

Upgrade Guide

Covers upgrading from Spring 2.5 to 3.0

The logo for Spring, featuring the word "Spring" in a large, black, serif font. A small, stylized green leaf with a brown stem is positioned above the letter 'i'.

Java Application Framework



2010-04-22

Copyright © 2010 SpringSource, a Division of VMware Inc.

Copies of this document may be made for your own use and for distribution to others, provided that you do not charge any fee for such copies and further provided that each copy contains this Copyright Notice, whether distributed in print or electronically.

Table of Contents

1. Overview of New Features	1
2. Requirements and Supported Environments	2
2.1. Java 5 and Java 6	2
2.2. Java EE Support	2
2.3. Compatibility with Other Spring Projects	3
2.4. Third-Party Libraries	3
2.5. Packaging	3
2.6. Required Spring Framework jar Files	4
2.7. Repositories	6
Obtaining Spring Releases From Maven Central	7
Obtaining Spring Releases From the Enterprise Bundle Repository (EBR)	9
Spring Project Productivity Tools	12
3. Removed and Deprecated Features	13
3.1. Removed Features	13
3.2. Deprecated Java Features	14
3.3. Deprecated XSD Schema Features	14
4. Application Migration and Compatibility Concerns	16
4.1. Changing Spring 2.5.6 Generics to Match 3.0 Generics	16
4.2. Implementing Remote Support Between Spring 2.5.6 Server Applications and 3.0 Client Applications	16
4.3. Deprecated Elements and Effects on Schema	17
4.4. Avoiding Naming Conflicts with Spring Bean conversionService	17
A. Code Changes Between Spring 2.5.6 and 3.0.0	18
A.1. API Change Codes	18
A.2. API Change Report	20
aop.framework	20
aop.interceptor	20
aop.support	21
aop.target	21
beans	21
beans.factory	22
beans.support	26
context	26
context.annotation	26
context.event	26
context.support	27
core	27
core.style	28
core.task	28
core.type	29

instrument.classloading	30
jca.endpoint	30
jca.work	30
jdbc.core	30
jdbc.support	32
jms.listener	32
jmx.export	34
jmx.support	34
metadata	34
metadata.commons	34
mock.web	34
orm.hibernate3	35
orm.ibatis	35
orm.jdo	35
orm.toplink	36
remoting.jaxrpc	38
remoting.jaxws	38
remoting.rmi	38
remoting.support	39
scheduling	39
scheduling.backportconcurrent	39
scheduling.concurrent	39
scheduling.quartz	40
scheduling.support	40
scheduling.timer	40
test.annotation	41
test.context	41
test.web	41
transaction	42
transaction.annotation	42
transaction.config	43
transaction.interceptor	43
transaction.jta	43
transaction.support	44
ui	44
util	45
validation	45
web.bind	45
web.context	46
web.multipart	47
web.portlet	48
web.servlet	51
web.util	54

1. Overview of New Features

Spring Framework 3.0 incorporates significant changes. This guide describes changes to the Spring Framework and issues you might encounter when you upgrade to Spring Framework 3.0.

The most notable change is the upgrade to Java 5 or later. This upgrade does alter the API by introducing generics and varargs, but Spring 3.0 provides backwards compatibility to the extent possible.

Spring 3.0 incorporates consistent use of generic Collections and Maps, consistent use of generic FactoryBeans, and consistent resolution of bridge methods in the Spring AOP API. Generic ApplicationListeners automatically receive specific event types only. All callback interfaces such as TransactionCallback and HibernateCallback now declare a generic result value. Overall, the Spring core code base is now freshly revised and optimized for Java 5.

Spring's TaskExecutor abstraction is updated for close integration with Java 5's `java.util.concurrent` facilities. We provide first-class support for Callables and Futures now, as well as ExecutorService adapters, ThreadFactory integration, and so on. This support is aligned with JSR-236 (Concurrency Utilities for Java EE 6) as much as possible. Spring 3.0 supports asynchronous method invocations through the new `@Async` annotation (or EJB 3.1's `@Asynchronous` annotation). There is also a new `@Scheduled` annotation that marks a method to be scheduled.

The new Spring expression language (SpEL) allows you to use references to nested bean structures (for example, properties of other beans) as well as to environmental data structures, such as system property values, through a common `#{...}` syntax in property values.

See [Spring Framework Reference](#) and the rest of this guide for information about more new features that you may want to use after you upgrade to Spring 3.0.

2. Requirements and Supported Environments

This chapter covers the build and runtime requirements for Spring Framework 3.0.

2.1 Java 5 and Java 6

The entire code base is updated to take advantage of Java 5 code style and features (generics, varargs, StringBuilder, and so on.)

Any application that uses Spring Framework 3.0 must use Java 5 or later. Spring requires Java 6 for building the code base of the Framework but Java 5 is sufficient for building and running applications that use Spring 3.0.

2.2 Java EE Support

Compatible with J2EE 1.4 and higher

Spring 3.0 maintains compatibility with J2EE 1.4 and higher but drops support for application servers that require Java 1.4 to run.

The following application servers are compatible with Spring 3.0:

- WebLogic Server 9.0, 9.2 and 10 and Oracle WebLogic Server 11g (10.3.2)
- WebSphere 6.1 (6.1.0.9 or later recommended) and 7
- Tomcat 5.0, 5.5 and 6.0
- SpringSource tc Server 6.0.19A and later
- SpringSource dm Server 2.0
- JBoss 4.2, 5.0 and 5.1
- Glassfish v1, v2 and v3
- Resin 3.0 and 3.1
- Jetty 5.1 and 6.1

2.3 Compatibility with Other Spring Projects

Other Spring projects have been tested and updated (as necessary) to work well with Spring 3.0.

Most Spring Projects depend on the new Spring 3.0 release for their new versions. Each project has also tested the new Spring 3.0 with the existing version and the following table shows the minimum version of the most commonly used projects that are compatible with Spring 3.0.

Table 2.1. Spring Projects and Spring 3.0 compatibility

Project	Versions
Spring Security	2.0.5 and 3.0
Spring Web Flow	2.0.6 and 3.0
Spring Web Services	1.5.8 and 2.0
Spring Batch	2.0.4 and 2.1

2.4 Third-Party Libraries

Spring is compatible with many third-party libraries; for example:

- JSF 1.1, 1.2 , and 2.0
- Tiles 2.1.2 and later
- Hibernate 3.2 or later
- Eclipse Link 1.0.1 or 2.0 and later
- OpenJPA 1.1 and later

You must upgrade from Tiles 2.0.x to 2.1.x because of internal API changes in the Tiles code. Tiles 2.0.x is not supported in Spring 3.0.

2.5 Packaging

You must use the individual module jars now because `spring.jar` is no longer provided.

The revised Spring Framework modules are now managed separately with one source-tree per module jar:

- `org.springframework.aop`
- `org.springframework.beans`
- `org.springframework.context`
- `org.springframework.context.support`
- `org.springframework.expression`
- `org.springframework.instrument`
- `org.springframework.jdbc`
- `org.springframework.jms`
- `org.springframework.orm`
- `org.springframework.oxm`
- `org.springframework.test`
- `org.springframework.transaction`
- `org.springframework.web`
- `org.springframework.web.portlet`
- `org.springframework.web.servlet`
- `org.springframework.web.struts`

2.6 Required Spring Framework jar Files

Because you no longer can use the "spring.jar" file, you must specify the individual module jars for your application.

For a typical web application you need the following module jars:

- org.springframework.web.servlet
- org.springframework.web
- org.springframework.asm
- org.springframework.beans
- org.springframework.core
- org.springframework.context
- org.springframework.expression

Since most web applications use logging and basic AOP features, you need the following required third-party offerings:

- commons-logging-1.1.1
- aopalliance-1.0 (only if you use org.springframework.aop)

The aopalliance-1.0 jar file was included in "spring.jar" but when you use the individual modules it needs to be included explicitly.

If you need database access using your preferred ORM tool, you also need these module jars:

- org.springframework.orm
- org.springframework.jdbc
- org.springframework.tx
- org.springframework.aop

You also need any dependencies used by your ORM tool of choice and by your JDBC driver.

Adjust any servlet/web dependencies to the versions needed for your servlet container:

- servlet-api-2.4
- jsp-api-2.1
- jstl-1.2

2.7 Repositories

If you use Maven, choose one of two repositories from which to obtain the Spring jars:

- Maven Central, which is the default repository Maven queries, and does not require special configuration
- Enterprise Bundle Repository (EBR), which is run by SpringSource and also hosts all the libraries that integrate with Spring

In general, if you care about OSGi, use the EBR, because it houses OSGi-compatible artifacts for all of Spring's dependencies, such as Hibernate and Freemarker. If OSGi does not matter to you, either option works. Choose one place or the other for your project; do not mix them. This is particularly important because EBR artifacts use a different naming convention from Maven Central artifacts.

The following table compares Maven Central to the EBR:

Table 2.2.

Feature	Maven Central	Enterprise Bundle Repository (EBR)
OSGi-compatible	No.	Yes.
Number of artifacts	Tens of thousands; all kinds.	Hundreds; those that Spring integrates and supports.
Consistent naming conventions for all artifacts?	No.	Yes.
Artifact naming convention	Group id: Varies. Newer artifacts use domain name, for example, org.slf4j; older artifacts use artifact id, for example, log4j. Artifact id: Varies. Typically the JAR file name minus extension, for example, log4j. Version: Varies. Most use numbers and periods, for example, 3.0.0.	Group id: <Domain name> such as org.springframework. Artifact id: <Bundle-SymbolicName>, derived from main package, for example, org.springframework.beans. If the JAR had to be patched to ensure OSGi compliance, com.springsource. is prepended, for example, com.springsource.org.apache.log4j. Version: OSGi version number format of <major>.<minor>.<micro>[.qualifier], for example, 3.0.0.RELEASE.
Publishing	Automatic (rSync via remote repositories).	Manual (JIRA processed by SpringSource).
Quality assurance	Accuracy is responsibility of publishing	Extensive (for both MANIFEST.mf and

	organization.	.pom); QA is performed by Spring Team.
Hosting	At Contegix funded by Sonatype with several mirrors.	Amazon S3 funded by SpringSource.
Search utilities	Various	http://www.springsource.com/repository
Integrated with SpringSource Tools (STS, Roo, CloudFoundry)	Yes, with STS.	Yes, with STS, Roo, and CloudFoundry.

