FAQ

How do I get/set ant properties?

To load a properties file:

```
Ant.property(file: 'common-versions.properties')
```

To get a property:

```
Ant.property(name: 'ivy.local.repository.dir', value: ivyLocalRepoDir)
```

To set:

```
logkitVersion = Ant.project.properties.'logkit.version'
```

I'd like to use Bazaar to work with Gant - what are my first steps?

If Bazaar is not already installed on your system take a look at this link:

Download Bazaar

The official Gant repository is a Subversion repository so you will need to have bzr-svn installed to use Bazaar. You might want to take a look at this link on how to use it:

Use it

Alternatively, up to date Bazaar branches of Gant Trunk and Gant 1_1_X can be found at:

http://www.russel.org.uk/Bazaar/Gant_Trunk
http://www.russel.org.uk/Bazaar/Gant_1_1_X

These are Bazaar branches with no working tree so if you use a browser to view them they will appear empty. However if you branch from them using "bzr branch ...", you will get a branch.

How can I access whether the debug flag is set in my build.gant files?

```
import org.codehaus.gant.GantState

// We use the Gant State or "-d" flag
// to switch debugging on in our build.
// This closure allows us to query that property
isDebug = { ->
 if (GantState.verbosity == GantState.DEBUG) {
   return true
 }
}

if (isDebug()) {
  ant.echoproperties()
}
```

I'd like to use the uptodate task from Ant repeatedly in the same Gant.binding, would you provide a reusable approach?

Given that Ant uses immutable properties, the resulting property name will always need to be different otherwise you will end up returning the same calculated value over-and-over. Try this closure:

```
isUpToDate = { includeList, excludeList, jar ->
  resultPropName = "result." + System.currentTimeMillis()
  Ant.uptodate(property: "${resultPropName}", targetfile: jar) {
    includeList.each {
      srcfiles(dir: it)
    }
    excludeList.each {
      srcfiles(dir: it, excludes: "*")
    }
  }
  return Boolean.valueOf(Ant.project.properties."${resultPropName}")
}
```

How do I set my classpath when calling Gant?

You have two options:

1) when calling Gant, pass in a -P=classpath parameter:

```
gant -P ../build/classes:../build/lib/myjar.jar
```

2) pass in a -L parameter to add in class directories:

```
gant -L ../build/lib -L ../build/lib/ext -L ../build/out/classes
```

Can targets take parameters like other closures?

Yes. Although targets are usually specified like:

```
target ( name : 'description' ) { . . . }
```

where the closure comprising the body of the target appears to have no parameters, in fact there is an implicit parameter it. When called from the command line it always has the value null but when called from within a Gant script the target closure can be called with a parameter. For details see Targets and Parameters

Why do I get "Error evaluating Gantfile: No such property: env" when env should be available?

In the following example:

```
ant.property(environment:'env')

...

ant.echo("${env.CATALINA_HOME}")
```

You will get an error similar to:

build, line 11 -- Error evaluating Gantfile: No such property: env for class: build

This is a phasing issue between groovy's GString and Ant -- Groovy attempts to interpret the env.CATALINA_HOME as part of a GString rather than allowing the expression to be sent as is to the Ant project instance for expansion of properties.

Use single quotes instead (switches of GString interpretation) and it will work just fine.