

NewAntFeaturesInDetail MacroDef

This task lets you define new tasks as macros based on other tasks.

[The manual page](#)

If you are using `<antcall>` as a macro substitute, you really should look into this task. It is not only going to simplify your build files but also speed up your builds considerably as you skip the huge overhead connected with `<antcall>`.

Let's take an example from Ant's own build file. Starting with Ant 1.6, we've split `optional.jar` into separate jars, one per library dependency. This led to

```
<jar destfile="${build.lib}/${optional.jars.prefix}-trax.jar"
      basedir="${build.classes}"
      manifest="${manifest.tmp}">
  <selector refid="needs.trax"/>
</jar>

<jar destfile="${build.lib}/${optional.jars.prefix}-xalan1.jar"
      basedir="${build.classes}"
      manifest="${manifest.tmp}">
  <selector refid="needs.xalan1"/>
</jar>
...
```

25 `<jar>`s that were identical in structure. With `<macrodef>` we now have

```
<macrodef name="optional-jar">
  <attribute name="dep"/>
  <sequential>
    <jar destfile="${build.lib}/${optional.jars.prefix}-${dep}.jar"
          basedir="${build.classes}"
          manifest="${manifest.tmp}">
      <selector refid="needs.${dep}" />
    </jar>
  </sequential>
</macrodef>

<optional-jar dep="trax"/>
<optional-jar dep="xalan1"/>
...
```

instead.

Starting a complete new Ant instance could be done with `<java>` task and the Launcher class. But the Launcher would exit the JVM. With the new `<permission>` datatype supported by `<java>` we can revoke that privilege. While trying that I realized that I have to grant some other privileges. Packed into a `<macrodef>` I got

```

<macrodef name="startAnt">
  <attribute name="options" default=""/>
  <sequential>
    <java classname="org.apache.tools.ant.launch.Launcher">
      <arg line="@{options}" />
      <permissions>
        <!-- grant all Ant needs (generously) but without VM exit -->
        <revoke class="java.lang.RuntimePermission" name="exitVM"/>
        <grant class="java.lang.RuntimePermission" name="*" />
        <grant class="java.lang.reflect.ReflectPermission" name="*" />
        <grant class="java.util.PropertyPermission" name="*" actions="read,write" />
        <grant class="java.io.FilePermission" name="*<<ALL FILES>>" actions="read,write,delete,
execute"/>
      </permissions>
    </java>
  </sequential>
</macrodef>

<!-- call another buildfile in quiet mode-->
<startAnt options="-f create.xml -quiet"/>

<!-- call the actual projecthelp -->
<startAnt options="-p"/>

<!-- call the actual default target -->
<startAnt/>

```

... mmh - be careful of endless loops 😊