

Java FX Tab

Use this tab to specify the settings for packaging your JavaFX application or application [preloader](#).

The settings depend on the artifact type (JavaFx Application or JavaFx Preloader).

- [JavaFx Application settings](#)
- [JavaFx Preloader settings](#)

JavaFx Application settings

Item	Description
Application class	<p>Specify the (qualified) application main class name. Normally, this is the class that extends the <code>javafx.application.Application</code> class and contains the <code>main()</code> method.</p> <p>Type the name of the class in the field, or click  (Shift+Enter) and select the class in the Choose Application Class dialog that opens.</p>
Title	<p>Specify the application title. In technical terms, this is the information for the <code><title></code> element in the corresponding deployment descriptor JNLP file.</p> <p>If not specified, the title <code>Sample JavaFX Application</code> is used.</p>
Vendor	<p>Specify the name of the application vendor (i.e. the information for the <code><vendor></code> element in the JNLP file).</p> <p>If not specified, the text <code>Unknown vendor</code> is used.</p>
Description	<p>Type a brief description of the application (i.e. the information for the <code><description></code> element in the JNLP file).</p> <p>If not specified, the text <code>Sample JavaFX 2.0 application</code> is used.</p>
Width	<p>Specify the width of the application window in pixels. This parameter is used mainly by applications embedded in a Web page.</p>
Height	<p>Specify the height of the application window in pixels. This parameter is used mainly by applications embedded in a Web page.</p>
HTML Parameters	<p>You may want to pass dynamic parameters to the application that runs in the Web Start or embedded in browser modes from the corresponding HTML page. In that case, you should create a <code>.properties</code> file and specify the necessary set of named parameters in that file. Then you should specify the path to the <code>.properties</code> file in this field.</p> <p>You can type the path, or click  (Shift+Enter) and select the file in the dialog that opens.</p>

Item	Description
Application Parameters	<p>You may want to pass named and unnamed parameters to the application. In that case, you should create a <code>.properties</code> file and specify the necessary parameters in that file. Then you should specify the path to the <code>.properties</code> file in this field. (The specified parameters will be included in the generated deployment descriptor JNLP file.)</p> <p>You can type the path, or click  (Shift+Enter) and select the file in the dialog that opens.</p> <p>Example</p> <p>If the <code>.properties</code> file contains</p> <pre>arg1=value1 arg2</pre> <p>the following elements will be generated within <code><jfx:javafx-desc></code>:</p> <pre><fx:param name="arg1" value="value1"/> <fx:argument>arg2</fx:argument></pre>
Update in background	<p>Use this check box to set the <code>check</code> attribute of the <code><update></code> element in the JNLP file. (This element is used to specify how Java Web Start should handle the application updates.)</p> <p>If selected, <code><update check="background"/></code>. The JNLP client will check for updates in the background while the application is being launched.</p> <p>If not selected, <code><update check="always"/></code>. The JNLP client will always check for updates before launching the application.</p>
Native bundle	<p>Specify which native bundles you want to be generated. (A native bundle is an operating system-specific self-contained application package. Such a bundle contains an application itself as well as a JRE, JavaFX runtime and a platform-specific application launcher.)</p> <p>Select:</p> <ul style="list-style-type: none"> ■ none. No OS-specific bundles are created. The application is packaged into the following files: <ul style="list-style-type: none"> ■ <code><artifact_name>.jar</code>. This file contains the compiled class files and images. ■ <code><artifact_name>.jnlp</code>. This is a deployment descriptor JNLP file for Web deployment modes (Web Start and embedded in browser). ■ <code><artifact_name>.html</code>. This file contains basic code for running the application in the Web Start and embedded in browser modes. ■ all. All the bundles listed below are generated. ■ deb. A Debian software package for Debian GNU/Linux is generated. ■ dmg. An Apple disk image file for OS X is generated. ■ exe. An executable file for Windows is generated. ■ image. A bundle image for the OS that you are using is generated. ■ msi. A Windows Installer (Microsoft Installer) file is generated. ■ rpm. A Red Hat Package Manager file for Linux is generated.

Item	Description
Convert css to bin	Select this check box if you want your application CSS file to be converted into binary format. (This may improve the application performance, especially for "large" CSS files.)
Enable signing	<p>Select this check box if you want the application package to be digitally signed.</p> <p>IntelliJ IDEA can generate a key and the corresponding self-signed certificate, or an existing key may be used to sign the package.</p> <p>To select which way of signing should be used, click Edit Certificates. Then, in the Choose Certificate dialog that opens, select:</p> <ul style="list-style-type: none"> ■ Self signed. The package will be signed with the generated self-signed certificate. ■ Signed by key. The package will be signed with your private key. Specify the associated settings: <ul style="list-style-type: none"> ■ Alias. Specify your private key alias. ■ Keystore. Specify the path to the keystore file. (This is where your private key and corresponding certificate are stored.) Type the path in the field or click  (Shift+Enter) and select the keystore file in the dialog that opens. ■ Storepass. Specify the password for accessing the keystore. ■ Keypass. Specify the password for accessing the key.

JavaFx Preloader settings

Item	Description
Preloader class	<p>Specify the (qualified) application preloader class name.</p> <p>Type the name of the class in the field, or click  (Shift+Enter) and select the class in the Choose Preloader Class dialog that opens.</p>

See Also

Procedures:

- [Configuring Artifacts](#)
- [Packaging JavaFX Applications](#)
- [Applications with a Preloader: Project Organization and Packaging](#)

Language and Framework-Specific Guidelines:

- [JavaFX](#)

Web Resources:

- [Developer Community](#) 