

Resolvendo conflitos entre dependências

Downloads

Caso queira usar o projeto imediatamente a partir desse vídeo, pode baixar o projeto [aqui](http://s3.amazonaws.com/caelum-online-public/PM-77/ivy-cap3-importar.zip) (<http://s3.amazonaws.com/caelum-online-public/PM-77/ivy-cap3-importar.zip>). O projeto deve ser importado no Eclipse.

Lidando com conflitos

Neste capítulo gerenciaremos duas dependências principais. A primeira, o Hibernate e a segunda uma implementação do JSF da Apache, o MyFaces. É claro que essas duas dependências terão dependências transitivas. Veremos como gerenciar essas dependências transitivas e o que fazer quando houver um conflito entre essas duas dependências.

Temos no nosso projeto inicial preparado no Eclipse com um `build.xml` e com um `ivy.xml`. Dentro do `build.xml` temos um target para limpar a pasta `lib` com nossas dependências e um target para atualizar essas dependências que usa a tarefa `ivy:retrieve`.

```

<project name="agenda" xmlns:ivy="antlib:org.apache.ivy.ant" default="atualiza-dependencias">

    <path id="ivy.lib.path">
        <fileset dir="ivy-lib" includes="*.jar"/>
    </path>

    <taskdef resource="org/apache/ivy/ant/antlib.xml" uri="antlib:org.apache.ivy.ant" classpathi

    <target name="limpar" >
        <delete dir="lib" />
        <mkdir dir="lib" />
        <mkdir dir="report" />
    </target>

    <target name="atualiza-dependencias" depends="limpar">
        <ivy:retrieve />
    </target>
</project>

```

`ivy:retrieve` realmente carregará as dependências, por isso já temos a tarefa declarada através do `taskdef`, além do caminho para encontrar a biblioteca do Ivy: o `ivy.jar`.

No `ivy.xml` temos o elemento `info` e o elemento `dependencies`. Vamos declarar a dependência do Hibernate, procurando no repositório do Maven por Hibernate. Usaremos a dependência o `hibernate-annotation` que por sua vez depende do `hibernate-core`:

```
<dependency org="org.hibernate" name="hibernate-annotations" rev="3.5.6-Final"/>
```

Vamos copiar então essa dependência colando-a dentro do nosso `ivy.xml`. Só garantindo através da configuração que apenas os JARs principais serão carregados, usando o atributo `conf="default"`:

```

<ivy-module version="2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://ant.apache.org/ivy/schemas/ivy.xsd">

  <info module="agenda" organisation="br.com.caelum" />

  <dependencies>
    <dependency org="org.hibernate" name="hibernate-annotations" rev="3.5.6-Final" conf="de-"
  </dependencies>

</ivy-module>

```

Testando a configuração na linha de comando, executando ANT chamando o target `atualiza-tarefas`, foram carregadas as dependências do `hibernate-annotation` e as dependências transitivas.

No relatório podemos ver que foram carregados 10 artifacts:

`atualiza-dependencias:`

```

.....
[ivy:retrieve]     confs: [default]
[ivy:retrieve]     found org.hibernate#hibernate-annotations;3.5.6-Final in public
[ivy:retrieve]     found org.hibernate#hibernate-core;3.5.6-Final in public
[ivy:retrieve]     found antlr#antlr;2.7.6 in public
[ivy:retrieve]     found commons-collections#commons-collections;3.1 in public
[ivy:retrieve]     found dom4j#dom4j;1.6.1 in public
[ivy:retrieve]     found xml-apis#xml-apis;1.0.b2 in public
[ivy:retrieve]     found javax.transaction#jta;1.1 in public
[ivy:retrieve]     found org.slf4j#slf4j-api;1.5.8 in public
[ivy:retrieve]     found org.hibernate#hibernate-commons-annotations;3.2.0.Final in public
[ivy:retrieve]     found org.hibernate.javax.persistence#hibernate-jpa-2.0-api;1.0.0.Final in pi
[ivy:retrieve]     :: resolution report :: resolve 418ms :: artifacts dl 12ms
-----
|           |           modules      ||   artifacts   |
|   conf    |   number| search|dwnlded|evicted||   number|dwnlded|
-----
|   default  |   10   |   0   |   0   |   0   ||   10   |   0   |
-----
[ivy:retrieve] :: retrieving :: br.com.caelum#agenda
[ivy:retrieve]     confs: [default]
[ivy:retrieve]     10 artifacts copied, 0 already retrieved (4461kB/18ms)
...

```

Voltando no projeto e atualizando-o, aparece a pasta `lib` com os JARs carregados, os JARs principais e também os transitivos.

