JAddin - Java Toolkit for Domino Addin Server Tasks (Freeware)

ABdata, Andy Brunner
Kappelistrasse 43
CH-8002 Zürich
E-Mail andy.brunner@abdata.ch

Table of content

1. Introduction 2

2. Freeware License 2

3. Download 2

4. Installation 2

5. Starting the Application 2

6. Debugging 2

7. Frequently asked questions 3

8. Release History 3

1. Introduction

The JAddin framework is a very thin and convenient layer between the Domino RunJava task and the Java application code. It greatly helps to create Java tasks for Domino by freeing the developer to learn all the Domino addin specifics such as message queue handling, sub-thread creation, Domino resource cleanup, etc. It is written entirely in Java to support all Domino versions since 8.5 and processor architectures (32-bit, 64-bit and 128-bit) on all platforms.

2. Freeware License

"This software shall be used for Good, not Evil."

3. Download

The current version of this tool is available on the website http://abdata.ch/JAddin.hmtl.

4. Installation

- The framework consists of the two Java class files JAddin.class and JAddinThread.class. To install the framework (and your application) you need to copy all necessary Java class files to the program directory (Windows) or to the data directory (Linux) on your Domino server.
- Alternatively you can combine all class files in one JAR container and add this JAR file name to the Domino Notes.Ini parameter, e.g.
  JavaUserClasses=C:\Lotus\Domino\traveler.jar;C:\Apps\Application.jar
- If you update any Java class file on the Domino server, you first need to stop the RunJava Domino task to free the class cache used by RunJava.

5. Starting the Application

There are several ways to start the addin:

- Manually by entering the command in the Domino console, e.g.
  Load RunJava JAddin Helloworld
- Adding RunJava in the ServerTasks line in the Domino server Notes.Ini, e.g.
  ServerTasks=Replica,Router,Update,RunJava JAddin HelloWorld,AMgr
- Creating a program document in the Domino Directory:
  Program name: RunJava
  Command line: JAddin HelloWorld
  Server to run on: Server/ACME
  Enabled/disabled: At server startup only

The JAddin may be started with the special parameter «Debug!» to activate the debugging during startup, e.g. *Load RunJava JAddin HelloWorld Debug!*

6. Debugging

To help the Java developer during the development and testing of the addin, there are several commands available thru the Domino console Tell command. These commands are processed by the JAddin framework and are not sent to the user thread via the addinCommand() method.

These commands are:

Version!	Display the JAddin, Java and OS version numbers
Debug!	Turns the debugging on (see logDebug() method)
NoDebug!	Turns the debugging off (see logDebug() method)
Quit!	Terminate the addin (a bit faster than the regular "Tell addin Quit")
HeartBeat!	Execute the heartbeat processing (normally done every minute)
Memory!	Display the Java virtual machine memory usage
GC!	Runs the Java virtual machine garbage collector

7. Frequently asked questions

- Q: How do I develop my JAddin project in Eclipse?
  A: Just make sure that you include the notes.jar file (installed with Notes and Domino) as an external JAR file in your project.
- Q: During startup, I see the error message
  RunJava: Can't find class JAddIn or lotus/notes/addins/jaddin/JAddIn in the classpath. Class names are case-sensitive.
  A: Make sure that the class names in the "Load RunJava" are case-correctly entered.
- Q: During startup, I see the error message
  JAddin: Error: Unable to load the Java class xxxxxx.
  A: The JAddin framework tried to load the Java class xxxxxx but the specified name could not be found. Make sure that the class file xxxxxx.class is found in the Domino program directory (Windows) or in the Domino data directory (Linux) and that the upper- and lowercase characters match exactly since the Java class loader is case sensitive.
- Q: During startup, I see the error message
  RunJava: Can't find stopAddin method for class xxxxxx.
  A: Make sure you start your addin thru JAddin with the command *Tell RunJava JAddin xxxxxx* and not directly thru RunJava.
- Q: I see out-of-memory errors in Java while executing my addin.
  A: Since all Java addins execute in a single JVM under the control of RunJava, the default Java heap space may not be large enough. The Domino Notes.Ini parameter JavaMaxHeapSize=xxxMB (default 64MB) may be used to increase the heap space.
- Q: What is the heartbeart in JAddin?
  A: The main thread in JAddin gets triggered every 15 seconds to perform some internal housekeeping checks. One of these checks is to make sure that the Java heap space does not get filled up to avoid out-of-memory errors in Java. If the free space falls below 10 percent, the Java virtual machine garbage collector is invoked and a message is written to the console.
- Q: How do I contact the author?
  A: Please send an email to Administrator@ABData.CH if you have any question, suggestion or if you have found a problem using this software.

8. Release History

Version 1.2.0 18/Dec/2013	Add: New command "Version!" to display JAddin, Java and OS version numbers. Change: Send the low heap memory warning message only once for each threshold reached.
Version 1.1.0 18/Jul/2012	Add: New methods addinNextHour() and addinNextDay() to allow for notification of next hour and next day. Change: Several runtime optimizations.
Version 1.0.0 28/Apr/2012	Add: First formal version. Add: New method sendMessage() to create and send a message.
Version 0.5.0	New: First public beta version.

22/Aug/2010	