

Reasons for using Ant Properties files over "Properties Tasks"

Asked 15 years, 11 months ago Modified 13 years, 4 months ago Viewed 8k times



4



I'm currently working with some developers who like to set up Ant tasks that define environment specific variables rather than using properties files. It seems they prefer to do this because it's easier to type:

```
ant <environment task> dist
```



Than it is to type:

```
ant -propertyfile <environment property file> dist
```

So for example:

```
<project name="whatever" default="dist">

<target name="local">
    <property name="webXml" value="WebContent/WEB-INF/web-local.xml"/>
</target>

<target name="remote">
    <property name="webXml" value="WebContent/WEB-INF/web-remote.xml"/>
</target>

<target name="build">
    <!-- build tasks here --->
</target>

<target name="dist" depends="build">
    <war destfile="/dist/foo.war" webxml="${webXml}">
        <!-- rest of war tasks here -->
    </war>
</target>
```

I am finding it hard to convince them that properties files are they right way to go. I believe properties files are better because:

- They provides more flexibility - if you need a new environment just add a new properties file
- It's clearer what's going on - You have to know about this little "trick" to realize what they're accomplishing

- Doesn't provide default values and the ability to use overrides - if they used property files they could provide defaults at the top of the project but have the ability to override them with a file
- Script won't break if an environment task isn't supplied on command line

Of course all they hear is that they need to change their Ant script and have to type more on the command line.

Can you provide any additional arguments in favor of properties files over "property tasks"?

ant properties build-process

Share Follow

asked Jan 14, 2009 at 17:55



Brian Kelly

5,644 ● 4 ● 29 ● 31

3 Answers

Sorted by: Highest score (default)



15



Properties tasks tightly couple the build file to environments. If your fellow developers are arguing that they "have to change their ant script" with your suggestions, why aren't they arguing about changing it every time they have to deploy to a new environment? :)

Perhaps you can convince them to allow both properties file and command-line configuration. I set up my Ant builds so that if a build.properties exists in the same directory as the build.xml, it reads it in. Otherwise it uses a set of default properties hard-coded into the build. This is very flexible.

```
<project name="example">

  <property file="build.properties"/>

  <property name="foo.property" value="foo"/>
  <property name="bar.property" value="bar"/>

  ...
</project>
```

I don't provide a build.properties with the project (i.e. build.properties is not versioned in SCM). This way developers aren't forced to use the property file. I do provide a build.properties.example file that developers can reference.

Since **Ant properties, once set, are immutable**, the build file will use properties defined in this order:

- Properties provided with `-D` or `-propertyfile` via the command line
- Properties loaded from `build.properties`
- Default properties within `build.xml`

Advantages of this approach:

- The build file is smaller and therefore more maintainable, less bug-prone
- Developers that just can't get away from setting properties at the command line can still use them.
- Properties files can be used, but aren't required

Share Follow

edited Aug 8, 2011 at 18:55

answered Jan 14, 2009 at 18:11



[Rob Hruska](#)

120k ● 32 ● 169 ● 192



3

The arguments you have are already pretty compelling. If those arguments haven't worked, then arguing isn't going to solve the problem. In fact, *nothing* is going to solve the problem. Don't assume that people are rational and will do the most practical thing. Their egos are involved.



Stop arguing. Even if you win, the resentment and irritation you create will not be worth it. Winning an argument can be worse than losing.



Make your case, then let it go. It's possible that after a while they will decide to switch to your way (because it actually is better). If that happens, they will act like it was their own idea. There will be no mention of your having proposed it.

On the other hand, they may never switch.

The only solution is to work towards a position of authority, where you can say how things are to be done.

Share Follow

answered Jan 14, 2009 at 18:23



[Ethan](#)

60k ● 65 ● 189 ● 241



0

The problem with the first solution (using ant property) is basically hardcoding. It can be convenient when you start a project for yourself but quickly you have to remove that bad habit.



I'm using a property file close to what said robhruska except that I have committed the build.properties file directly. This way you have a default one.



In other hand, I understand I could add those default values in the build.xml. (I will probably try that in the next hours/days ;-)).

Anyway, I really don't like the first approach and I would force those guys to follow the second one ...

Share Follow

answered Jan 14, 2009 at 20:29



Nicolas Dufour

203 ● 1 ● 5