Obtaining Spring Releases From Maven Central

You do not have to add a repository to your .pom to obtain final releases of Spring projects from Maven Central. Simply add the dependencies your project requires. A .pom <dependency> snippet for each Spring Framework 3 artifact as it will be indexed in Maven Central is listed below.

```

<!--
  Core utilities used by other modules.
  Define this if you use Spring Utility APIs
  (org.springframework.core.* / org.springframework.util.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Expression Language (depends on spring-core)
  Define this if you use Spring Expression APIs (org.springframework.expression.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-expression</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Bean Factory and JavaBeans utilities (depends on spring-core)
  Define this if you use Spring Bean APIs (org.springframework.beans.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Aspect Oriented Programming (AOP) Framework (depends on spring-core, spring-beans)
  Define this if you use Spring AOP APIs (org.springframework.aop.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-aop</artifactId>
  <version>3.0.0.RELEASE</version>

```

```
</dependency>

<!--
  Application Context (depends on spring-core, spring-expression, spring-aop, spring-beans)
  This is the central artifact for Spring's Dependency Injection Container and is generally
  always defined
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Various Application Context utilities, including EhCache, JavaMail, Quartz, and
  Freemarker integration
  Define this if you need any of these integrations
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Transaction Management Abstraction (depends on spring-core, spring-beans, spring-aop,
  spring-context)
  Define this if you use Spring Transactions or DAO Exception Hierarchy
  (org.springframework.transaction.* / org.springframework.dao.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  JDBC Data Access Library (depends on spring-core, spring-beans, spring-context, spring-tx)
  Define this if you use Spring's JdbcTemplate API (org.springframework.jdbc.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Object-to-Relation-Mapping (ORM) integration with Hibernate, JPA, and iBatis.
  (depends on spring-core, spring-beans, spring-context, spring-tx)
  Define this if you need ORM (org.springframework.orm.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Object-to-XML Mapping (OXM) abstraction and integration with JAXB, JibX, Castor, XStream,
  and XML Beans. (depends on spring-core, spring-beans, spring-context)
  Define this if you need OXM (org.springframework.oxm.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-oxm</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>
```

```

<!--
  Web application development utilities applicable to both Servlet and Portlet Environments
  (depends on spring-core, spring-beans, spring-context)
  Define this if you use Spring MVC, or wish to use Struts, JSF, or another web framework
  with Spring (org.springframework.web.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Spring MVC for Servlet Environments (depends on spring-core, spring-beans, spring-context,
  spring-web)
  Define this if you use Spring MVC with a Servlet Container such as Apache Tomcat
  (org.springframework.web.servlet.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Spring MVC for Portlet Environments (depends on spring-core, spring-beans, spring-context,
  spring-web)
  Define this if you use Spring MVC with a Portlet Container
  (org.springframework.web.portlet.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc-portlet</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

```

Obtaining Spring Releases From the Enterprise Bundle Repository (EBR)

To obtain final releases of Spring projects from the EBR, add the following repositories to your .pom:

```

<repository>
  <id>com.springsource.repository.bundles.release</id>
  <name>EBR Spring Release Repository</name>
  <url>http://repository.springsource.com/maven/bundles/release </url>
</repository>
<repository>
  <id>com.springsource.repository.bundles.external</id>
  <name>EBR External Release Repository</name>
  <url>http://repository.springsource.com/maven/bundles/external </url>
</repository>

```

Then simply add the dependencies your project requires, keeping in mind the EBR artifact naming conventions.

A .pom <dependency> snippet for each Spring Framework 3 artifact as it will be indexed in the EBR is listed below:

```

<!--
  Core utilities used by other modules.

```

```
    Define this if you use Spring Utility APIs
    (org.springframework.core.* / org.springframework.util.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.core</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Expression Language (depends on core)
  Define this if you use Spring Expression APIs (org.springframework.expression.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.expression</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Bean Factory and JavaBeans utilities (depends on core)
  Define this if you use Spring Bean APIs (org.springframework.beans.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.beans</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Aspect Oriented Programming (AOP) Framework (depends on core, beans)
  Define this if you use Spring AOP APIs (org.springframework.aop.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.aop</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Application Context (depends on core, expression, aop, beans)
  This is the central artifact for Spring's Dependency Injection Container and is
  generally always defined
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.context</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Various Application Context utilities, including EhCache, JavaMail, Quartz, and
  Freemarker integration
  Define this if you need any of these integrations
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.context.support</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Transaction Management Abstraction (depends on core, beans, aop, context)
  Define this if you use Spring Transactions or DAO Exception Hierarchy
  (org.springframework.transaction.* / org.springframework.dao.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
```

```
<artifactId>org.springframework.transaction</artifactId>
<version>3.0.0.RELEASE</version>
</dependency>

<!--
  JDBC Data Access Library (depends on core, beans, context, transaction)
  Define this if you use Spring's JdbcTemplate API (org.springframework.jdbc.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.jdbc</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Object-to-Relation-Mapping (ORM) integration with Hibernate, JPA, and iBatis.
  (depends on core, beans, context, transaction)
  Define this if you need ORM (org.springframework.orm.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.orm</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Object-to-XML Mapping (OXM) abstraction and integration with JAXB, JiBX, Castor,
  XStream, and XML Beans. (depends on core, beans, context)
  Define this if you need OXM (org.springframework.oxm.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.oxm</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Web app development utilities common across Servlet/Portlet environments
  (depends on core, beans, context)
  Define this if you use Spring MVC, or wish to use Struts, JSF, or another web
  framework with Spring (org.springframework.web.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.web</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Spring MVC for Servlet Environments (depends on core, beans, context, web)
  Define this if you use Spring MVC with a Servlet Container such as Apache Tomcat
  (org.springframework.web.servlet.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.web.servlet</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>

<!--
  Spring MVC for Portlet Environments (depends on core, beans, context, web)
  Define this if you use Spring MVC with a Portlet Container
  (org.springframework.web.portlet.*)
-->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>org.springframework.web.portlet</artifactId>
  <version>3.0.0.RELEASE</version>
</dependency>
```

```
</dependency>
```

Spring Project Productivity Tools

Spring tools can be useful for projects that use Maven. Both the SpringSource Tool Suite and Spring Roo provide wizards that can generate new Spring projects with pre-configured .poms. Roo can actually manage your .pom for you as you execute code generation commands that require additional artifacts to be downloaded. Cloud Foundry also has a new capability that allows cloud deployments to be made without external dependencies, greatly reducing deployment times. To make this work, Cloud Foundry syncs with the EBR after publishing to complete a deployment.

3. Removed and Deprecated Features

To maintain a lean Spring Framework, we have removed outdated features and deprecated others in favor of more up-to-date alternatives.

3.1 Removed Features

The following features and classes are no longer used by the vast majority of Spring users and have been removed:

- WebLogic 8.1 and WebSphere 5.1 support

Any application server that requires Java 1.4 cannot support applications that use Spring Framework 3.0. For Java 1.4 environments, we recommend the continued use of Spring 2.5.x.

- Native TopLink API support (superseded by JPA)

Support for TopLink native APIs has been removed from Spring 3.0. The support provided in Spring Framework 2.5.x is for TopLink 9 and 10. The newer TopLink 11 has a modified API, which Spring does not support. If you use TopLink 9 or 10 and want to use Spring 3.0, we recommend that you migrate your code to TopLink or Eclipse Link JPA.

- Commons Attributes support (superseded by Java 5 annotations)

The Commons Attributes support made sense before Java 5 and its annotation support. This was a long time ago and if you haven't already migrated your code to use Java 5 annotations, now is the time to do so.

- Jakarta JSTL support for JSP 1.2 (superseded by JSP 2.0)

Apache Tomcat version 4.1 and some other application servers based on J2EE 1.3 used older Servlet and JSP specifications that are no longer supported. Tomcat 5.0 and later as well as all other application servers supported by Spring (see Section 2.2, “Java EE Support”) use Servlet Specification 2.4 and JSP Specification 2.0 or later. If you are using one of these supported servers all you need to do is have your application include the correct version of the JSP JSTL API jar (version 1.1.2 for JSP 2.0 and version 1.2.0 for JSP 2.1). If you are using an older application server you would need to upgrade to a more recent version and also upgrade your web application to use a newer Servlet and JSP specification.

- ContextLoaderServlet and Log4jConfigServlet (use the corresponding Listener instead)

The application servers that required the use of the ContextLoaderServlet and Log4jConfigServlet (there were some issues with the order the listeners were called on startup) are no longer supported and there is no need to use these startup servlets any more. The ContextLoaderListener and

Log4jConfigListener provide the same functionality.

3.2 Deprecated Java Features

Some features have been deprecated in favor of more up-to-date features introduced in recent Spring versions.

- form controller hierarchy deprecated in favor of @MVC form object handling.

This change affects the building of web applications with the AbstractFormController class. We recommend that you use the MVC annotations such as @Controller, @RequestMapping, @SessionAttributes, @ModelAttribute, @InitBinder, and so on, instead.

- Struts 1.x Action delegation support deprecated in favor of integration provided by Struts 2.

This change affects Struts 1. It is recommended that you move to Struts 2 or Spring MVC for future web development.

- JUnit 3.8 test class hierarchy deprecated in favor of test context framework.

This change affects JUnit 3.8 support. It is recommended that you move to JUnit 4 or TestNG for your unit tests.

3.3 Deprecated XSD Schema Features

The current spring-beans 3.0 XSD favors a more explicit configuration and relies less on the container to provide some configuration "magic".

The following attributes were removed:

- "default-dependency-check" attribute of the <beans> root element
- "dependency-check" attribute of the <bean> element

These attributes provided dependency check defaulting that forced the container to "guess" how the checks should be done. We now recommend that you either code programmatic checks in the afterPropertiesSet method or use the @Required annotation.

Also, the "autodetect" option for the following attributes has been removed:

- "default-autowire" attribute of the <beans> root element
- "autowire" attribute of the <bean> element

The "autodetect" option again provided too much "magic" and a more explicit declaration is preferred.

You can still continue to use these deprecated features using the 2.5 schema definition. You just need to use the "<http://www.springframework.org/schema/beans/spring-beans-2.5.xsd>" version of the schema.

The removed attributes only affect the 3.0 schema declaration and the code supporting the 2.5 schema is still provided in Spring 3.0.

4. Application Migration and Compatibility Concerns

This chapter covers compatibility issues you might encounter as you migrate Spring 2.5.6 applications to Spring 3.0.

4.1 Changing Spring 2.5.6 Generics to Match 3.0 Generics

If you use generics in your 2.5.6 code, you might find a mismatch with the generics declarations used in Spring 3.0.

For example, the following code worked in Spring 2.5.6:

```
List<Map> list = jdbcTemplate.queryForList(queryString);
```

This code does not work in Spring 3.0 because the actual return type of that method is now declared as `List<Map<String, Object>>`. The Java compiler gives an "inconvertible types" error. You must change the code to use the matching generic return type:

```
List<Map<String, Object>> list = jdbcTemplate.queryForList(queryString);
```

The problem mentioned above only causes a problem when you try to compile the code. Binary compatibility should still be maintained and the first, compiled with Spring 2.5 jars, example should execute without problems with the Spring 3.0 jar files.

