

Empaquetando tu programa en un archivo JAR distribuible

Basado en una guía por Eric Roberts and Brandon Burr. Traducido por Jorge Esguerra, para CSBridge Uniandes.

Ahora que has escrito todos estos programas, ¿no sería grandioso poder distribuirlos a tus amigos y familiares para que puedan ver los frutos de todo tu esfuerzo?

Los archivos JAR (Java Archives) te permiten hacer eso justamente. A continuación, encontrarás una corta guía para crear JARs para tus programas desde eclipse.

Paso 1: Agregar un método main

Nuestros programas, hasta el momento, han usado las librerías de la ACM y de la configuración de Eclipse manejada por el curso CS 106A de la Universidad de Stanford. Se planteó que el método `public void run()` fuera el encargado de empezar a correr nuestros programas. Esto se hizo para que las cosas fueran más fáciles para ti. No obstante, un programa Java necesita un método particularmente importante desde donde arrancará su ejecución dentro de una clase determinada. Este método es: `public static void main(String[] args)`. Para poder exportar tus programas, necesitarás editar tu código para que incluya explícitamente el método `main()`. Después, tu programa correrá de manera adecuada en cualquier compilador de Java. La manera de realizar este cambio es la siguiente:

```
public static void main(String[] args) {  
    (new MyClass()).start(args);  
}
```

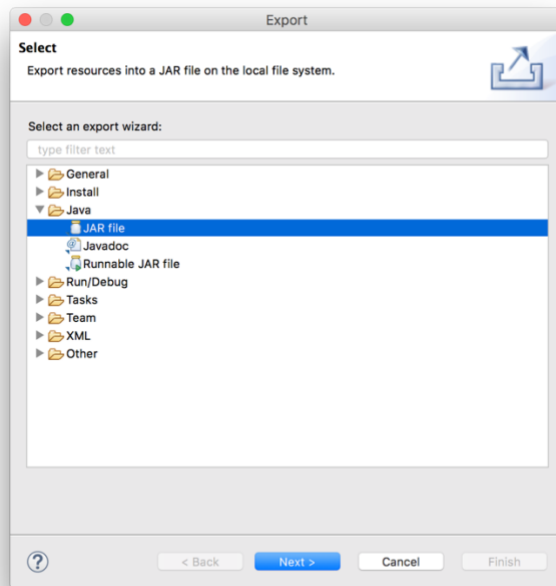
Nótese que este método deberá ir en la clase en donde anteriormente se había puesto el método `public void run()`, y debe sustituirse `MyClass` por el nombre de la clase en cuestión.

Paso 2: Exportando el programa en formato JAR.

Ahora que ya tenemos un programa Java normal, podemos empaquetarlo en un JAR. Un JAR es simplemente un *Java ARchive* – un archivo que contiene un conjunto de clases Java así como otros archivos que podrían ser necesarios para la ejecución del programa. Debemos exportar tanto un JAR como un archivo o fichero de manifiesto (**manifest file**). Este último permite especificar algunas reglas, como por ejemplo: la clase principal (**donde está "main()"**) que se ejecutará cuando el JAR sea abierto, o cuales librerías externas utiliza, o incluso información de seguridad acerca del archivo JAR. Si no estás usando otros JAR's, puedes omitir el archivo de manifiesto.

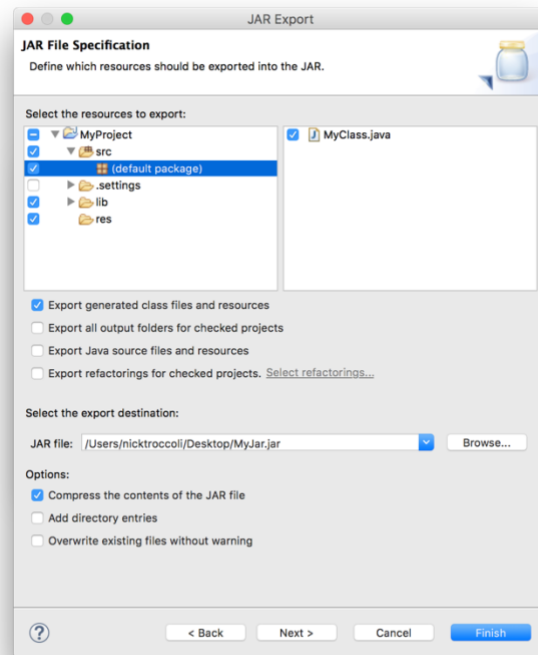
Sigue los pasos para exportar tu programa.