Desabilitando dependências transitivas

O tag `dependency` do IVY possui elemento `transitive` que por padrão é `true`, mas mudaremos para `false` para carregar apenas a dependência declarada, o `hibernate-annotation` na versão 3.5.6.

atualiza-dependencias:

```
....  
[ivy:retrieve]     confs: [default]  
[ivy:retrieve]     found org.hibernate#hibernate-annotations;3.5.6-Final in public  
[ivy:retrieve] :: resolution report :: resolve 111ms :: artifacts dl 3ms  
-----  
|           |       modules       ||   artifacts   |  
|   conf    |   number| search|dwnlded|evicted||   number|dwnlded|  
-----  
| default  |   1   |   0   |   0   |   0   ||   1   |   0   |  
-----  
[ivy:retrieve] :: retrieving :: br.com.caelum#agenda  
[ivy:retrieve]     confs: [default]  
[ivy:retrieve]     1 artifacts copied, 0 already retrieved (356kB/5ms)
```

Para continuar vamos deixar o elemento `transitive="true"` .

Mais dependências e o primeiro conflito

Em nossa aplicação queremos utilizar o Hibernate junto com JSF, escolhemos a implementação MyFaces. Então vamos procurar pelo MyFaces no repositório do Maven e copiar a declaração da dependência do Ivy, colando-a dentro do elemento `dependencies` , garantindo também que será usada a configuração padrão apenas (`conf="default"`):

```
<dependency org="org.apache.myfaces.core" name="myfaces-impl" rev="2.1.7" conf="default"/>
```

Testando novamente o `build.xml` . Chamando `ant atualiza-dependencias` mostra no relatório que 15 artifacts foram usadas, mas 16 módulos foram carregados.

```
[ivy:retrieve] :: evicted modules:  
[ivy:retrieve] commons-collections#commons-collections;3.1 by [commons-collections#commons-collections;3.2]  
-----  
|           |       modules       ||   artifacts   |  
|   conf    |   number| search|dwnlded|evicted||   number|dwnlded|  
-----  
| default  |   16  |   0   |   0   |   1   ||   15  |   0   |  
-----
```

Podemos ver que um módulo foi tirado (evicted). Um pouco mais para cima é mostrado qual dependência não foi usada, nesse caso a `commons-collections` na versão 3.1. Olhando por dentro da pasta `lib` , percebemos que o `commons-collections` na versão 3.2 foi o escolhido, ou seja a versão mais recente.

Gerenciador de conflitos

Existe um gerenciador (conflict manager) que decide quais são os JAR a utilizar, mas podemos decidir qual gerenciador utilizar. O conflict manager padrão se chama `lastest-revision` , ou seja como já vimos, usa a versão mais nova, e por isso foi utilizado o `commons-collections` na versão 3.2 . O gerenciador é definido no `ivy.xml` :

```
<conflicts>
  <manager name="latest-revision" />
</conflicts>
```

Mas podemos definir um outro conflict manager, por exemplo, `all`. Vamos testar o conflict manager `all` executando novamente ANT.

Dentro do `ivy.xml`:

```
<conflicts>
  <manager name="all" />
</conflicts>
```

Testando ant atualiza-dependencias :

```
[ivy:retrieve] :: resolution report :: resolve 588ms :: artifacts dl 22ms
-----
|           |           modules      ||  artifacts  |
|   conf     |   number| search|dwnlded|evicted||  number|dwnlded|
-----
|   default  |   16   |   0   |   0   |   0   ||   16   |   0   |
-----
[ivy:retrieve] :: retrieving :: br.com.caelum#agenda
[ivy:retrieve]   confs: [default]
[ivy:retrieve]   16 artifacts copied, 0 already retrieved (7427kB/210ms)
```

Podemos ver no relatório que foram 16 módulos carregados e 16 módulos adicionados. Ou seja, foram carregadas as duas versões do `commons-collections`, a versão 3.1 E a versão 3.2.

E também temos um conflict manager que se chama `strict`. Se algum conflito ocorrer, nada é carregado e o build falha. No `ivy.xml`:

```
<conflicts>
  <manager name="strict" />
</conflicts>
```

```
[ivy:retrieve] :: problems summary :: 
[ivy:retrieve] ::::: ERRORS
[ivy:retrieve]   commons-collections#commons-collections;3.2 (needed by [org.apache.myfaces.co
[ivy:retrieve]
[ivy:retrieve] :: USE VERBOSE OR DEBUG MESSAGE LEVEL FOR MORE DETAILS

BUILD FAILED
```

Vamos continuar usando o `lastest-revision`.

Criação do relatório de dependências

Existe uma outra forma de visualizar as dependências. O Ivy vem uma tarefa já integrada para gerar confortavelmente um relatório. Ela se chama `ivy:report` e recebe um atributo que define o nome da pasta do relatório. Vamos gerar um relatório com Ivy na pasta `report`:

```
<ivy:report todir="report"/>
```

O target fica como:

```
<target name="atualiza-dependencias" depends="limpar">
  <ivy:retrieve />
  <ivy:report todir="report"/>
</target>
```

Executando ANT novamente gerar então um relatório umas páginas html. Abrindo a página html temos o status das dependências, os conflitos e um resumo de todas as dependências. No nosso relatório temos duas dependências principais: o MyFaces e o Hibernate-annotation e as dependências transitivas de cada um. Depois uma listagem de cada dependência principal e transitiva.