4.2 Implementing Remote Support Between Spring 2.5.6 Server Applications and 3.0 Client Applications

The build process for the remote support has changed; we no longer generate the stub and skeleton classes that were used for the `RmiInvocationWrapper` class. These classes are no longer needed with Spring 3.0.

However, if you have a Spring 2.5.6 server application accessed by a new Spring 3.0 client application, then you do need the stub class. You can simply include the `org.springframework.remoting.rmiRmiInvocationWrapper_Stub.class` from Spring 2.5.6 in the classpath of the Spring 3.0 client application.

If you prefer, another way of achieving this is to create a "compatibility" jar containing the 2.5.6 stub class. Open a command window and change the current directory to the directory in which you extracted the most recent Spring Framework 2.5.6 download.

Extract the needed class with the jar utility (ships with your Java distribution):

```
jar -xvf modules/spring-context.jar org/springframework/remoting/rmi/RmiInvocationWrapper_Stub.class
```

You should see this message:

```
inflated:  
org/springframework/remoting/rmi/RmiInvocationWrapper_Stub.class
```

Create the compatibility jar, again using the jar utility:

```
jar -cvf spring-2.5.6-rmi-compatibility.jar org/springframework/remoting/rmi
```

This command creates a jar file in the current directory that you can add to the classpath of the Spring 3.0 client application.

4.3 Deprecated Elements and Effects on Schema

If you use the unversioned schema in your namespace references and use any deprecated elements (see Section 3.3, “Deprecated XSD Schema Features”), you will get XML parsing errors. Either change your application context configuration, removing the deprecated attributes, or change to explicitly use the 2.5 version of the schema as in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
       xsi:schemaLocation="http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd"  
       default-dependency-check="all"  
       default-autowire="autodetect">  
  