1. Has clic en la carpeta del Proyecto que quieres exportar en la barra lateral de Eclipse, y luego has click en **File -> Export...** La siguiente ventana debería aparecer:

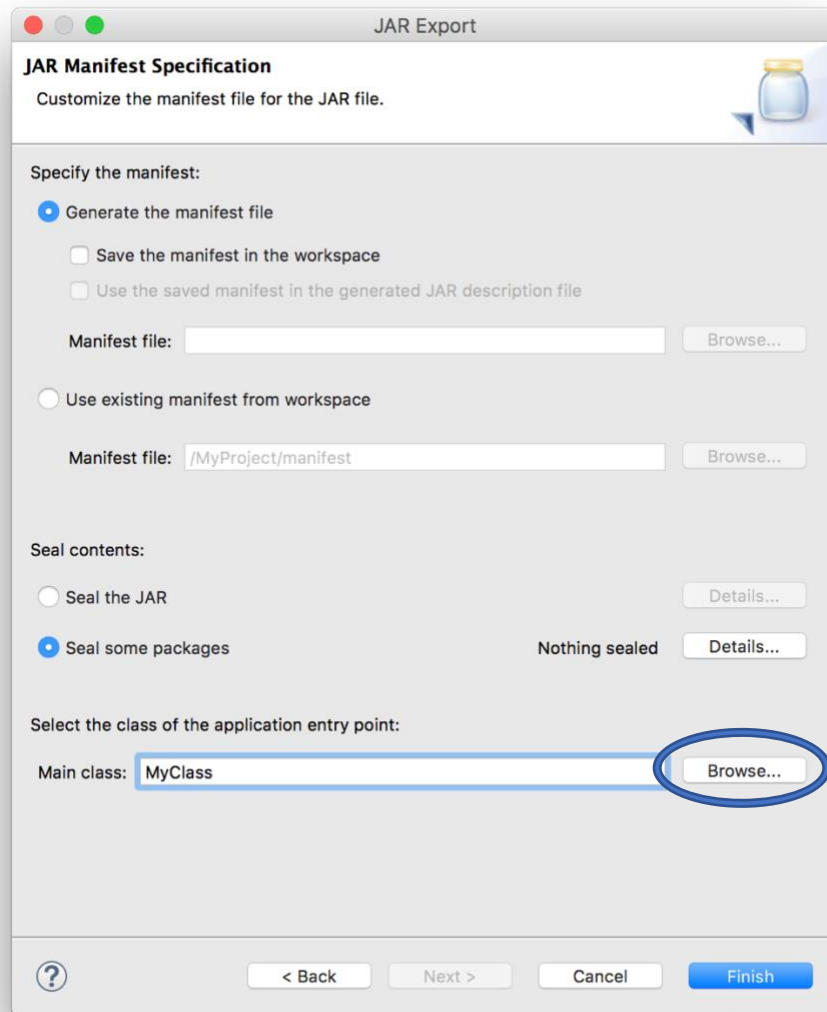


2. Expande la carpeta de **Java**, y selecciona **JAR File**. Oprime **Next**.
3. Deberías ver la ventana de exportación de JAR. Expande la carpeta del proyecto a la izquierda y asegúrate de que **default package** está seleccionado. Selecciona la carpeta destino donde quieres guardar el JAR mediante el botón '**Browse...**' .





