

PRB: Compiled application cannot find resources (Excelsior JET 4.1 and below)

Article ID: 000014

Last Revised On: 25-Jul-2006

The information in this article applies to:

- Excelsior JET versions 2.5 through 4.1

This article **does not** apply to Excelsior JET 4.5 and above.

The similar article for Excelsior JET 4.5 and above can be found here [000030](#).

SYMPTOMS

The compiled application fails to load resource files (images, audio clips, etc.) For a GUI application this typically results in its window and controls displayed improperly or not displayed at all, and `NullPointerException` be thrown.

CAUSE

A Java application running on a conventional JRE typically searches for resource files along the *class path*. It is a list of directories and jars contained in the `java.class.path` property, which in turn may be set using the `-classpath (-cp)` option of the Java launcher (`java/javaw`), or the `CLASSPATH` environment variable.

A JET-compiled application loads resource files in exactly the same way. So you have to ensure that resource files will be available to the executable and that the `java.class.path` property will be set accordingly at run time.

RESOLUTION

The easiest way to make the directories and jars containing resource files listed in the `java.class.path` property without recompiling the application is by setting the `CLASSPATH` environment variable:

```
SET CLASSPATH=%CLASSPATH%;.\resources;C:\Res\A.jar
MyApp.exe
```

This will work on the developer's system.

When you prepare the application for deployment using the JetPackII tool, select resource files to be distributed along with the application executable on the **Resources** page. The installer will then hardwire the correct value for `java.class.path` into the executable during its installation.

The above approach involves distributing resources separately from the executable, as a set of jars and/or separate files. However, if resource files are packed into jars, you may consider using the *resource packing*¹ feature instead. Excelsior JET can pack jar files into the resulting executables, much like

¹In the Excelsior JET 3.7 and below this feature is called *resource binding*.

resources are included into native Windows programs. On executable launch, those “embedded” jars are added to the class path.

You set the desired resource packing mode on the **Resources** page of the JET Control Panel. In a new project, resource packing is enabled by default, i.e. any jar you add to the project on the **Classes** page, will automatically get listed on the **Resources** page. However, if you have any jars that contain no classes, just resource files, you will need to specify them manually. By default, class files are removed from jars as they are packed into the executable (the original jars are left intact).

When working with Excelsior JET command-line compiler, use the options **BINDRESOURCES** and **BINDCLASSES** to control resource packing:

Include resources only:

```
% MyApp.prj
. . .
-bindresources+    % Pack resources only,
-bindclasses-     % do not pack classes into the executable
. . .
!module MyApp.jar
!module resources.jar
```

Include the entire jars:

```
% MyApp.prj
. . .
-bindresources+
-bindclasses+
. . .
!module MyApp.jar
!module resources.jar
```

You may include resources into the resulting executable only if they are contained in the jars. There is no way to include separate resource files into executables.

STATUS

This behavior is by design. It is not possible to determine statically which resource files from which locations will be loaded by the application at run time and adjust the generated code accordingly.

REFERENCES

1. Excelsior JET User’s Guide (<http://www.excelsior-usa.com/doc/jet/jetw.html>), Chapter “Application Considerations”, section “Resource packing”.
2. Excelsior JET User’s Guide (<http://www.excelsior-usa.com/doc/jet/jetw.html>), Chapter “Deployment Automation”, section “Setting Up Standalone Resources and Java System Properties”.