    <!-- bean definitions ... -->  
  
</beans>
```

The removed attributes only affect the schema. The underlying code supporting the 2.5 schema is still provided in Spring 3.0.

4.4 Avoiding Naming Conflicts with Spring Bean conversionService

Spring Framework 3.0 looks for a bean named 'conversionService' and expects it to be of the type `org.springframework.core.convert.ConversionService`. If a project has an existing bean named 'conversionService' (not unlikely especially for Spring Web Flow projects), you need to rename the bean to avoid a conflict.

Appendix A. Code Changes Between Spring 2.5.6 and 3.0.0

This appendix provides a report that documents API changes between Spring Framework 2.5.6 and 3.0.0. API codes used in the report are listed first, followed by the report. The report was produced using CLIRR version 0.7. It is organized by package and only lists classes that have changes that could potentially break your application. If you use the listed classes you should investigate the impact these changes have on your application code.

A.1 API Change Codes

The report uses the following codes.

- 1000 = SCOPE_INCREASED
- 1001 = SCOPE_DECREASED
- 1002 = ERROR_DETERMINING_SCOPE_OLD
- 1003 = ERROR_DETERMINING_SCOPE_NEW
- 2000 = GENDER_CLASS_TO_INTERFACE
- 2001 = GENDER_INTERFACE_TO_CLASS
- 3000 = MODIFIER_UNABLE_TO_DETERMINE_CLASS_SCOPE
- 3001 = MODIFIER_REMOVED_FINAL
- 3002 = MODIFIER_ADDED_FINAL_TO_EFFECTIVE_FINAL
- 3003 = MODIFIER_ADDED_FINAL
- 3004 = MODIFIER_REMOVED_ABSTRACT
- 3005 = MODIFIER_ADDED_ABSTRACT
- 4000 = IFACE_ADDED
- 4001 = IFACE_REMOVED
- 5000 = ADDED_CLASS_TO_SUPERCLASSES
- 5001 = REMOVED_CLASS_FROM_SUPERCLASSES

- 6000 = FIELD_ADDED
- 6001 = FIELD_REMOVED
- 6002 = FIELD_NOT_CONSTANT
- 6003 = FIELD_CONSTANT_CHANGED
- 6004 = FIELD_TYPE_CHANGED
- 6005 = FIELD_NOW_NON_FINAL
- 6006 = FIELD_NOW_FINAL
- 6007 = FIELD_NOW_NON_STATIC
- 6008 = FIELD_NOW_STATIC
- 6009 = FIELD_MORE_ACCESSIBLE
- 6010 = FIELD_LESS_ACCESSIBLE
- 6011 = CONSTANT_FIELD_REMOVED
- 7000 = METHOD_NOW_IN_SUPERCLASS
- 7001 = METHOD_NOW_IN_INTERFACE
- 7002 = METHOD_REMOVED
- 7003 = METHOD_OVERRIDE_REMOVED
- 7004 = METHOD_ARGCOUNT_CHANGED
- 7005 = METHOD_PARAMTYPE_CHANGED
- 7006 = METHOD_RETURNTYPE_CHANGED
- 7007 = METHOD_DEPRECATED
- 7008 = METHOD_UNDEPRECATED
- 7009 = METHOD_LESS_ACCESSIBLE
- 7010 = METHOD_MORE_ACCESSIBLE
- 7011 = METHOD_ADDED
- 7012 = METHOD_ADDED_TO_INTERFACE

- 7013 = ABSTRACT_METHOD_ADDED
- 7014 = METHOD_NOW_FINAL
- 7015 = METHOD_NOW_NONFINAL
- 8000 = CLASS_ADDED
- 8001 = CLASS_REMOVED
- 9000 = UNABLE_TO_DETERMINE_CLASS_SCOPE
- 10000 = CLASS_FORMAT_VERSION_INCREASED
- 10001 = CLASS_FORMAT_VERSION_DECREASED

A.2 API Change Report

Produced using CLIRR version 0.7 SNAPSHOT

aop.framework

- **org.springframework.aop.framework.AbstractSingletonProxyFactoryBean**
Method 'protected java.lang.Object getProxy(org.springframework.aop.framework.AopProxy)' has been removed [7002]
- **org.springframework.aop.framework.AopProxyUtils**
Method 'public java.lang.Class getTargetClass(java.lang.Object)' has been removed [7002]
- **org.springframework.aop.framework.ProxyFactory**
Removed org.springframework.aop.framework.AopProxy from the set of implemented interfaces [4001]
- **org.springframework.aop.framework.autoproxy.AbstractAutoProxyCreator**
Method 'protected boolean isInfrastructureClass(java.lang.Class, java.lang.String)' has been removed [7002]

aop.interceptor

- **org.springframework.aop.interceptor.ClassLoaderAnalyzerInterceptor**

Class `org.springframework.aop.interceptor.ClassLoaderAnalyzerInterceptor` removed [8001]

aop.support

- **org.springframework.aop.support.IntroductionInfoSupport**

Removed field `logger` [6001]

Field `publishedInterfaces` is now final [6006]

aop.target

- **org.springframework.aop.target.AbstractBeanFactoryBasedTargetSource**

Method `'protected java.lang.Object writeReplace()'` has been removed [7002]

beans

- **org.springframework.beans.BeanUtils**

Method `'public boolean isAssignable(java.lang.Class, java.lang.Class)'` has been removed [7002]

Method `'public boolean isAssignable(java.lang.Class, java.lang.Object)'` has been removed [7002]

- **org.springframework.beans.BeanWrapper**

Method `'public boolean isAutoGrowNestedPaths()'` has been added to an interface [7012]

Method `'public void setAutoGrowNestedPaths(boolean)'` has been added to an interface [7012]

Method `'public void setWrappedInstance(java.lang.Object)'` has been removed [7002]

- **org.springframework.beans.BeanWrapperImpl**

Method `'public java.lang.Object doTypeConversionIfNecessary(java.lang.Object, java.lang.Class)'` has been removed [7002]

- **org.springframework.beans.ConfigurablePropertyAccessor**

Method `'public org.springframework.core.convert.ConversionService getConversionService()'` has been added to an interface [7012]

Method `'public void setConversionService(org.springframework.core.convert.ConversionService)'` has been added to an interface [7012]

- **org.springframework.beans.MutablePropertyValues**

Method 'public void clear()' has been removed [7002]

- **org.springframework.beans.PropertyAccessor**

Method 'public org.springframework.core.convert.TypeDescriptor getPropertyTypeDescriptor(java.lang.String)' has been added to an interface [7012]

beans.factory

- **org.springframework.beans.factory.BeanFactory**

Method 'public java.lang.Object getBean(java.lang.Class)' has been added to an interface [7012]

- **org.springframework.beans.factory.ListableBeanFactory**

Method 'public java.lang.annotation.Annotation findAnnotationOnBean(java.lang.String, java.lang.Class)' has been added to an interface [7012]

Method 'public java.util.Map getBeansWithAnnotation(java.lang.Class)' has been added to an interface [7012]

- **org.springframework.beans.factory.annotation.Autowired**

Removed field INHERITED [6001]

- **org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor**

Method 'protected java.lang.Class getAutowiredAnnotationType()' has been removed [7002]

- **org.springframework.beans.factory.annotation.InjectionMetadata**

In method 'public InjectionMetadata(java.lang.Class)' the number of arguments has changed [7004]

Method 'public InjectionMetadata()' has been removed [7002]

Method 'public void addInjectedField(org.springframework.beans.factory.annotation.InjectionMetadata\$InjectedElement)' has been removed [7002]

Method 'public void addInjectedMethod(org.springframework.beans.factory.annotation.InjectionMetadata\$InjectedElement)' has been removed [7002]

Method 'public void injectFields(java.lang.Object, java.lang.String)' has been removed [7002]

Method 'public void injectMethods(java.lang.Object, java.lang.String,

`org.springframework.beans.PropertyValues`)' has been removed [7002]

- **org.springframework.beans.factory.config.BeanDefinition**

Method `'public java.lang.String[] getDependsOn()'` has been added to an interface [7012]

Method `'public boolean isPrimary()'` has been added to an interface [7012]

Method `'public boolean isPrototype()'` has been added to an interface [7012]

Method `'public void setDependsOn(java.lang.String[])'` has been added to an interface [7012]

Method `'public void setLazyInit(boolean)'` has been added to an interface [7012]

Method `'public void setPrimary(boolean)'` has been added to an interface [7012]

- **org.springframework.beans.factory.config.ConfigurableBeanFactory**

Method `'public void addEmbeddedValueResolver(org.springframework.util.StringValueResolver)'` has been added to an interface [7012]

Method `'public java.security.AccessControlContext getAccessControlContext()'` has been added to an interface [7012]

Method `'public org.springframework.beans.factory.config.BeanExpressionResolver getBeanExpressionResolver()'` has been added to an interface [7012]

Method `'public org.springframework.core.convert.ConversionService getConversionService()'` has been added to an interface [7012]

Method `'public void registerCustomEditor(java.lang.Class, java.beans.PropertyEditor)'` has been removed [7002]

Method `'public java.lang.String resolveEmbeddedValue(java.lang.String)'` has been added to an interface [7012]

Method `'public void setBeanExpressionResolver(org.springframework.beans.factory.config.BeanExpressionResolver)'` has been added to an interface [7012]

Method `'public void setConversionService(org.springframework.core.convert.ConversionService)'` has been added to an interface [7012]

- **org.springframework.beans.factory.config.ConstructorArgumentValues**

Parameter 3 of `'public org.springframework.beans.factory.config.ConstructorArgumentValues$ValueHolder getArgumentValue(int, java.lang.Class, java.util.Set)'` has changed its type to `java.lang.String` [7005]

Parameter 2 of 'public

org.springframework.beans.factory.config.ConstructorArgumentValues\$ValueHolder

getGenericArgumentValue(java.lang.Class, java.util.Set)' has changed its type to java.lang.String [7005]

- **org.springframework.beans.factory.config.DependencyDescriptor**

Return type of method 'public java.lang.Object[] getAnnotations()' has been changed to java.lang.annotation.Annotation[] [7006]

- **org.springframework.beans.factory.config.PropertiesFactoryBean**

Method 'protected java.lang.Object createInstance()' has been removed [7002]

- **org.springframework.beans.factory.config.Scope**

Method 'public java.lang.Object resolveContextualObject(java.lang.String)' has been added to an interface [7012]

- **org.springframework.beans.factory.generic.GenericBeanFactoryAccessor**

Class org.springframework.beans.factory.generic.GenericBeanFactoryAccessor removed [8001]

- **org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory**

In method 'protected void invokeCustomInitMethod(java.lang.String, java.lang.Object, java.lang.String, boolean)' the number of arguments has changed [7004]

- **org.springframework.beans.factory.support.AbstractBeanFactory**

Accessibility of method 'protected java.lang.Object doGetBean(java.lang.String, java.lang.Class, java.lang.Object[], boolean)' has been decreased from protected to private [7009]

Method 'public void registerCustomEditor(java.lang.Class, java.beans.PropertyEditor)' has been removed [7002]

Method 'protected java.lang.Class

resolveBeanClass(org.springframework.beans.factory.support.RootBeanDefinition, java.lang.String)' has been removed [7002]

- **org.springframework.beans.factory.support.AutowireCandidateResolver**

Method 'public java.lang.Object

getSuggestedValue(org.springframework.beans.factory.config.DependencyDescriptor)' has been added to an interface [7012]

- **org.springframework.beans.factory.support.BeanDefinitionReader**

Method 'public org.springframework.beans.factory.support.BeanDefinitionRegistry getBeanFactory()' has been removed [7002]

- **org.springframework.beans.factory.support.BeanDefinitionReaderUtils**

Method 'public org.springframework.beans.factory.support.AbstractBeanDefinition createBeanDefinition(java.lang.String, java.lang.String, org.springframework.beans.factory.config.ConstructorArgumentValues, org.springframework.beans.MutablePropertyValues, java.lang.ClassLoader)' has been removed [7002]

- **org.springframework.beans.factory.support.DefaultListableBeanFactory**

Parameter 2 of 'protected java.lang.String determinePrimaryCandidate(java.util.Map, java.lang.Class)' has changed its type to org.springframework.beans.factory.config.DependencyDescriptor [7005]

- **org.springframework.beans.factory.support.RootBeanDefinition**

Return type of method 'public org.springframework.beans.factory.support.AbstractBeanDefinition cloneBeanDefinition()' has been changed to org.springframework.beans.factory.support.RootBeanDefinition [7006]

- **org.springframework.beans.factory.xml.AbstractSimpleBeanDefinitionParser**

In method 'protected void doParse(org.w3c.dom.Element, org.springframework.beans.factory.support.BeanDefinitionBuilder)' the number of arguments has changed [7004]

- **org.springframework.beans.factory.xml.BeanDefinitionParserDelegate**

In method 'protected java.lang.Object buildTypedStringValue(java.lang.String, java.lang.String, org.w3c.dom.Element)' the number of arguments has changed [7004]

Accessibility of method 'public java.lang.Object parseKeyElement(org.w3c.dom.Element, org.springframework.beans.factory.config.BeanDefinition, java.lang.String)' has been decreased from public to protected [7009]

- **org.springframework.beans.factory.xml.NamespaceHandlerSupport**

Method 'protected org.springframework.beans.factory.xml.BeanDefinitionDecorator findDecoratorForNode(org.w3c.dom.Node)' has been removed [7002]

