

# No CustomWebBuildParticipant in IDEA 9

---



Frank Wienberg 22 posts since

Oct 2, 2002

Hi,

for my IDEA 8 plugin, I use `com.intellij.javaee.web.make.CustomWebBuildParticipant`, which is part of `plugins/JavaEE/lib/javaee-openapi.jar`. Despite of the JAR's name ("openapi"), this class disappeared in IDEA 9.

My use case is that for dependencies to certain libraries, certain additional files have to be copied into the Web app, similar to copying JARs to `WEB-INF/lib`. I simply had to extend `CustomWebBuildParticipant`, declare my subclass in `plugin.xml`, and implement `registerBuildInstructions()`, like so:

plugin.xml:

```
<extensions defaultExtensionNs="com.intellij"> <javaee.web.customBuildParticipant
implementation="net.jangaroo.ide.idea.JangarooWebBuildParticipant"/></extensions>
```

JangarooWebBuildParticipant.java:

...

```
public class JangarooWebBuildParticipant extends
com.intellij.javaee.web.make.CustomWebBuildParticipant {
```

```
    public void registerBuildInstructions(WebFacet webFacet, BuildRecipe buildRecipe,
CompileContext compileContext) {
```

No CustomWebBuildParticipant in IDEA 9

```
...  
}  
}
```

As an attachment, find the complete Java code of that class.

How can I achieve the same effect in IDEA 9? Any help would be appreciated!

Greetings,

Frank

**Attachments:**

- [JangarooWebBuildParticipant.java.zip](#) (1.2 K)

Tags: build, plugin, diana, web, api, maia



[Yann Cebron](#) 1,289 posts since

Aug 22, 2002 **1. Re: No CustomWebBuildParticipant in IDEA 9** Jan 9, 2010 9:05 PM

blind guess:

`com.intellij.javaee.web.artifact.WebArtifactUtil`



Guest **2. Re: No CustomWebBuildParticipant in IDEA 9** Jan 11, 2010 3:59 PM

Hello,

exploded/war configurations have been moved from facets to artifacts so it wasn't possible to keep old API. In order to achieve the same effect in IDEA

9 you can add `DirectoryCopyPackagingElement` to an artifact output layout when your facet is created. Use the following methods to add directory copy

No CustomWebBuildParticipant in IDEA 9

to an artifact layout:

```
ArtifactManager.getInstance(project).addElementToDirectory(artifact, relativeOutputPath,  
PackagingElementFactory.getInstance().createDirectoryCopyWithParentDirectories(pathToDirectory,  
""));
```

To obtain all artifacts which contains some web facet use method

```
JavaeeArtifactUtil.getInstance().getArtifactsContainingFacet(webFacet, null)
```

Hi,

for my IDEA 8 plugin, I use `com.intellij.javaee.web.make.CustomWebBuildParticipant`, which is part of `plugins/JavaEE/lib/javaee-openapi.jar`. Despite of the JAR's name ("openapi"), this class disappeared in IDEA 9.

My use case is that for dependencies to certain libraries, certain additional files have to be copied into the Web app, similar to copying JARs to `WEB-INF/lib`. I simply had to extend `CustomWebBuildParticipant`, declare my subclass in `plugin.xml`, and implement `registerBuildInstructions()`, like so:

plugin.xml:

```
<extensions defaultExtensionNs="com.intellij">  
  
  <javaee.web.customBuildParticipant  
implementation="net.jangaroo.ide.idea.JangarooWebBuildParticipant"/>  
  
</extensions>
```

JangarooWebBuildParticipant.java:

..

```
public class JangarooWebBuildParticipant extends  
com.intellij.javaee.web.make.CustomWebBuildParticipant {
```

```
    public void registerBuildInstructions(WebFacet webFacet, BuildRecipe  
buildRecipe, CompileContext compileContext) {
```

```
        ...
```

```
    }
```

```
}
```

As an attachment, find the complete Java code of that class.

How can I achieve the same effect in IDEA 9? Any help would be appreciated!

Greetings,

Frank

---

Original message URL: <http://www.jetbrains.net/devnet/message/5253633#5253633>

--

Nikolay Chashnikov

Software Developer

JetBrains, Inc

<http://www.jetbrains.com>

"Develop with pleasure!"



Frank Wienberg *22 posts since*

Oct 2, 2002 3. Re: No CustomWebBuildParticipant in IDEA 9 Jan 11, 2010 4:30 PM

Thanks, that helps a lot!

I already stumbled upon "artefacts" and was successful in extending `com.intellij.openapi.roots.ui.configuration.artifacts.sourceItems.FacetBasedPackagingSourceItemsProvider`. The only thing that is missing now is that my `MavenImporter` adds my custom compile output automatically to the corresponding Web artefacts. I guess your pointers should make it easy implementing this logic!