

# Managing File Sets

---

Spring configuration file sets, or just *file sets*, for short, are a way of grouping Spring configuration files in IntelliJ IDEA. They are intended for making your work with *mutually related* configuration files more effective.

IntelliJ IDEA lets you create the file sets that contain XML configuration files or `@Configuration`-annotated Java classes.

- [About file sets](#)
- [Parent-child relationships between the file sets](#)
- [Creating a file set](#)
- [Editing a file set](#)
- [Removing a file set](#)

## About file sets

By combining Spring configuration files in a file set you tell IntelliJ IDEA that these files are mutually related. In this way you form a common editing context for these configuration files and get all the associated coding assistance.

To illustrate, let us consider an imaginary XML configuration file `config-one.xml` containing two bean definitions:

```
...
<bean id="beanOne" class="BeanOne">
  ...
</bean>
<bean id="beanTwo" class="BeanTwo">
  <constructor-arg ref="beanOne"/>
  ...
</bean>
...
```

As you can see, the second of the beans (`beanTwo`) references the first one (`beanOne`). Obviously, there is no problem with this reference while both bean definitions are in the same file.

Now let us assume that you want to move the definition of `beanTwo` to a separate configuration file, say, `config-two.xml`. In this case, you should somehow indicate that `ref="beanOne"` in `config-two.xml` is a reference to `beanOne` defined in `config-one.xml`. In IntelliJ IDEA, you can do that by combining `config-one.xml` and `config-two.xml` in a file set.

In a similar way, you can form file sets that contain `@Configuration`-annotated Java classes.

You can create as many file sets as you need; any configuration file may be included in more than one file set. The configuration files that are external to your project can also be included in your file sets.

On the other hand, you don't have to create the file sets if you don't need them.

In certain cases, IntelliJ IDEA can create the file sets automatically. This is the case for Spring Web applications when the corresponding configuration files are specified in `web.xml` or have recognizable file names (for example, `<servlet-name>-servlet.xml`).

## Parent-child relationships between the file sets

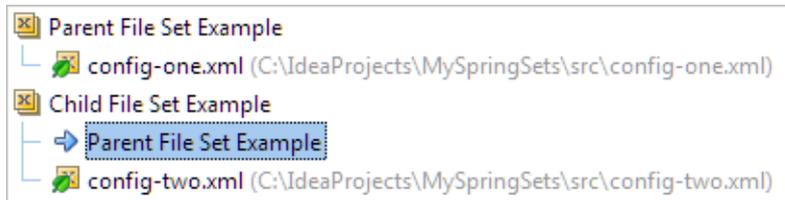
For each file set that you create, you can specify a parent file set. This lets you build file set hierarchies and thus control bean visibilities at the stage of developing your application.

This IntelliJ IDEA's feature fully mirrors the concept of hierarchical application contexts existing in the Spring framework where it's possible to use common beans from parent contexts, and inherit and overload the beans in child contexts.

The beans defined in a parent file set (or, to be more exact, in one of the configuration files belonging to a parent file set) are all visible in its child file sets (and all the descendent file sets). The beans defined in a child file set, however, are not visible in its parent file set.

Going back to the example we have discussed [earlier](#), you could place `config-two.xml` in a file set which is a child of a file set containing `config-one.xml`. In such a case, the reference `ref="beanOne"` would still be valid (as a reference to a bean defined in a parent file set). On the other hand, the beans in `config-one.xml` could not reference the beans defined in `config-two.xml`.

In the user interface of IntelliJ IDEA, parent file sets have a blue arrow next to them (⇒).



### To create a file set

1. [Open the Spring facet settings](#).
2. On the [Spring Facet page](#), click **+** (Alt+Insert) and select one of the following options:
  - **XML**. Select this option to create a file set containing XML configuration files.
  - **Annotation Config**. Select this option to create a file set containing `@Configuration`-annotated Java classes.
3. In the [Edit Application Context dialog](#):
  1. Specify the name of the file set in the **Name** field.
  2. If necessary, select a [parent file set](#) from the **Parent fileset** list.
  3. Select the configuration files to be included in the file set by selecting the corresponding check boxes.
  4. If you want to add a configuration file external to your module or project, click **Locate File** and select the necessary configuration file in the [Select Path dialog](#).
  5. Click **OK**.
4. Click **OK** in the **Project Structure** dialog.

You can initiate creation of a file set in the editor, if an XML configuration file which is not included in any of the file sets is currently open (see [Adding an XML configuration file to a file set](#)).

### To edit a file set

1. [Open the Spring facet settings](#).
2. If you just want to remove unnecessary configuration files from a file set, select the corresponding configuration files, click **-** (Alt+Delete), and then click **OK**. Note that this does not delete the configuration files physically but just removes them from the corresponding file set.  
Otherwise, select the necessary file set and click **↵** (Enter).
3. [Make the necessary changes](#) in the [Edit Application Context dialog](#).
4. Click **OK** in the **Project Structure** dialog.

## To remove a file set

1. [Open the Spring facet settings.](#)
2. Select the file set to be removed, click **—** (Alt+Delete), and then click **OK**. Note that this operation does not delete the corresponding configuration files.

## See Also

### Procedures:

- [Changing Facet Settings](#)

### Language and Framework-Specific Guidelines:

- [Spring](#)
- [Managing Spring Configuration Files](#)

### Reference:

- [Spring Facet Page](#)
- [Edit Application Context Dialog](#)
- [Spring Dependencies Diagram](#)

### External Links:

- [AnnotationConfigApplicationContext support in IntelliJ IDEA 11.1](#) 

### Web Resources:

- [Developer Community](#) 