Method 'protected org.springframework.beans.factory.xml.BeanDefinitionParser findParserForElement(org.w3c.dom.Element)' has been removed [7002]

- **org.springframework.beans.factory.xml.PluggableSchemaResolver**

Method 'protected java.lang.String getSchemaMapping(java.lang.String)' has been removed [7002]

- **org.springframework.beans.factory.xml.XmlBeanDefinitionParser**

Class org.springframework.beans.factory.xml.XmlBeanDefinitionParser removed [8001]

- **org.springframework.beans.factory.xml.XmlBeanDefinitionReader**

Method 'public void setParserClass(java.lang.Class)' has been removed [7002]

beans.support

- **org.springframework.beans.support.PagedListSourceProvider**

Class org.springframework.beans.support.PagedListSourceProvider removed [8001]

- **org.springframework.beans.support.RefreshablePagedListHolder**

Class org.springframework.beans.support.RefreshablePagedListHolder removed [8001]

context

- **org.springframework.context.ConfigurableApplicationContext**

Method 'public void setId(java.lang.String)' has been added to an interface [7012]

context.annotation

- **org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider**

Accessibility of field DEFAULT_RESOURCE_PATTERN has been weakened from protected to private [6010]

- **org.springframework.context.annotation.ComponentScanBeanDefinitionParser**

In method 'protected void parseTypeFilters(org.w3c.dom.Element, org.springframework.context.annotation.ClassPathBeanDefinitionScanner, org.springframework.beans.factory.xml.XmlReaderContext)' the number of arguments has changed [7004]

- **org.springframework.context.annotation.Scope**

Method 'public org.springframework.context.annotation.ScopedProxyMode proxyMode()' has been added to an interface [7012]

context.event

- **org.springframework.context.event.AbstractApplicationEventMulticaster**

Method 'public void setCollectionClass(java.lang.Class)' has been removed [7002]

Method 'public void setConcurrentUpdates(boolean)' has been removed [7002]

- **org.springframework.context.event.ApplicationEventMulticaster**

Method 'public void addApplicationListenerBean(java.lang.String)' has been added to an interface [7012]

Method 'public void removeApplicationListenerBean(java.lang.String)' has been added to an interface [7012]

- **org.springframework.context.event.SimpleApplicationEventMulticaster**

Return type of method 'protected org.springframework.core.task.TaskExecutor getTaskExecutor()' has been changed to java.util.concurrent.Executor [7006]

Parameter 1 of 'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)' has changed its type to java.util.concurrent.Executor [7005]

- **org.springframework.context.event.SourceFilteringListener**

Removed org.springframework.context.ApplicationListener from the set of implemented interfaces [4001]

context.support

- **org.springframework.context.support.AbstractApplicationContext**

Return type of method 'public java.util.List getApplicationListeners()' has been changed to java.util.Collection [7006]

- **org.springframework.context.support.ResourceMapFactoryBean**

Class org.springframework.context.support.ResourceMapFactoryBean removed [8001]

core

- **org.springframework.core.MethodParameter**

Return type of method 'public java.lang.Object[] getParameterAnnotations()' has been changed to java.lang.annotation.Annotation[] [7006]

- **org.springframework.core.ReflectiveVisitorHelper**

Class `org.springframework.core.ReflectiveVisitorHelper` removed [8001]

core.style

- **org.springframework.core.style.DefaultToStringStyler**

Parameter 1 of 'public void styleEnd(java.lang.StringBuffer, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleField(java.lang.StringBuffer, java.lang.String, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'protected void styleFieldEnd(java.lang.StringBuffer, java.lang.String)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleFieldSeparator(java.lang.StringBuffer)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'protected void styleFieldStart(java.lang.StringBuffer, java.lang.String)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleStart(java.lang.StringBuffer, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleValue(java.lang.StringBuffer, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

- **org.springframework.core.style.ToStringStyler**

Parameter 1 of 'public void styleEnd(java.lang.StringBuffer, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleField(java.lang.StringBuffer, java.lang.String, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleFieldSeparator(java.lang.StringBuffer)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleStart(java.lang.StringBuffer, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

Parameter 1 of 'public void styleValue(java.lang.StringBuffer, java.lang.Object)' has changed its type to `java.lang.StringBuilder` [7005]

core.task

- **org.springframework.core.task.AsyncTaskExecutor**

Method 'public java.util.concurrent.Future submit(java.lang.Runnable)' has been added to an interface [7012]

Method 'public java.util.concurrent.Future submit(java.util.concurrent.Callable)' has been added to an interface [7012]

- **org.springframework.core.task.SimpleAsyncTaskExecutor**

Removed field DEFAULT_THREAD_NAME_PREFIX [6001]

Method 'public int getConcurrencyLimit()' is now final [7014]

Method 'public boolean isThrottleActive()' is now final [7014]

- **org.springframework.core.task.TaskRejectedException**

Removed org.springframework.core.NestedRuntimeException from the list of superclasses [5001]

- **org.springframework.core.task.TaskTimeoutException**

Removed org.springframework.core.NestedRuntimeException from the list of superclasses [5001]

core.type

- **org.springframework.core.type.AnnotationMetadata**

Method 'public java.util.Set getAnnotatedMethods(java.lang.String)' has been added to an interface [7012]

Method 'public java.util.Map getAnnotationAttributes(java.lang.String, boolean)' has been added to an interface [7012]

Method 'public boolean hasAnnotatedMethods(java.lang.String)' has been added to an interface [7012]

Method 'public boolean isAnnotated(java.lang.String)' has been added to an interface [7012]

- **org.springframework.core.type.ClassMetadata**

Method 'public boolean isFinal()' has been added to an interface [7012]

- **org.springframework.core.type.classreading.MetadataReader**

Method 'public org.springframework.core.io.Resource getResource()' has been added to an interface [7012]

instrument.classloading

- **org.springframework.instrument.classloading.ShadowingClassLoader**

Method 'protected boolean isClassNameExcludedFromShadowing(java.lang.String)' has been removed [7002]

- **org.springframework.instrument.classloading.glassfish.GlassFishLoadTimeWeaver**

Method 'protected com.sun.enterprise.loader.InstrumentableClassLoader determineClassLoader(java.lang.ClassLoader)' has been removed [7002]

jca.endpoint

- **org.springframework.jca.endpoint.GenericMessageEndpointManager**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

jca.work

- **org.springframework.jca.work.SimpleTaskWorkManager**

Parameter 1 of 'public void setAsyncTaskExecutor(org.springframework.core.task.TaskExecutor)' has changed its type to org.springframework.core.task.AsyncTaskExecutor [7005]

jdbc.core

- **org.springframework.jdbc.core.metadata.TableMetaDataContext**

Method 'public boolean isAccessTableParameterMetaData()' has been removed [7002]

Method 'public void setAccessTableParameterMetaData(boolean)' has been removed [7002]

- **org.springframework.jdbc.core.namedparam.NamedParameterJdbcOperations**

Method 'public int[] batchUpdate(java.lang.String, java.util.Map[])' has been added to an interface [7012]

Method 'public int[] batchUpdate(java.lang.String, org.springframework.jdbc.core.namedparam.SqlParameterSource[])' has been added to an interface [7012]

- **org.springframework.jdbc.core.simple.SimpleJdbcCall**

In method 'public java.util.Map execute()' the number of arguments has changed [7004]

Parameter 2 of 'public java.lang.Object executeFunction(java.lang.Class, org.springframework.jdbc.core.namedparam.MapSqlParameterSource)' has changed its type to java.lang.Object[] [7005]

Parameter 2 of 'public java.lang.Object executeObject(java.lang.Class, org.springframework.jdbc.core.namedparam.MapSqlParameterSource)' has changed its type to java.lang.Object[] [7005]

- **org.springframework.jdbc.core.simple.SimpleJdbcCallOperations**

In method 'public java.util.Map execute()' the number of arguments has changed [7004]

Parameter 2 of 'public java.lang.Object executeFunction(java.lang.Class, org.springframework.jdbc.core.namedparam.MapSqlParameterSource)' has changed its type to java.lang.Object[] [7005]

Method 'public java.lang.Object executeFunction(java.lang.Class, org.springframework.jdbc.core.namedparam.SqlParameterSource)' has been added to an interface [7012]

Parameter 2 of 'public java.lang.Object executeObject(java.lang.Class, org.springframework.jdbc.core.namedparam.MapSqlParameterSource)' has changed its type to java.lang.Object[] [7005]

Method 'public java.lang.Object executeObject(java.lang.Class, org.springframework.jdbc.core.namedparam.SqlParameterSource)' has been added to an interface [7012]

- **org.springframework.jdbc.core.simple.SimpleJdbcInsertOperations**

Method 'public org.springframework.jdbc.core.simple.SimpleJdbcInsertOperations includeSynonymsForTableColumnMetaData()' has been added to an interface [7012]

Method 'public org.springframework.jdbc.core.simple.SimpleJdbcInsertOperations withoutTableColumnMetaDataAccess()' has been added to an interface [7012]

- **org.springframework.jdbc.core.simple.SimpleJdbcOperations**

Method 'public java.util.List query(java.lang.String, org.springframework.jdbc.core.RowMapper, java.util.Map)' has been added to an interface [7012]

Method 'public java.util.List query(java.lang.String, org.springframework.jdbc.core.RowMapper, org.springframework.jdbc.core.namedparam.SqlParameterSource)' has been added to an interface [7012]

Method 'public java.util.List query(java.lang.String, org.springframework.jdbc.core.RowMapper, java.lang.Object[])' has been added to an interface [7012]

Method 'public java.lang.Object queryForObject(java.lang.String, org.springframework.jdbc.core.RowMapper, java.util.Map)' has been added to an interface [7012]

Method 'public java.lang.Object queryForObject(java.lang.String, org.springframework.jdbc.core.RowMapper, org.springframework.jdbc.core.namedparam.SqlParameterSource)' has been added to an interface [7012]

Method 'public java.lang.Object queryForObject(java.lang.String, org.springframework.jdbc.core.RowMapper, java.lang.Object[])' has been added to an interface [7012]

jdbc.support

- **org.springframework.jdbc.support.nativejdbc.XAPoolNativeJdbcExtractor**

Class org.springframework.jdbc.support.nativejdbc.XAPoolNativeJdbcExtractor removed [8001]

jms.listener

- **org.springframework.jms.listener.AbstractJmsListeningContainer**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

- **org.springframework.jms.listener.AbstractMessageListenerContainer**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

- **org.springframework.jms.listener.AbstractPollingMessageListenerContainer**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

- **org.springframework.jms.listener.DefaultMessageListenerContainer**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

Parameter 1 of 'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)' has changed its type to java.util.concurrent.Executor [7005]

- **org.springframework.jms.listener.DefaultMessageListenerContainer102**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

- **org.springframework.jms.listener.SimpleMessageListenerContainer**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

Parameter 1 of 'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)' has changed its type to java.util.concurrent.Executor [7005]

- **org.springframework.jms.listener.SimpleMessageListenerContainer102**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

- **org.springframework.jms.listener.endpoint.JmsMessageEndpointManager**

Removed org.springframework.context.Lifecycle from the set of implemented interfaces [4001]

- **org.springframework.jms.listener.endpoint.StandardJmsActivationSpecFactory**

Parameter 1 of 'public void setDefaultProperties(java.util.Properties)' has changed its type to java.util.Map [7005]

- **org.springframework.jms.listener.serversession.AbstractPoolingServerSessionFactory**

