



Configurable Java Launcher for Windows

About

WinRun4j is a java launcher for windows. It is an alternative to javaw.exe and provides the following benefits:

- **NEW:** Flexible native access to any Windows API, compatible with [PINVOKE.NET](#). See [native binding examples](#) for more information. This is currently BETA quality.
- Uses an INI file for specifying classpath, main class, vm args, program args.
- Custom executable name that appears in task manager.
- Additional JVM args for more flexible memory use.
- Built-in icon replacer for custom icon.
- Pre-JVM splash screen with auto-hide.
- DDE implementation for file associations.
- Windows NT Service wrapper.
- Windows EventLog API
- Console version
- Support for 64-bit JVM.
- Supports embedding (inside the executable) the INI file. See [Embedded Resources](#) section below.
- Supports embedding a splash image.
- Supports embedding JAR files. These will be included in the classpath (without extraction). This is currently BETA quality.
- [Eclipse Plugin](#) for integrated launching, debugging and exporting as an executable bundle.

WinRun4J is licensed under the [Common Public License \(CPL\)](#).

Download

The latest [download](#) is available from the Project Page.

The source code for WinRun4J has moved to GitHub. Git patches are now the preferred way of accepting patches (bug fixes and enhancements).

The latest version of the [Eclipse Plugin for WinRun4J](#) can be downloaded from within Eclipse using the following Update URL:

<http://winrun4j.sourceforge.net/>

Usage

The launcher is designed to be used as follows:

1. Copy WinRun4J.exe to [YourApp].exe
2. Create [YourApp].ini (in the same directory)
3. Customize [YourApp].ini (see the table below for information)
4. Create [YourApp].ico (in the same directory)
5. Run `RCEDIT.exe /I [YourApp].exe [YourApp].ico` (this will inject your icon into the executable).
6. Launch [YourApp].exe

The INI file accepts the following settings:

Key	Description
working.directory	This will be the current directory for the app. It can be relative to your executable.
classpath.1, classpath.2, ..., classpath.n	Classpath entries. These will be relative to the working directory above. They can be wildcards (eg. *.jar)
main.class	This is the java class that will be run
vmarg.1, vmarg.2, ..., vmarg.n	Java VM args. These will be passed on to the VM.
vm.version.max	The maximum allowed version (1.0, 1.1, 1.2, 1.3, 1.4, 1.5).
vm.version.min	The minimum allowed version (1.0, 1.1, 1.2, 1.3, 1.4, 1.5).
vm.version	Specify an exact version to use (eg. 1.5.0_07).

