

Runtime and Locked Scripts Contents

Runtime Overview

Runtime Scripts (ijr files)

Runtime Applications

Locked Scripts (ijl files)

Runtime Overview

A runtime application:

- can be distributed royalty-free to run on other computers
- may use window driver forms for the user interface
- has no session manager
- shows only the application interface, the J system is invisible
- uses `ijs` scripts as well as encoded `ijr` and locked `ijl` scripts

A J runtime system uses the same executable as the development system, but with an `/rt` command line parameter. A runtime system may be freely distributed.

A runtime application starts by loading the script file given on the command line following the `/rt` parameter. The application terminates normally if it is ready for events and there are no forms or if it executes `2! :55 ' ' .` Session manager output is ignored, but a request for session manager input is an error. If there is an error, the application displays a message box and then terminates.

Your application can have its own icon to use when its forms are minimized. The icon must be in the same directory as the script file given on the command line and must have the same name as the script, but with an `ico` suffix. You can use that same icon as the Program Manager icon for starting the application.

Directory `system\system\examples\demo` contains example files `runtime.ijs` and `runtime.ico`. Run that application with a command line:

```
c:\j401\j.exe /rt system\examples\demo\runtime.ijs
```

Add an icon for starting the application and select `runtime.ico` as the icon.

Runtime Scripts (ijr files)

J Runtime can load normal `ijs` script files. In addition it can load the `ijr` encoded form of script files.

In development your application will likely require several script files, but in general it is preferable for a runtime application to have only a single script file. This can be built using the Project Manager.

The `makejir` command converts a script file to an encoded form. For example:

```
wd 'makejir app.ijs app.ijr'
```

J loaded with the `/rt` parameter will load `app.ijr` and work exactly as if it were the `app.ijs` file. Encoding script files for use with runtime has several advantages:

- prevents accidental modifications
- prevents unapproved modifications
- provides a closed application
- protects proprietary material

The `ijr` encoding is adequate protection from legitimate users. However, it is not proof against a serious attack by skilled computer professionals. If you are concerned with the proprietary nature of your `ijr` files, you must ensure that they are not loaded into a system that contains untrusted code. Similarly, you must ensure you do not load scripts that could contain untrusted code. To do this, your main `ijr` script should first verify that the `namelist` in all locales is empty before continuing. In most cases, this main script should not load any other scripts. If it must, then it must validate that the script is the right one before running it.

If you have security concerns that require more than these precautions, please contact us.

Runtime Applications

Testing a Runtime Application

Since runtime applications have no session manager, there is no way to debug using runtime. Therefore to test the application, first create it using only standard `ijs` script files. Create an icon to load J with the `/noprofile` parameter, that loads the application, for example:

```
c:\j401\j.exe /noprofile system\examples\demo\runtime.ijs
```

Once you are sure the application works correctly, you can then rebuild it into an `ijr` runtime file.

Runtime Enquiry

Your application can determine if it is executing under the runtime system using the command `wd'grt'` which returns a 1 if so, otherwise 0.

Packaging a Runtime Application

Directory `system\examples\runtime` contains the files required to build a runtime setup. File `bldjrt.bat` builds a simple runtime setup directory.

Macintosh

The Macintosh doesn't have command line parameters so there is no `/rt` parameter. If J is started by double-clicking a script file with a suffix of `.ijr`, then J is started as a runtime application. The `ijr` file does not have to be an encoded `ijr` file.

There is no `system\examples\runtime` directory for the Macintosh.

Locked Scripts (ijl files)

The J development system can load definitions from a locked script file (ijl file). The names defined by running an ijl file can be used normally, but their definitions are hidden from the user. An ijl file normally has a file suffix of ijl. An ijl file is similar to an ijr file, but an ijr file can only be loaded by a runtime system, and an ijl file can be loaded by a development system. This facilitates using proprietary or protected definitions within the framework of an open J development system.

```
wd'makej1 xyz.ijs abc.ijl' NB. create ijl file
```

The text for abc.ijl is encoded and running it defines its verbs, adverbs, and conjunctions as locked. Locked definitions will not be disclosed by the J interpreter.

<pre>0!:1 <'foo.ijs' plus =: 4 : 'x.+y.' mean =: 3 : '(plus/y.)%#y.' plust=: + meant=: plust/ % # 3 plus 'abc' domain error: plus x. +y. mean 'abc' domain error: plus x. +y. 13!:0 [1 mean 'abc' domain error: plus x. +y. plus[:0] 13!:1 '' domain error x. +y. plus[:0] (plus/y.)%#y. mean[0] mean'abc' 13!:0 [1 plus 4 : 'x.+y.' meant f. + / % #</pre>	<pre>wd'makej1 foo.ijs foo.ijl' 0!:1 <'foo.ijl' 3 plus 'abc' domain error: plus 3 plus'abc' mean 'abc' domain error: plus mean 'abc' 13!:0 [1 mean 'abc' domain error: plus mean 'abc' 13!:1 '' 13!:0 [1 plus plus meant f. meant</pre>
--	--