Class org.springframework.jms.listener.serversession.AbstractPoolingServerSessionFactory removed [8001]

- **org.springframework.jms.listener.serversession.CommonsPoolServerSessionFactory**

Class org.springframework.jms.listener.serversession.CommonsPoolServerSessionFactory removed [8001]

- **org.springframework.jms.listener.serversession.ListenerSessionManager**

Class org.springframework.jms.listener.serversession.ListenerSessionManager removed [8001]

- **org.springframework.jms.listener.serversession.ServerSessionFactory**

Class org.springframework.jms.listener.serversession.ServerSessionFactory removed [8001]

- **org.springframework.jms.listener.serversession.ServerSessionMessageListenerContainer**

Class org.springframework.jms.listener.serversession.ServerSessionMessageListenerContainer removed [8001]

- **org.springframework.jms.listener.serversession.ServerSessionMessageListenerContainer102**

Class org.springframework.jms.listener.serversession.ServerSessionMessageListenerContainer102 removed [8001]

- **org.springframework.jms.listener.serversession.SimpleServerSessionFactory**

Class org.springframework.jms.listener.serversession.SimpleServerSessionFactory removed [8001]

jmx.export

- **org.springframework.jmx.export.MBeanExportOperations**

Method 'public void unregisterManagedResource(javax.management.ObjectName)' has been added to an interface [7012]

- **org.springframework.jmx.export.assembler.AbstractReflectiveMBeanInfoAssembler**

Changed type of field ATTRIBUTE_OPERATION_VISIBILITY from java.lang.Integer to int [6004]

- **org.springframework.jmx.export.metadata.AttributesJmxAttributeSource**

Class org.springframework.jmx.export.metadata.AttributesJmxAttributeSource removed [8001]

- **org.springframework.jmx.export.metadata.JmxAttributeSource**

Method 'public org.springframework.jmx.export.metadata.ManagedMetric getManagedMetric(java.lang.reflect.Method)' has been added to an interface [7012]

jmx.support

- **org.springframework.jmx.support.WebLogicJndiMBeanServerFactoryBean**

Class org.springframework.jmx.support.WebLogicJndiMBeanServerFactoryBean removed [8001]

- **org.springframework.jmx.support.WebLogicMBeanServerFactoryBean**

Class org.springframework.jmx.support.WebLogicMBeanServerFactoryBean removed [8001]

metadata

- **org.springframework.metadata.Attributes**

Class org.springframework.metadata.Attributes removed [8001]

metadata.common

- **org.springframework.metadata.common.CommonsAttributes**

Class org.springframework.metadata.common.CommonsAttributes removed [8001]

mock.web

- **org.springframework.mock.web.MockHttpServletRequest**

Method 'public void addRole(java.lang.String)' has been removed [7002]

orm.hibernate3

- **org.springframework.orm.hibernate3.FilterDefinitionFactoryBean**

Parameter 1 of 'public void setParameterTypes(java.util.Properties)' has changed its type to java.util.Map [7005]

- **org.springframework.orm.hibernate3.HibernateTemplate**

Method 'public java.lang.Object execute(org.springframework.orm.hibernate3.HibernateCallback, boolean)' has been removed [7002]

orm.ibatis

- **org.springframework.orm.ibatis.SqlMapClientOperations**

Method 'public com.ibatis.common.util.PaginatedList queryForPaginatedList(java.lang.String, int)' has been removed [7002]

Method 'public com.ibatis.common.util.PaginatedList queryForPaginatedList(java.lang.String, java.lang.Object, int)' has been removed [7002]

- **org.springframework.orm.ibatis.SqlMapClientTemplate**

Method 'public com.ibatis.common.util.PaginatedList queryForPaginatedList(java.lang.String, int)' has been removed [7002]

Method 'public com.ibatis.common.util.PaginatedList queryForPaginatedList(java.lang.String, java.lang.Object, int)' has been removed [7002]

orm.jdo

- **org.springframework.orm.jdo.DefaultJdoDialect**

Method 'public void flush(javax.jdo.PersistenceManager)' has been removed [7002]

- **org.springframework.orm.jdo.JdoDialect**

Method 'public void flush(javax.jdo.PersistenceManager)' has been removed [7002]

- **org.springframework.orm.jdo.JdoOperations**

Method 'public java.lang.Object attachCopy(java.lang.Object)' has been removed [7002]

Method 'public java.util.Collection attachCopyAll(java.util.Collection)' has been removed [7002]

- **org.springframework.orm.jdo.JdoTemplate**

Method 'public java.lang.Object attachCopy(java.lang.Object)' has been removed [7002]

Method 'public java.util.Collection attachCopyAll(java.util.Collection)' has been removed [7002]

orm.toplink

- **org.springframework.orm.toplink.AbstractSessionFactory**

Class org.springframework.orm.toplink.AbstractSessionFactory removed [8001]

- **org.springframework.orm.toplink.LocalSessionFactory**

Class org.springframework.orm.toplink.LocalSessionFactory removed [8001]

- **org.springframework.orm.toplink.LocalSessionFactoryBean**

Class org.springframework.orm.toplink.LocalSessionFactoryBean removed [8001]

- **org.springframework.orm.toplink.ServerSessionFactory**

Class org.springframework.orm.toplink.ServerSessionFactory removed [8001]

- **org.springframework.orm.toplink.SessionBrokerSessionFactory**

Class org.springframework.orm.toplink.SessionBrokerSessionFactory removed [8001]

- **org.springframework.orm.toplink.SessionFactory**

Class org.springframework.orm.toplink.SessionFactory removed [8001]

- **org.springframework.orm.toplink.SessionFactoryUtils**

Class org.springframework.orm.toplink.SessionFactoryUtils removed [8001]

- **org.springframework.orm.toplink.SessionHolder**

Class org.springframework.orm.toplink.SessionHolder removed [8001]

- **org.springframework.orm.toplink.SessionReadCallback**

Class org.springframework.orm.toplink.SessionReadCallback removed [8001]

- **org.springframework.orm.toplink.SingleSessionFactory**
Class org.springframework.orm.toplink.SingleSessionFactory removed [8001]
- **org.springframework.orm.toplink.TopLinkAccessor**
Class org.springframework.orm.toplink.TopLinkAccessor removed [8001]
- **org.springframework.orm.toplink.TopLinkCallback**
Class org.springframework.orm.toplink.TopLinkCallback removed [8001]
- **org.springframework.orm.toplink.TopLinkInterceptor**
Class org.springframework.orm.toplink.TopLinkInterceptor removed [8001]
- **org.springframework.orm.toplink.TopLinkJdbcException**
Class org.springframework.orm.toplink.TopLinkJdbcException removed [8001]
- **org.springframework.orm.toplink.TopLinkOperations**
Class org.springframework.orm.toplink.TopLinkOperations removed [8001]
- **org.springframework.orm.toplink.TopLinkOptimisticLockingFailureException**
Class org.springframework.orm.toplink.TopLinkOptimisticLockingFailureException removed [8001]
- **org.springframework.orm.toplink.TopLinkQueryException**
Class org.springframework.orm.toplink.TopLinkQueryException removed [8001]
- **org.springframework.orm.toplink.TopLinkSystemException**
Class org.springframework.orm.toplink.TopLinkSystemException removed [8001]
- **org.springframework.orm.toplink.TopLinkTemplate**
Class org.springframework.orm.toplink.TopLinkTemplate removed [8001]
- **org.springframework.orm.toplink.TopLinkTransactionManager**
Class org.springframework.orm.toplink.TopLinkTransactionManager removed [8001]
- **org.springframework.orm.toplink.UnitOfWorkCallback**
Class org.springframework.orm.toplink.UnitOfWorkCallback removed [8001]
- **org.springframework.orm.toplink.support.CommonsLoggingSessionLog**

Class org.springframework.orm.toplink.support.CommonsLoggingSessionLog removed [8001]

- **org.springframework.orm.toplink.support.CommonsLoggingSessionLog904**

Class org.springframework.orm.toplink.support.CommonsLoggingSessionLog904 removed [8001]

- **org.springframework.orm.toplink.support.TopLinkDaoSupport**

Class org.springframework.orm.toplink.support.TopLinkDaoSupport removed [8001]

- **org.springframework.orm.toplink.support.TransactionAwareSessionAdapter**

Class org.springframework.orm.toplink.support.TransactionAwareSessionAdapter removed [8001]

remoting.jaxrpc

- **org.springframework.remoting.jaxrpc.support.AxisBeanMappingServicePostProcessor**

Class org.springframework.remoting.jaxrpc.support.AxisBeanMappingServicePostProcessor removed [8001]

remoting.jaxws

- **org.springframework.remoting.jaxws.AbstractJaxWsServiceExporter**

Abstract method 'protected void publishEndpoint(javax.xml.ws.Endpoint, javax.xml.ws.WebServiceProvider)' has been added [7013]

Method 'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)' has been removed [7002]

- **org.springframework.remoting.jaxws.LocalJaxWsServiceFactory**

Method 'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)' has been removed [7002]

remoting.rmi

- **org.springframework.remoting.rmi.RmiInvocationWrapper_Skel**

Class org.springframework.remoting.rmi.RmiInvocationWrapper_Skel removed [8001]

- **org.springframework.remoting.rmi.RmiInvocationWrapper_Stub**

Class org.springframework.remoting.rmi.RmiInvocationWrapper_Stub removed [8001]

- **org.springframework.remoting.rmi._RmiInvocationHandler_Stub**

Class org.springframework.remoting.rmi._RmiInvocationHandler_Stub removed [8001]

- **org.springframework.remoting.rmi._RmiInvocationWrapper_Tie**

Class org.springframework.remoting.rmi._RmiInvocationWrapper_Tie removed [8001]

remoting.support

- **org.springframework.remoting.support.SimpleHttpServerFactoryBean**

Method 'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)' has been removed [7002]

scheduling

- **org.springframework.scheduling.SchedulingTaskExecutor**

Removed org.springframework.core.task.TaskExecutor from the set of implemented interfaces [4001]

scheduling.backportconcurrent

- **org.springframework.scheduling.backportconcurrent.ConcurrentTaskExecutor**

Method 'public edu.emory.mathcs.backport.java.util.concurrent.Executor getConcurrentExecutor()' is now final [7014]

Method 'public void setConcurrentExecutor(edu.emory.mathcs.backport.java.util.concurrent.Executor)' is now final [7014]

- **org.springframework.scheduling.backportconcurrent.ScheduledExecutorFactoryBean**

Class org.springframework.scheduling.backportconcurrent.ScheduledExecutorFactoryBean removed [8001]

- **org.springframework.scheduling.backportconcurrent.ScheduledExecutorTask**

Class org.springframework.scheduling.backportconcurrent.ScheduledExecutorTask removed [8001]

scheduling.concurrent

- **org.springframework.scheduling.concurrent.ConcurrentTaskExecutor**

Removed `java.util.concurrent.Executor` from the set of implemented interfaces [4001]

Method `'public java.util.concurrent.Executor getConcurrentExecutor()'` is now final [7014]

Method `'public void setConcurrentExecutor(java.util.concurrent.Executor)'` is now final [7014]

- **`org.springframework.scheduling.concurrent.ScheduledExecutorFactoryBean`**

Removed field `logger` [6001]

- **`org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor`**

Removed `java.util.concurrent.Executor` from the set of implemented interfaces [4001]

Removed field `logger` [6001]

scheduling.quartz

- **`org.springframework.scheduling.quartz.SchedulerFactoryBean`**

Removed `org.springframework.context.Lifecycle` from the set of implemented interfaces [4001]

Return type of method `'public org.springframework.core.task.TaskExecutor getConfigTimeTaskExecutor()'` has been changed to `java.util.concurrent.Executor` [7006]

Parameter 1 of `'public void setTaskExecutor(org.springframework.core.task.TaskExecutor)'` has changed its type to `java.util.concurrent.Executor` [7005]

scheduling.support

- **`org.springframework.scheduling.support.DelegatingExceptionProofRunnable`**

Class `org.springframework.scheduling.support.DelegatingExceptionProofRunnable` removed [8001]

scheduling.timer

- **`org.springframework.scheduling.timer.TimerFactoryBean`**

Method `'protected java.util.Timer createTimer(boolean)'` has been removed [7002]

- **`org.springframework.scheduling.timer.TimerTaskExecutor`**

Parameter 1 of `'public void setDelay(int)'` has changed its type to `long` [7005]

test.annotation

- **org.springframework.test.annotation.DirtiesContext**

Method 'public org.springframework.test.annotation.DirtiesContext\$ClassMode classMode()' has been added to an interface [7012]

test.context

- **org.springframework.test.context.ContextConfiguration**

Method 'public java.lang.String[] value()' has been added to an interface [7012]

- **org.springframework.test.context.TestExecutionListener**

Method 'public void afterTestClass(org.springframework.test.context.TestContext)' has been added to an interface [7012]

Method 'public void beforeTestClass(org.springframework.test.context.TestContext)' has been added to an interface [7012]