vm.location	Specify the location of the JVM dll you want to use. This is useful when you package your own JRE with your app.
vm.heapsize.max.percent	Specify a proportion of the available physical memory to use (ie. relates to -Xmx arg). For example, vm.heapsize.max.percent=75. Note that this will use the maximum memory possible.
vm.heapsize.min.percent	Specify a proportion of the available physical memory to use as the minimum starting heap size (ie. relates to -Xms arg).
vm.heapsize.preferred	Specify a preferred amount (in MB) for the heap size (ie. relates to -Xmx arg). If this amount is not available it will use the maximum amount possible given the physical memory available.
arg.1, arg.2, ..., arg.n	Program arguments. These will be sent before any command line arguments.
log	Standard out and error streams will be redirected to this file (including launcher messages and JNI logging).
log.level	Specify the logging level. One of "info", "warning", "error", "none". Default is "info".
log.overwrite	Set to "true" to cause the log file to be overwritten each time the application/service is launched.
log.file.and.console	Set to "true" to output to the console and the log file (if present). Note: this only applies to WinRun4J log messages and Java logging using the Log class.
log.roll.size	A decimal value in megabytes for the max log size before rolling. Old logs are moved to logname-[timestamp].extension.
log.roll.prefix	Customizes the rolled log file prefix to [prefix]-[timestamp].extension
log.roll.suffix	Customizes the rolled log file suffix to [prefix]-[timestamp][suffix]
log.output.debug.monitor	Useful for monitoring services. Set this to "true" to log to the debug monitor. Use DebugView to view. For Vista/Windows 7 see here if you don't see the output.
splash.image	The name of the splash image file to display (This can be gif, jpg or bmp). This will auto-hide itself when it detects the first application window.
splash.autohide	A flag to disable the splash screen autohide feature ("true").
dde.enabled	This flag needs to be set to "true" to enable DDE.
dde.class	Optional flag to send execute commands to your class.
dde.server.name, dde.topic, dde.window.class	Override the DDE server name, topic and window class.
single.instance	<p>This will detect another instance of the application running and will shutdown if one is found. It takes the following options:</p> <ul style="list-style-type: none"> • "process", this will simply detect if a process for the same executable is present and shutdown. • "window", this will detect if a process for the same executable is present and if there is a visible window - if this is the case it will set the window to the front and then shutdown. • "dde", this will simply detect if a process for the same executable and if dde is enabled it will fire a dde activation call (and then shutdown), which can be picked up by the other process. If dde is not enabled it will simply defer to the "window" method.
process.priority	This can be one of "idle", "below_normal", "normal", "above_normal", "high", "realtime".
service.class	This is the java class that will be run (for a service)
service.id	This is the ID of the service (used for registration)
service.name	The name of the service. This will appear in the service control panel.
service.description	The description on the service. This will appear in the service control panel.
service.controls	The control commands accepted by the service. This can be one or multiple of "stop", "shutdown", "pause", "param", "netbind", "hardward", "power", "session". For multiple simply OR together (eg. service.controls=stop shutdown). The default is "stop shutdown".
service.startup	Can be one of "auto", "boot", "demand", "disabled", "system". The default is "demand"
service.dependency.1, service.dependency.2,	Specifies a set of services that this service depends on.
service.loadordergroup	Specifies the service's load order group.

service.user	Specifies the account to run the service under.
service.password	Specifies the password for the user account.
console.title	Sets the console title (only for console version of launcher).
ini.file.location	The addin will include INI keys from the file location specified (eg "C:\Program Files\MyApp\include.ini")
ini.registry.location	The addin will include INI keys from the registry location specified (eg "HKEY_CURRENT_USER\Software\MyApp"). Currently only string and DWORD values are supported
ini.override	A flag ("true"/"false") to indicate if an external INI file can override values from an embedded one.

Note: INI values can contain environment variables, which will be substituted on startup, eg `log.file=%TEMP%/mylog.txt`

Error Messages

Error messages emitted by the launcher can be customized via the INI file. These can be placed in an "[ErrorMessages]" section:

Key	Description
show.popup	Set this to "false" to disable popups, default is "true".
java.not.found	This message will be displayed as a message box popup when the launcher cannot find an appropriate JVM to load.
java.failed	This message will be displayed as a message box popup when the JVM fails to startup, for example if invalid VM args are entered in the INI file

An example is as follows:

```
[ErrorMessages]
java.not.found=A suitable version of Java could not be found on your system. Please contact VendorX.
java.failed=Java failed to startup successfully. Please contact VendorX.
```

Embedded Resources

The following shows the help information for RCEDIT, the resource editor included in the download:

```
WinRun4J Resource Editor v1.0 (winrun4j.sf.net)

Edits resources in executables (EXE) and dynamic link-libraries (DLL).

RCEDIT <option> <exe/dll> [resource]

filename      Specifies the filename of the EXE/DLL.
resource      Specifies the name of the resource to add to the EXE/DLL.
/I            Set the icon as the default icon for the executable.
/A            Adds an icon to the EXE/DLL.
/N            Sets the INI file.
/J            Adds a JAR file.
/E            Extracts a JAR file from the EXE/DLL.
/S            Sets the splash image.
/C            Clears all resources from the EXE/DLL.
/L            Lists the resources in the EXE/DLL.
/P            Outputs the contents of the INI file in the EXE.
```

Note:

- The embedded INI entries are overridden by an external INI file (if present).