4. Oprime **Next**.
5. Clic en **Next** de nuevo en la siguiente pantalla.
6. Ahora estarás en la pantalla final:

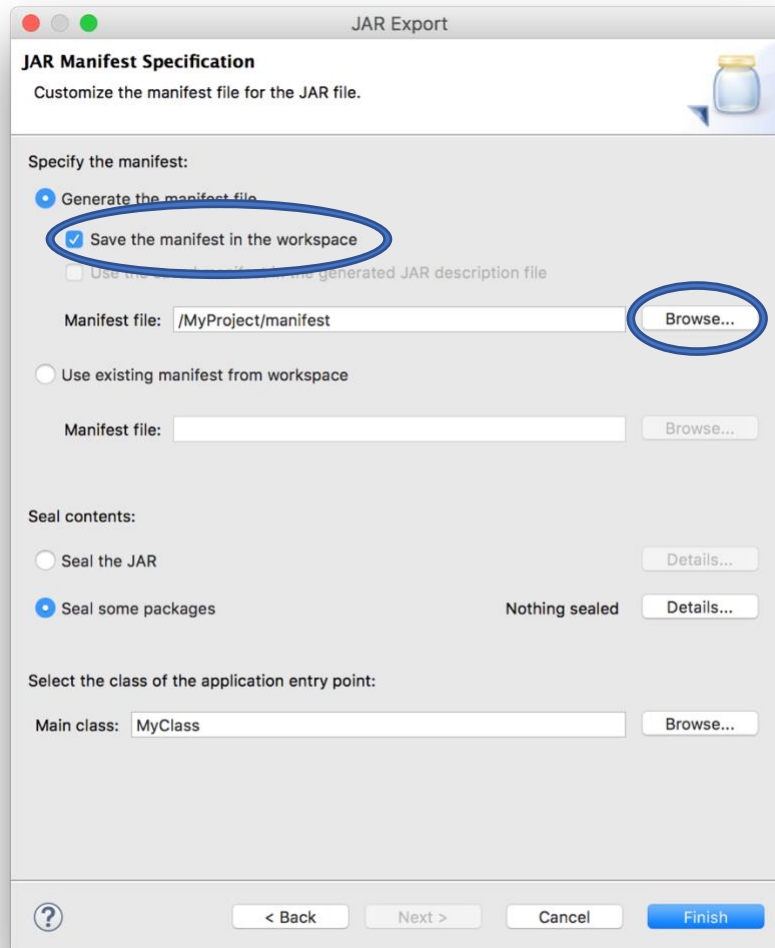


Aquí tenemos que ingresar el nombre de la clase principal que debe ejecutarse cuando el archivo JAR es iniciado. Cerca de la parte inferior de la ventana, selecciona la clase principal (**Main class**) mediante el botón '**Browse...**'. La clase principal, en este caso será **MyClass**. Esta clase debería aparecer en la lista si pusiste correctamente el método main como se describió anteriormente.

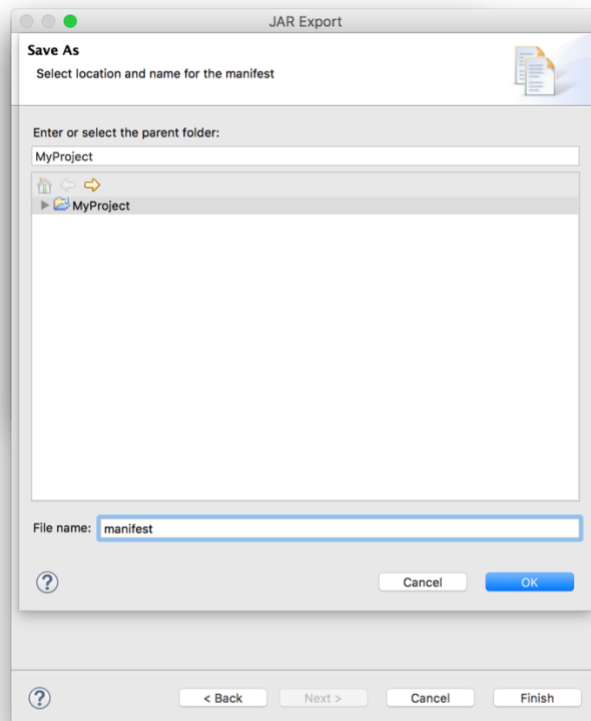
Si estás exportando un proyecto que no utiliza otros archivos JAR, (e.g. sin las librerías de ACM), solamente te queda dar clic en "*Finish*". ¡Has terminado! Ahora puedes enviar el JAR a amigos y familiares. Recuerda que ellos deben tener instalado Java también. Revisa el final de esta guía si tienes dudas al respecto.

Si estás exportando un proyecto que utiliza otros archivos JAR (e.g. las librerías de ACM), continua con el siguiente paso.

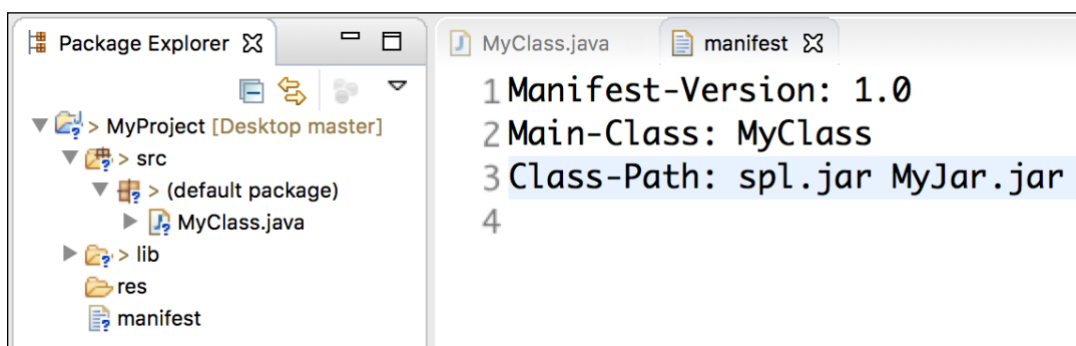
7. Ahora debemos crear un archivo o fichero de manifiesto. Para hacerlo, debemos recurrir *dos* veces al proceso de exportación. La primera vez es para generar un archivo de manifiesto, mientras que la segunda será aquella que, en efecto, exporte tu programa. Desde aquí, asegurate de que el botón '**Generate the manifest file**' cerca a la parte superior de la pantalla esté seleccionado, y que la caja de '**Save the manifest in the workspace**' esté seleccionada. Has Clic en el botón '**Browse...**' asociado con el archivo de manifiesto para escoger la ubicación del archivo de manifiesto que se creará.