- **org.springframework.test.context.TestExecutionListeners**

Method 'public java.lang.Class[] listeners()' has been added to an interface [7012]

- **org.springframework.test.context.junit4.SpringJUnit4ClassRunner**

Removed org.junit.internal.runners.JUnit4ClassRunner from the list of superclasses [5001]

Method 'protected void invokeTestMethod(java.lang.reflect.Method, org.junit.runner.notification.RunNotifier)' has been removed [7002]

test.web

- **org.springframework.test.web.AbstractModelAndViewTests**

Parameter 2 of 'protected java.lang.Object assertAndReturnModelAttributeOfType(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.lang.Class)' has changed its type to java.lang.String [7005]

Parameter 2 of 'protected void assertCompareListModelAttribute(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.util.List)' has changed its type to java.lang.String [7005]

Parameter 2 of 'protected void assertModelAttributeAvailable(org.springframework.web.servlet.ModelAndView, java.lang.Object)'

has changed its type to `java.lang.String` [7005]

Parameter 2 of 'protected void

`assertModelAttributeValue(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.lang.Object)`' has changed its type to `java.lang.String` [7005]

Parameter 2 of 'protected void

`assertSortAndCompareListModelAttribute(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.util.List, java.util.Comparator)`' has changed its type to `java.lang.String` [7005]

- **org.springframework.test.web.ModelAndViewAssert**

Parameter 2 of 'public java.lang.Object

`assertAndReturnModelAttributeOfType(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.lang.Class)`' has changed its type to `java.lang.String` [7005]

Parameter 2 of 'public void

`assertCompareListModelAttribute(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.util.List)`' has changed its type to `java.lang.String` [7005]

Parameter 2 of 'public void

`assertModelAttributeAvailable(org.springframework.web.servlet.ModelAndView, java.lang.Object)`' has changed its type to `java.lang.String` [7005]

Parameter 2 of 'public void

`assertModelAttributeValue(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.lang.Object)`' has changed its type to `java.lang.String` [7005]

Parameter 2 of 'public void

`assertSortAndCompareListModelAttribute(org.springframework.web.servlet.ModelAndView, java.lang.Object, java.util.List, java.util.Comparator)`' has changed its type to `java.lang.String` [7005]

transaction

- **org.springframework.transaction.TransactionStatus**

Method 'public void flush()' has been added to an interface [7012]

transaction.annotation

- **org.springframework.transaction.annotation.Transactional**

Method 'public java.lang.String value()' has been added to an interface [7012]

transaction.config

- **org.springframework.transaction.config.JtaTransactionManagerBeanDefinitionParser**

Field `DEFAULT_TRANSACTION_MANAGER_BEAN_NAME` has been removed, but it was previously a constant [6011]

transaction.interceptor

- **org.springframework.transaction.interceptor.AttributesTransactionAttributeSource**

Class `org.springframework.transaction.interceptor.AttributesTransactionAttributeSource` removed [8001]

- **org.springframework.transaction.interceptor.TransactionAspectSupport**

In method 'protected
`org.springframework.transaction.interceptor.TransactionAspectSupport$TransactionInfo
createTransactionIfNecessary(org.springframework.transaction.interceptor.TransactionAttribute,
java.lang.String)`' the number of arguments has changed [7004]

In method 'protected
`org.springframework.transaction.interceptor.TransactionAspectSupport$TransactionInfo
prepareTransactionInfo(org.springframework.transaction.interceptor.TransactionAttribute,
java.lang.String, org.springframework.transaction.TransactionStatus)`' the number of arguments has
changed [7004]

- **org.springframework.transaction.interceptor.TransactionAspectSupport\$TransactionInfo**

Added final modifier to class [3003]

In method 'public
`TransactionAspectSupport$TransactionInfo(org.springframework.transaction.interceptor.TransactionAspectSupport,
org.springframework.transaction.interceptor.TransactionAttribute, java.lang.String)`' the number of
arguments has changed [7004]

- **org.springframework.transaction.interceptor.TransactionAttribute**

Method 'public `java.lang.String getQualifier()`' has been added to an interface [7012]

transaction.jta

- **org.springframework.transaction.jta.JotmFactoryBean**

Class `org.springframework.transaction.jta.JotmFactoryBean` removed [8001]

- **org.springframework.transaction.jta.TransactionFactory**

Method 'public boolean supportsResourceAdapterManagedTransactions()' has been added to an interface [7012]

- **org.springframework.transaction.jta.WebSphereTransactionManagerFactoryBean**

Class org.springframework.transaction.jta.WebSphereTransactionManagerFactoryBean removed [8001]

transaction.support

- **org.springframework.transaction.support.DefaultTransactionDefinition**

Return type of method 'protected java.lang.StringBuffer getDefinitionDescription()' has been changed to java.lang.StringBuilder [7006]

- **org.springframework.transaction.support.ResourceHolderSynchronization**

Added abstract modifier to class [3005]

- **org.springframework.transaction.support.SmartTransactionObject**

Method 'public void flush()' has been added to an interface [7012]

- **org.springframework.transaction.support.TransactionSynchronization**

Method 'public void flush()' has been added to an interface [7012]

ui

- **org.springframework.ui.ExtendedModelMap**

Return type of method 'public org.springframework.ui.ModelMap addAttribute(java.lang.Object)' has been changed to org.springframework.ui.Model [7006]

Return type of method 'public org.springframework.ui.ModelMap addAttribute(java.lang.String, java.lang.Object)' has been changed to org.springframework.ui.Model [7006]

Return type of method 'public org.springframework.ui.Model addAttribute(java.lang.Object)' has been changed to org.springframework.ui.ModelMap [7006]

Return type of method 'public org.springframework.ui.Model addAttribute(java.lang.String, java.lang.Object)' has been changed to org.springframework.ui.ModelMap [7006]

util

- **org.springframework.util.AntPathMatcher**

In method 'protected boolean doMatch(java.lang.String, java.lang.String, boolean)' the number of arguments has changed [7004]

- **org.springframework.util.CachingMapDecorator**

Added abstract modifier to class [3005]

Accessibility of field NULL_VALUE has been weakened from protected to private [6010]

- **org.springframework.util.ClassLoaderUtils**

Class org.springframework.util.ClassLoaderUtils removed [8001]

- **org.springframework.util.PathMatcher**

Method 'public java.lang.String combine(java.lang.String, java.lang.String)' has been added to an interface [7012]

Method 'public java.util.Map extractUriTemplateVariables(java.lang.String, java.lang.String)' has been added to an interface [7012]

Method 'public java.util.Comparator getPatternComparator(java.lang.String)' has been added to an interface [7012]

- **org.springframework.util.ResponseTimeMonitor**

Class org.springframework.util.ResponseTimeMonitor removed [8001]

- **org.springframework.util.ResponseTimeMonitorImpl**

Class org.springframework.util.ResponseTimeMonitorImpl removed [8001]

validation

- **org.springframework.validation.BindException**

Removed field ERROR_KEY_PREFIX [6001]

web.bind

- **org.springframework.web.bind.MissingServletRequestParameterException**

Method 'public java.lang.String getParameterName()' is now final [7014]

Method 'public java.lang.String getParameterType()' is now final [7014]

- **org.springframework.web.bind.RequestUtils**

Class org.springframework.web.bind.RequestUtils removed [8001]

- **org.springframework.web.bind.annotation.RequestMapping**

Method 'public java.lang.String[] headers()' has been added to an interface [7012]

- **org.springframework.web.bind.annotation.RequestParam**

Method 'public java.lang.String defaultValue()' has been added to an interface [7012]

- **org.springframework.web.bind.annotation.support.HandlerMethodInvoker**

In method 'public HandlerMethodInvoker(org.springframework.web.bind.annotation.support.HandlerMethodResolver, org.springframework.web.bind.support.WebBindingInitializer, org.springframework.web.bind.support.SessionAttributeStore, org.springframework.core.ParameterNameDiscoverer, org.springframework.web.bind.support.WebArgumentResolver[])' the number of arguments has changed [7004]

In method 'protected void doBind(org.springframework.web.context.request.NativeWebRequest, org.springframework.web.bind.WebDataBinder, boolean)' the number of arguments has changed [7004]

Accessibility of method 'protected void doBind(org.springframework.web.context.request.NativeWebRequest, org.springframework.web.bind.WebDataBinder, boolean)' has been decreased from protected to private [7009]

- **org.springframework.web.bind.annotation.support.HandlerMethodResolver**

In method 'public HandlerMethodResolver(java.lang.Class)' the number of arguments has changed [7004]

web.context

- **org.springframework.web.context.ContextLoaderServlet**

Class org.springframework.web.context.ContextLoaderServlet removed [8001]

- **org.springframework.web.context.request.FacesRequestAttributes**

Method 'protected javax.faces.context.ExternalContext getExternalContext()' is now final [7014]

Method 'protected javax.faces.context.FacesContext getFacesContext()' is now final [7014]

- **org.springframework.web.context.request.RequestAttributes**

Method 'public java.lang.Object resolveReference(java.lang.String)' has been added to an interface [7012]

- **org.springframework.web.context.request.RequestContextListener**

Removed field logger [6001]

- **org.springframework.web.context.request.WebRequest**

Method 'public java.lang.String getHeader(java.lang.String)' has been added to an interface [7012]

Method 'public java.util.Iterator getHeaderNames()' has been added to an interface [7012]

Method 'public java.lang.String[] getHeaderValues(java.lang.String)' has been added to an interface [7012]

Method 'public java.util.Iterator getParameterNames()' has been added to an interface [7012]

- **org.springframework.web.context.support.PerformanceMonitorListener**

Class org.springframework.web.context.support.PerformanceMonitorListener removed [8001]

web.multipart

- **org.springframework.web.multipart.MultipartRequest**

Method 'public java.util.List getFiles(java.lang.String)' has been added to an interface [7012]

Method 'public org.springframework.util.MultiValueMap getMultiFileMap()' has been added to an interface [7012]

- **org.springframework.web.multipart.commons.CommonsFileUploadSupport\$MultipartParsingResult**

Parameter 1 of 'public CommonsFileUploadSupport\$MultipartParsingResult(java.util.Map, java.util.Map)' has changed its type to org.springframework.util.MultiValueMap [7005]

Return type of method 'public java.util.Map getMultipartFiles()' has been changed to org.springframework.util.MultiValueMap [7006]

- **org.springframework.web.multipart.support.AbstractMultipartHttpServletRequest**

Return type of method 'protected java.util.Map getMultipartFiles()' has been changed to org.springframework.util.MultiValueMap [7006]

Parameter 1 of 'protected void setMultipartFiles(java.util.Map)' has changed its type to org.springframework.util.MultiValueMap [7005]

- **org.springframework.web.multipart.support.DefaultMultipartHttpServletRequest**

Parameter 2 of 'public DefaultMultipartHttpServletRequest(javax.servlet.http.HttpServletRequest, java.util.Map, java.util.Map)' has changed its type to org.springframework.util.MultiValueMap [7005]

web.portlet

- **org.springframework.web.portlet.DispatcherPortlet**

Parameter 3 of 'protected void doRender(org.springframework.web.servlet.View, java.util.Map, javax.portlet.RenderRequest, javax.portlet.RenderResponse)' has changed its type to javax.portlet.PortletRequest [7005]

Parameter 4 of 'protected void doRender(org.springframework.web.servlet.View, java.util.Map, javax.portlet.RenderRequest, javax.portlet.RenderResponse)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 2 of 'protected void render(org.springframework.web.portlet.ModelAndView, javax.portlet.RenderRequest, javax.portlet.RenderResponse)' has changed its type to javax.portlet.PortletRequest [7005]