- Any JARs added to the executable will automatically be added to the classpath (before all classpath entries specified in the INI file and in the order in which they are embedded). They don't need to be specified in the INI file.
- If an embedded splash image is present it will automatically appear (it doesn't need to be specified in the INI file).

JNI Library

The JNI has been replaced with dynamic native binding. See [native binding](#) examples for more information.

Examples

An example INI file (included in the download) is:

```
classpath.1=*.jar
main.class=org.boris.winrun4j.WinRunTest
```

An example INI file to run eclipse is:

```
working.directory=.
main.class=org.eclipse.equinox.launcher.Main
classpath.1=plugins\org.eclipse.equinox.launcher*
arg.1=--showsplash
vm.heapsize.percent=80
```

Service Example

The download contains a simple example of a windows service. Its only function is to log to the Windows Event Log every five minutes. To run the example:

1. Create a test directory C:/test.
2. Copy WinRun4Jc.exe to C:/test and rename to service.exe.
3. Copy service.ini to C:/test.
4. Copy WinRun4JTest.jar to C:/test.
5. Run service.exe --WinRun4J:RegisterService.
6. To start/stop the service use the control panel services console (The service name will be "WinRun4J Test Service")
7. To unregister run service.exe --WinRun4J:UnregisterService.

The following shows the included example of a basic Service implementation.

```
package org.boris.winrun4j.test;

import org.boris.winrun4j.AbstractService;
import org.boris.winrun4j.EventLog;
import org.boris.winrun4j.ServiceException;

/**
 * A basic service.
 */
public class ServiceTest extends AbstractService
{
    public int serviceMain(String[] args) throws ServiceException {
        int count = 0;
        while (!shutdown) {
            try {
                Thread.sleep(6000);
            } catch (InterruptedException e) {
            }

            if (++count % 10 == 0)
                EventLog.report("WinRun4J Test Service", EventLog.INFORMATION, "Ping");
        }

        return 0;
    }
}
```

```
}
```

The INI file for this service looks like this:

```
service.class=org.boris.winrun4j.test.ServiceTest
service.id=ServiceTest
service.name=WinRun4J Test Service
service.description=An example service using WinRun4J.

classpath.1=*.jar

vmarg.1=-Xdebug
vmarg.2=-Xnoagent
vmarg.3=-Xrunjdpw:transport=dt_socket,address=8787,server=y,suspend=n
```

Change History

0.4.3

- Fixed loading issue with Java 7
- Added standalone java service launcher.

0.4.2

- Fixed issue with the ini.override flag.
- Fixed some JNI issues with Service and DDE functionality.
- Added ability to log to the debug monitor.
- RCEDIT supports /M option to add a manifest file.

0.4.1

- Fix for loading embedded jars.

0.4.0

- Dynamic native binding implementation using [libffi](#). See [native binding examples](#) for more information.
- Fixed a race condition in the service implementation for quick starting applications.
- The launcher java library now requires java 1.5 minimum (due to use of annotations). The launcher executable is compatible with java 1.4 and above.
- Fixed lowercased key issue with INI file
- Fixed max heap size issue on 64-bit VM
- Added option to set console title via INI file
- Added option to suppress error popups
- DDE activate message sends command line

0.3.3

- Moved service name, description and controls accepted to INI file
- Refactored Service interface and added AbstractService helper class
- Implemented FileAssociations helper class
- Greatly improved the Registry API
- Option to log to console and file
- Log file rolling
- Fix for relative log file path

0.3.2

- Implemented `--WinRun4J:ExecuteINI` built-in command - allows the launcher to execute arbitrary INI files
- Fix for splash.autohide enable/disable option.
- Fix for service startup args
- Fix for dde activate thread hang

Older Change History

Blog

You can find a blog about this launcher, with more information and tips at [Boris Inc.](#)

Contributors

- Peter Smith
- Karl von Randow
- [AquaFold](#), launcher used in Aqua Data Studio.
- Frederic Canut