8. En la ventana que aparece, has clic en la carpeta **MyProject**, o cualquiera que sea el nombre de tu Proyecto, y luego escribe como nombre de archivo "**manifest**". Presiona "OK" para finalizar. El archivo de manifiesto debería aparecer ahora como **/MyProject/manifest**. Has clic en el botón de '**Finish**'.



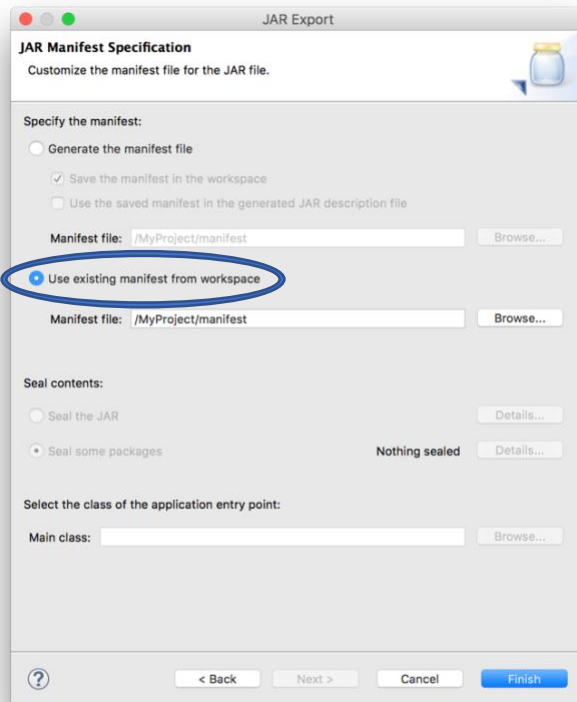
9. Ahora, deberás ver que el archivo de manifiesto aparece en la barra lateral de Eclipse. Ábrelo con doble clic, porque deberás editarlo. Añade lo siguiente: "**Class-Path:** ", seguido de los nombres de todos los JAR's utilizados por el programa, separados por espacios. En este caso, se incluye tanto spl.jar (librerías ACM), como el archive JAR recién creado (MyJar.jar). Al terminar, el archivo de manifiesto deberá ser parecido al siguiente ejemplo.



Asegúrate de que se guarde el archivo de manifiesto actualizado. Teniéndolo, repite el proceso entero para exportar un archivo JAR, que fue descrito anteriormente. En la última pantalla, haremos un paso un poco distinto que será explicado a continuación.

10. Al llegar a la última pantalla, no debes generar un nuevo manifiesto o especificar la clase principal (main). Solamente debes seleccionar el botón de "**Use existing**

manifest from workspace". Oprime el botón de **"Finish"**. Eclipse usará el manifiesto para hacer nuestro propio JAR. Si te pide sobrescribir el viejo JAR, escoge sí.



11.

Ahora tienes tu archive JAR que contiene el Código de tu Proyecto, pero no puedes simplemente enviarlo a tus amigos o familiares. Este JAR no contiene el código que está en los otros archivos JAR (e.g librerías) de las cuales depende, ni tampoco tiene los archivos de datos que el programa puede llegar a necesitar (archivos de texto, imágenes, sonidos). Lo que harás será crear una nueva carpeta, en la que pondrás todos los JAR's utilizados por tu programa y los archivos de datos que utilizas. Asegúrate de que pones los archivos de datos en una carpeta llamada **res** si tu código espera eso. Nota que *no* necesitas poner, por ejemplo, **sp1.jar** en una carpeta **lib**; simplemente ponla en la misma carpeta que tu archivo JAR exportado. Una vez tengas todos estos archivos dentro de una misma carpeta, deberías poder hacer doble clic sobre tu JAR para ejecutarlo. Deberás distribuir la carpeta completa a cualquiera que use la aplicación, por lo que te recomendamos que comprimas en formato ZIP dicho directorio, y simplemente lo envíes completo. Así, el receptor del correo podrá solamente descomprimir dicha carpeta y empezar a usar el programa a través del ejecutable JAR.

Una nota final con respecto al Java Runtime Environment (JRE), (entorno de ejecución)

Cualquiera que no tenga tus archivos JAR podrá correr el programa sin instalar Eclipse, pero debe asegurarse de que tiene el JRE instalado en su computadora. De hecho, lo más probable es que ya tuvieras JRE en tu ordenador antes de empezar a programar, solo para poder correr programas Java. La mayoría de los computadores vienen con JRE preinstalado. Asegúrate de darle este aviso a la gente a la que envíes tus programas, o si no ¡nunca van a poder ejecutar tus maravillosas aplicaciones!