

Frequently asked questions (FAQ)

(Last modified: December 17, 2004)

Question

When I run the HelloWorld example, I get the error message `java.lang.UnsupportedClassVersionError`. What can I do?

Answer

Please make sure that

1. you are running under Java SDK 1.4.0 or a later version.
2. you are using – as specified in the book – FreeTTS 1.2.0.

Question

In the class `AnimatedAudioOutput` shown on page 79 the super class `AudioOutput` cannot be resolved. Is there an `import` statement missing in the text?

Answer

No. The class `AudioOutput` is located in the same package `com.sun.speech.freetts.relp`. The classes of this package are contained in archive `freetts.jar`. If the class `AudioOutput` cannot be resolved, obviously the archive `freetts.jar` is not in the *Java Build Path* of project `DukeSpeaks`. This can have the following reason:

- The project `FreeTTS` was not added as a “required project” to the *Build Path* of project `DukeSpeaks`.

Question

I have successfully run the Hex example application within the Eclipse SDK. After exporting this application and installing it under the Eclipse RCP binary distribution and after installing

the additional plug-ins, I get the error message „No application id has been found“. What did I do wrong?

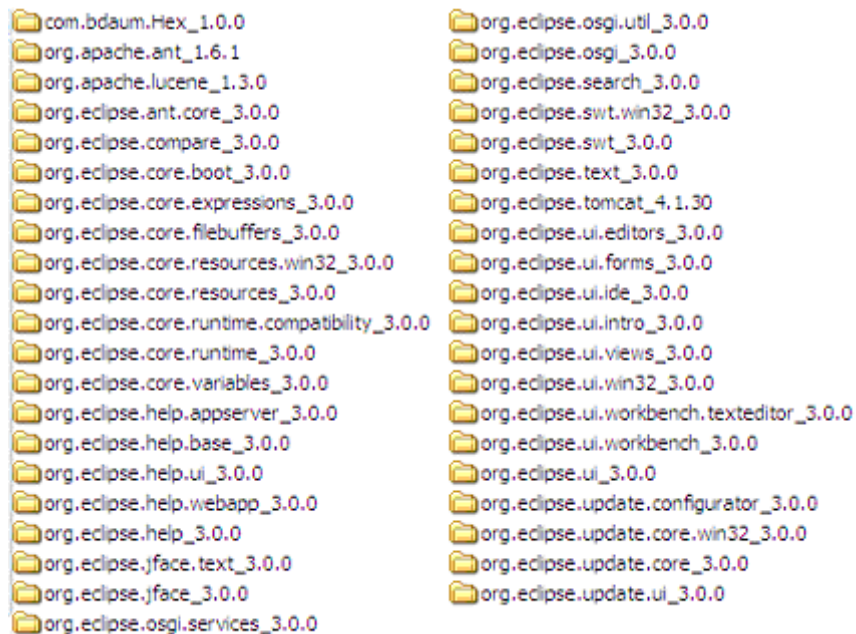
Answer

1. First check that the `config.ini` file was defined as specified in the book.
2. Since `config.ini` file did not specify a value for the parameter `osgi.bundles`, it's default value is used:

```
osgi.bundles=org.eclipse.core.runtime@2:start,org.eclipse.update.configurator@3:start.
```

In consequence, the plug-in `org.eclipse.update.configurator` must be present.

3. Unfortunately, there are some hidden dependencies between plug-ins in Eclipse 3.0 which can cause the necessity of installing additional plug-ins. Here is a plug-in selection under which the Hex application runs successfully.



4. After each configuration change (installing a new plug-in, manifest modification, or `config.ini` modification), it is necessary to delete all files in the configuration folder except `config.ini`. These files cache processed configuration data. By deleting these files you force Eclipse to create new updated configuration files at the next start.