Parameter 3 of 'protected void render(org.springframework.web.portlet.ModelAndView, javax.portlet.RenderRequest, javax.portlet.RenderResponse)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 3 of 'protected org.springframework.web.servlet.View resolveViewName(java.lang.String, java.util.Map, javax.portlet.RenderRequest)' has changed its type to javax.portlet.PortletRequest [7005]

- **org.springframework.web.portlet.FrameworkPortlet**

Abstract method 'protected void doEventService(javax.portlet.EventRequest, javax.portlet.EventResponse)' has been added [7013]

Abstract method 'protected void doResourceService(javax.portlet.ResourceRequest, javax.portlet.ResourceResponse)' has been added [7013]

Method 'public java.lang.String[] getUserinfoUsernameAttributes()' has been removed [7002]

Method 'public boolean isPublishContext()' has been removed [7002]

Method 'public boolean isPublishEvents()' has been removed [7002]

- **org.springframework.web.portlet.HandlerAdapter**

Method 'public void handleEvent(javax.portlet.EventRequest, javax.portlet.EventResponse, java.lang.Object)' has been added to an interface [7012]

Method 'public org.springframework.web.portlet.ModelAndView handleResource(javax.portlet.ResourceRequest, javax.portlet.ResourceResponse, java.lang.Object)' has been added to an interface [7012]

- **org.springframework.web.portlet.HandlerExceptionResolver**

Method 'public org.springframework.web.portlet.ModelAndView resolveException(javax.portlet.ResourceRequest, javax.portlet.ResourceResponse, java.lang.Object, java.lang.Exception)' has been added to an interface [7012]

- **org.springframework.web.portlet.HandlerInterceptor**

Method 'public void afterEventCompletion(javax.portlet.EventRequest, javax.portlet.EventResponse, java.lang.Object, java.lang.Exception)' has been added to an interface [7012]

Method 'public void afterResourceCompletion(javax.portlet.ResourceRequest, javax.portlet.ResourceResponse, java.lang.Object, java.lang.Exception)' has been added to an interface [7012]

Method 'public void postHandleResource(javax.portlet.ResourceRequest, javax.portlet.ResourceResponse, java.lang.Object, org.springframework.web.portlet.ModelAndView)' has been added to an interface [7012]

Method 'public boolean preHandleEvent(javax.portlet.EventRequest, javax.portlet.EventResponse, java.lang.Object)' has been added to an interface [7012]

Method 'public boolean preHandleResource(javax.portlet.ResourceRequest, javax.portlet.ResourceResponse, java.lang.Object)' has been added to an interface [7012]

- **org.springframework.web.portlet.bind.MissingPortletRequestParameterException**

Method 'public java.lang.String getParameterName()' is now final [7014]

Method 'public java.lang.String getParameterType()' is now final [7014]

- **org.springframework.web.portlet.handler.AbstractMapBasedHandlerMapping\$PortletRequestMappingPredicate**

Method 'public void validate(javax.portlet.PortletRequest)' has been added to an interface [7012]

- **org.springframework.web.portlet.handler.PortletContentGenerator**

Parameter 1 of 'protected void applyCacheSeconds(javax.portlet.RenderResponse, int)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 1 of 'protected void cacheForSeconds(javax.portlet.RenderResponse, int)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 1 of 'protected void checkAndPrepare(javax.portlet.RenderRequest, javax.portlet.RenderResponse)' has changed its type to javax.portlet.PortletRequest [7005]

Parameter 2 of 'protected void checkAndPrepare(javax.portlet.RenderRequest, javax.portlet.RenderResponse)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 1 of 'protected void checkAndPrepare(javax.portlet.RenderRequest, javax.portlet.RenderResponse, int)' has changed its type to javax.portlet.PortletRequest [7005]

Parameter 2 of 'protected void checkAndPrepare(javax.portlet.RenderRequest, javax.portlet.RenderResponse, int)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 1 of 'protected void preventCaching(javax.portlet.RenderResponse)' has changed its type to javax.portlet.MimeResponse [7005]

- **org.springframework.web.portlet.handler.SimpleMappingExceptionResolver**

Removed field logger [6001]

Method 'protected java.lang.String buildLogMessage(java.lang.Exception, javax.portlet.RenderRequest)' has been removed [7002]

Parameter 2 of 'protected java.lang.String determineViewName(java.lang.Exception, javax.portlet.RenderRequest)' has changed its type to javax.portlet.PortletRequest [7005]

Parameter 1 of 'protected org.springframework.web.portlet.ModelAndView doResolveException(javax.portlet.RenderRequest, javax.portlet.RenderResponse, java.lang.Object, java.lang.Exception)' has changed its type to javax.portlet.PortletRequest [7005]

Parameter 2 of 'protected org.springframework.web.portlet.ModelAndView doResolveException(javax.portlet.RenderRequest, javax.portlet.RenderResponse, java.lang.Object, java.lang.Exception)' has changed its type to javax.portlet.MimeResponse [7005]

Parameter 3 of 'protected org.springframework.web.portlet.ModelAndView getModelAndView(java.lang.String, java.lang.Exception, javax.portlet.RenderRequest)' has changed its type to javax.portlet.PortletRequest [7005]

Method 'protected void logException(java.lang.Exception, javax.portlet.RenderRequest)' has been removed [7002]

Method 'protected boolean shouldApplyTo(javax.portlet.RenderRequest, java.lang.Object)' has been

removed [7002]

- **org.springframework.web.portlet.multipart.DefaultMultipartActionRequest**

Removed org.springframework.web.portlet.util.ActionRequestWrapper from the list of superclasses [5001]

Removed org.springframework.web.portlet.util.PortletRequestWrapper from the list of superclasses [5001]

Parameter 2 of 'public DefaultMultipartActionRequest(javax.portlet.ActionRequest, java.util.Map, java.util.Map)' has changed its type to org.springframework.util.MultiValueMap [7005]

Return type of method 'protected java.util.Map getMultipartFiles()' has been changed to org.springframework.util.MultiValueMap [7006]

Parameter 1 of 'protected void setMultipartFiles(java.util.Map)' has changed its type to org.springframework.util.MultiValueMap [7005]

- **org.springframework.web.portlet.mvc.PortletWrappingController**

Parameter 1 of 'public void setInitParameters(java.util.Properties)' has changed its type to java.util.Map [7005]

- **org.springframework.web.portlet.util.ActionRequestWrapper**

Class org.springframework.web.portlet.util.ActionRequestWrapper removed [8001]

- **org.springframework.web.portlet.util.PortletRequestWrapper**

Class org.springframework.web.portlet.util.PortletRequestWrapper removed [8001]

web.servlet

- **org.springframework.web.servlet.handler.AbstractUrlHandlerMapping**

In method 'protected java.lang.Object buildPathExposingHandler(java.lang.Object, java.lang.String)' the number of arguments has changed [7004]

In method 'protected void exposePathWithinMapping(java.lang.String, javax.servlet.http.HttpServletRequest)' the number of arguments has changed [7004]

- **org.springframework.web.servlet.handler.SimpleMappingExceptionResolver**

Removed field logger [6001]

- **org.springframework.web.servlet.handler.metadata.AbstractPathMapHandlerMapping**

Class org.springframework.web.servlet.handler.metadata.AbstractPathMapHandlerMapping removed [8001]

- **org.springframework.web.servlet.handler.metadata.CommonsPathMapHandlerMapping**

Class org.springframework.web.servlet.handler.metadata.CommonsPathMapHandlerMapping removed [8001]

- **org.springframework.web.servlet.handler.metadata.PathMap**

Class org.springframework.web.servlet.handler.metadata.PathMap removed [8001]

- **org.springframework.web.servlet.handler.metadata.PathMap\$__attributeRepository**

Class org.springframework.web.servlet.handler.metadata.PathMap\$__attributeRepository removed [8001]

- **org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter**

Method 'protected org.springframework.web.servlet.ModelAndView handleNoSuchRequestHandlingMethod(org.springframework.web.servlet.mvc.multiaction.NoSuchRequestHandlingMethod javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)' has been removed [7002]

- **org.springframework.web.servlet.mvc.multiaction.MultiActionController**

Method 'protected void initBinder(javax.servlet.ServletRequest, org.springframework.web.bind.ServletRequestDataBinder)' has been removed [7002]

- **org.springframework.web.servlet.mvc.throwaway.ThrowawayController**

Class org.springframework.web.servlet.mvc.throwaway.ThrowawayController removed [8001]

- **org.springframework.web.servlet.mvc.throwaway.ThrowawayControllerHandlerAdapter**

Class org.springframework.web.servlet.mvc.throwaway.ThrowawayControllerHandlerAdapter removed [8001]

- **org.springframework.web.servlet.support.RequestContext**

In method 'public RequestContext(javax.servlet.http.HttpServletRequest, javax.servlet.ServletContext, java.util.Map)' the number of arguments has changed [7004]

In method 'protected void initContext(javax.servlet.http.HttpServletRequest, javax.servlet.ServletContext, java.util.Map)' the number of arguments has changed [7004]

- **org.springframework.web.servlet.view.AbstractView**

In method 'protected org.springframework.web.servlet.support.RequestContext createRequestContext(javax.servlet.http.HttpServletRequest, java.util.Map)' the number of arguments has changed [7004]

- **org.springframework.web.servlet.view.RedirectView**

Parameter 1 of 'protected void appendQueryProperties(java.lang.StringBuffer, java.util.Map, java.lang.String)' has changed its type to java.lang.StringBuilder [7005]

- **org.springframework.web.servlet.view.freemarker.FreeMarkerView**

Method 'protected void checkTemplate()' has been removed [7002]

Parameter 2 of 'protected void processTemplate(freemarker.template.Template, java.util.Map, javax.servlet.http.HttpServletResponse)' has changed its type to freemarker.template.SimpleHash [7005]

- **org.springframework.web.servlet.view.jasperreports.AbstractJasperReportsSingleFormatView**

Method 'protected java.util.Map mergeExporterParameters(java.util.Map)' has been removed [7002]

- **org.springframework.web.servlet.view.jasperreports.AbstractJasperReportsView**

Method 'protected net.sf.jasperreports.engine.design.JRCompiler getReportCompiler()' has been removed [7002]

Method 'public void setReportCompiler(net.sf.jasperreports.engine.design.JRCompiler)' has been removed [7002]

- **org.springframework.web.servlet.view.jasperreports.JasperReportsMultiFormatView**

Parameter 1 of 'public void setFormatMappings(java.util.Properties)' has changed its type to java.util.Map [7005]

- **org.springframework.web.servlet.view.jasperreports.JasperReportsViewResolver**

Method 'public void setReportCompiler(net.sf.jasperreports.engine.design.JRCompiler)' has been removed [7002]

- **org.springframework.web.servlet.view.tiles2.TilesConfigurer**

Method 'protected org.apache.tiles.TilesContainer createTilesContainer(javax.servlet.ServletContext)' has been removed [7002]

- **org.springframework.web.servlet.view.velocity.VelocityLayoutView**

Method 'protected void checkTemplate()' has been removed [7002]

- **org.springframework.web.servlet.view.velocity.VelocityView**

Method 'protected void checkTemplate()' has been removed [7002]

Parameter 1 of 'public void setToolAttributes(java.util.Properties)' has changed its type to java.util.Map [7005]

Method 'public void setVelocityFormatterAttribute(java.lang.String)' has been removed [7002]

- **org.springframework.web.servlet.view.velocity.VelocityViewResolver**

Method 'public void setVelocityFormatterAttribute(java.lang.String)' has been removed [7002]

- **org.springframework.web.servlet.view.xslt.AbstractXsltView**

Method 'protected java.util.Map getParameters()' has been removed [7002]

web.util

- **org.springframework.web.util.CookieGenerator**

Return type of method 'public int getCookieMaxAge()' has been changed to java.lang.Integer [7006]

Parameter 1 of 'public void setCookieMaxAge(int)' has changed its type to java.lang.Integer [7005]

- **org.springframework.web.util.ExpressionEvaluationUtils**

Field EXPRESSION_CACHE_CONTEXT_PARAM has been removed, but it was previously a constant [6011]

- **org.springframework.web.util.Log4jConfigServlet**

Class org.springframework.web.util.Log4jConfigServlet removed [8001]

- **org.springframework.web.util.UrlPathHelper**

Field INCLUDE_CONTEXT_PATH_REQUEST_ATTRIBUTE has been removed, but it was previously a constant [6011]

Field INCLUDE_SERVLET_PATH_REQUEST_ATTRIBUTE has been removed, but it was previously a constant [6011]

Field INCLUDE_URI_REQUEST_ATTRIBUTE has been removed, but it was previously a constant [6011]