

# NewAntFeaturesInDetail PresetDef

[the task's manual page](#)

If you want a version of <ant> with the logical default of inheritall="false", here it is

```
<presetdef name="better-ant">
  <ant inheritall="false"/>
</presetdef>
```

wherever you use <better-ant>, it will behave exactly like the <ant> task but with a different default value.

You can also define child elements. Say you want all your <javac> tasks to compile against a set of jars living in /our/jar/repository/, you could define

```
<presetdef name="my-javac">
  <javac>
    <classpath>
      <fileset dir="/our/jar/repository/" includes="*.jar"/>
    </classpath>
  </javac>
</presetdef>
```

and use <my-javac> wherever you'd use <javac> instead.

It is interesting to note that both <presetdef> and <macrodef> dynamically define a task that can then be invoked as any other. Properly declared, the resulting tasks can be used interchangeably. Here is a simple example, in which the *filter* target will echo a message either to the console or to a file depending on whether the property *destfile* has been set:

```
<target name="-tocon" unless="destfile">
  <macrodef name="myecho">
    <attribute name="message" />
    <sequential>
      <echo taskname="myecho"
        message="Echoing a message to the console:" />
      <echo taskname="myecho" message="@{message}" />
    </sequential>
  </macrodef>
</target>

<target name="-tofile" if="destfile">
  <presetdef name="myecho">
    <echo file="{destfile}" />
  </presetdef>
</target>

<target name="filter" depends="-tocon,-tofile">

  <loadfile srcFile="{srcfile}" property="message">
    <filterchain refid="myfilter" />
  </loadfile>

  <myecho message="{message}" />
</target>
```

Another current use for <presetdef> is to provide a work-around for a bug with the JDK 1.5beta1 javac compiler. The compiler accepts target="1.1" but for some reason this does not work unless one sets source="1.3" as well. As this is a temporary issue one does not want to modify the build script much. This can be achieved as follows:

```
<available property="jdk1.5+" classname="java.util.concurrent.Callable"/>

<target name="check-1.5" if="jdk1.5+">
  <presetdef name="javac">
    <javac source="1.3"/>
  </presetdef>
</target>

<target name="init" depends="check-1.5">
  ....
</target>
```

This will make a new javac task which will call the current javac task with the source option set to 1.3. It will also whine that one is overwriting the javac task.