used with other Struts extensions that use a custom RequestProcessor
- Can create new Actions for every request (alleviating need for thread-safe Actions)
- Includes modified Struts example

## 3. What's New

### 3.1. Discontinued - April 12, 2004

- This plugin is now part of Spring (as of 1.0.1). You can read more about Spring's Struts support on [Matt Raible's Spring Live blog](#).

### 3.2. 0.1 - October 17, 2003

- Initial release

## 4. Requirements

This plugin requires Struts 1.1 or greater.

## 5. Usage

To use the Struts Spring plugin, add the `ContextLoaderPlugIn` to your Struts config file (usually `struts-config.xml`):

```
<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
  <set-property property="contextConfigLocation"
    value="/WEB-INF/applicationContext.xml"/>
</plug-in>
```

The "contextConfigLocation" property is the location of the Spring beans configuration file.

For each action that uses Spring, you need to define the action mapping to use `org.springframework.web.struts.DelegatingActionProxy` and declare a matching (action "path" == bean "name") Spring bean for the actual Struts action. This is an example of an action that requires an instance of `UserDatabase`:

```
<action path="/logon"
  type="org.springframework.web.struts.DelegatingActionProxy">
  <forward name="success" path="/logon.jsp"/>
</action>
```

The corresponding Spring bean configuration:

```
<bean id="userDatabase" class="org.apache.struts.webapp.example.memory.MemoryUserDa

<bean name="/logon" class="org.apache.struts.webapp.example.LogonAction">
  <property name="userDatabase"><ref bean="userDatabase" /></property>
</bean>
```

For more information on the Spring configuration file format, see the [Spring beans DTD](#).

The Struts action `org.apache.struts.webapp.example.LogonAction` will automatically receive a reference to `UserDatabase` without any work on its part or references to Spring by adding a standard JavaBean setter:

```
private UserDatabase database = null;

public void setUserDatabase(UserDatabase database) {
  this.database = database;
}
```

## 6. Contact

Please contact the [Spring Framework User List](#) with comments and suggestions.