Title: Maven spring boot run debug with arguments

Post Body:

Usually I'm running my Spring Boot application with command:

```
mvn spring-boot:run -Drun.arguments=--server.port=9090 \    -Dpath.to.config.dir=/var/data/my/config/dir
```

I want to set custom port to debug, so I can connect from eclipse. When I add arguments from example Spring Boot Maven Plugin

```
mvn spring-boot:run -Drun.arguments=--server.port=9090 \    -Dpath.to.config.dir=/var/data/my/config/dir \    -Drun.jvmArgumen
```

it works but other arguments like `server.port` or `path.to.config.dir` are no longer recognized and I get exception like:

```
org.springframework.beans.factory.BeanDefinitionStoreException: Failed to parse configuration class [com.my.app.Controller]; n
```

Question: How I can run with all arguments?

Accepted Answer:

The behavior and the change you notice is happening because you started using the `jvmArguments` option:

> JVM arguments that should be associated with the forked process used to run the application. On command line, make sure to wrap multiple values between quotes.

By default, when using it, the Spring Boot Maven plugin will also fork its execution, as described by the `fork` option:

> Flag to indicate if the run processes should be forked. By default process forking is only used if an agent or `jvmArguments` are specified.

Hence, the usage of `jvmArguments` also activated the fork mode of the plugin execution. By forking, you are actually not picking up the others `-D` arguments passed from command line.

Solution: if you want to use the `jvmArguments`, then pass all of the required arguments to it.

```
mvn spring-boot:run -Drun.jvmArguments="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,address=8787 -Dserver.port=9090 -Dpath.
```

-- Edit 22/09/2020 Check also the other answer from @Stephane to complent this answer (prefix of parameters)

Highest Rated Answer:

The parameter name has to be prefixed with `spring-boot.` as in `-Dspring-boot.run.jvmArgument`

The *Spring Boot Reference Documentation*, in the Debug the Application section, provided me the solution as I'm running `Spring Boot 2.0.3`

```
mvn spring-boot:run -Dspring-boot.run.jvmArguments="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=5005"
```