Dependencies Stats							
Modules	15	Revisions	16 (0 searched 🔍, 0 downloaded 📥, 0 evicted 🗑, 0 errors 🚨)	Artifacts	16 (0 downloaded, 0 failed)	Artifacts size	7428 kB (0 kB downloaded, 7428 kB in cache)
Conflicts							
Module	Selected	Evicted					
commons-collections by commons-collections	3.2.3.1						
Dependencies Overview							
Module	Revision	Status	Resolver	Default	Licenses	Size	
myfaces-impl by org.apache.myfaces.core	2.1.7	release	public	false	1936 kB		
--- commons-digester by commons-digester	1.8	release	public	false	The Apache Software License, Version 2.0	140 kB	
----- commons-beanutils by commons-beanutils	1.8.3	release	public	false		227 kB	
----- commons-logging by commons-logging	1.1.1	release	default	false		59 kB	
----- commons-logging by commons-logging	1.1.1	release	default	false		59 kB	
--- commons-beanutils by commons-beanutils	1.8.3	release	public	false		227 kB	
----- commons-logging by commons-logging	1.1.1	release	default	false		59 kB	
--- commons-codec by commons-codec	1.3	release	public	false	The Apache Software License, Version 2.0	46 kB	
--- commons-collections by commons-collections	3.2	release	default	false	The Apache Software License, Version 2.0	558 kB	
hibernate-annotations by org.hibernate	3.5.6-Final	release	public	false		357 kB	
--- hibernate-jpa-2.0-api by org.hibernate.javax.persistence	1.0.0.Final	release	public	false	Unknown License	99 kB	
--- hibernate-commons-annotations by org.hibernate	3.2.0.Final	release	public	false	GNU LESSER GENERAL PUBLIC LICENSE	70 kB	
----- slf4j-api by org.slf4j	1.5.8	release	public	false		23 kB	
----- hibernate-core by org.hibernate	3.5.6-Final	release	public	false		2507 kB	

Exclusão de dependências

Pode ainda, para cada dependência, solicitar a exclusão de alguma biblioteca para que possamos adicionar a versão que desejamos. Para isso, em nosso exemplo, utilizaremos tag `exclude` colocando o nome da biblioteca que queremos excluir, neste caso, vamos excluir `commons-collections`. Para a dependência `hibernate-annotations` e `myfaces`. No `ivy.xml`:

```
<dependencies>

  <dependency org="org.hibernate" name="hibernate-annotations" rev="3.5.6-Final" transitive="1"
    <exclude name="commons-collections" />
  </dependency>

  <dependency org="org.apache.myfaces.core" name="myfaces-impl" rev="2.1.6" conf="default">
    <exclude name="commons-collections" />
  </dependency>
```

```
</dependency>
</dependencies>
```

Agora vamos deixar o dependência `commons-collections` explicita, mas na versão 3.0. Para que isso funcione, é necessário adicionar o elemento `force` com o valor `true` para permitir que o elemento seja carregado:

```
<dependencies>

<dependency org="commons-collections" name="commons-collections" force="true" rev="3.0" con-
<dependency org="org.hibernate" name="hibernate-annotations" rev="3.5.6-Final" transitive="i-
    <exclude name="commons-collections" />
</dependency>

<dependency org="org.apache.myfaces.core" name="myfaces-impl" rev="2.1.6" conf="default">
    <exclude name="commons-collections" />
</dependency>
</dependencies>
```

Testando novamente com Ant:

```
-----
|           |           modules           ||   artifacts   |
|   conf     |   number|   search|dwnlded|evicted||   number|dwnlded|
-----
|   default  |   17  |   0   |   0   |   2   ||   15  |   0   |
-----
[ivy:retrieve] :: retrieving :: br.com.caelum#agenda
[ivy:retrieve]     confs: [default]
[ivy:retrieve]     15 artifacts copied, 0 already retrieved (6833kB/273ms)
```

Podemos ver que também foram carregados 15 artefatos (de 17 modules) e o conflict manager escolheu a versão 3.0 do `commons-collections`.

Resumo

Nesse vídeo vimos como resolver conflitos entre dependências transitivas com ivy. Em nosso caso, ocorreu um conflito, pois as dependências do Hibernate e MyFaces, utilizavam a mesma dependência transitiva: a `commons-collections`, mas em versões diferentes.

No início da aula vimos como configurar o ivy para simplesmente ignorar as dependências transitivas, usando o attribute `transitive` com valor falso.

Outra opção para excluir uma dependência específica, é embutir uma tag `exclude`.

Caso exista um conflito, quem tentará resolvê-lo é o gerenciador. O padrão usado é o conflict manager `latested-revision`, que sempre utilizará a versão mais recente, porém, podemos escolher outros gerenciadores como `all` ou `strict`.

