Atlanta Lotus User Group

August 26, 2010

Debugging Java Agents for Humans

(“TAMING THE 2-HEADED BEAST”)

Bob Balaban
Looseleaf Software
bbalaban@gmail.com

In This Session ...

- Have you ever written a Java agent for Notes or Domino?
- Have you ever needed to debug it?
- What did you do?
  1. Curse the lack of a built-in debugger meant for human use?
  2. Add lots of System.out.println() statements?
     a. And then had to remove them later?
     b. And then had to put them back in again?
  3. Used WAS (or .NET) instead??
  4. Well, despair no more! There is an easy solution!

What We’ll Cover

- Introduction
- Java in Domino — why? (Reminder)
- Eclipse – It’s not just about astronomy anymore
- The "Two-Headed Beast"
  - And why you will love him/her/it
- Demos: Simple and less-simple agents
- Demo: Building a real Java agent in Eclipse
- Wrap-up

Introduction: Bob Balaban

- Developer at Lotus/Iris 1987-1997
- 123/G, 123/W, Notes Agent Manager, LotusScript/Java back-end classes for Notes…
- Founded Looseleaf Software, 1997
- Many articles (The View, Lotus Advisor, etc.)
- A book (*Programming Domino 4.6 With Java*)
- Re-joined IBM August, 2005
- Architect for Notes/SAP integration
- Programming Services Architect for Notes and Domino
- Left IBM Feb. 2008, worked on migration technologies for a year
- Re-started Looseleaf Software Feb. 2009
  - Consulting, development, product strategies

Assumption

- You've already decided to build part of your application as an agent
- Why?
  - WQO/WQS agents are the core of Domino hosted Web applications
  - You might need scheduled functionality
  - You might need triggered functionality
- Agents: Not dead yet

http://www.youtube.com/watch?v=grbSQ6O6kbs

What We’ll Cover

- Introduction
- Java in Domino — why? (Reminder)
- Eclipse — it’s not just about astronomy anymore
- The "Two-Headed Beast"
  - And why you will love him/her/it
- Demos: Simple and less-simple agents
- Demo: Building a real Java agent in Eclipse
- Wrap-up

Java in Domino

Review

- Why would we do that?
  - Work with existing Java code
  - Multithreading
  - Easy network access (especially HTTP)
  - You might move it to J2EE later
- Why would we NOT do that?
  - No Java debugging in Designer (that humans can use)
    - STILL! After 14 years!
  - Domino support for Java lags the rest of the world
    - Better in 8.5.x

Well, At Least We Can Fix the Debugging Problem

- Eclipse!
  - (Everything we’re showing you can also be done in Rational Application Developer (RAD)…)
  - Or (free) Eclipse
  - (Mostly doesn’t work in Domino Designer…) ☹
- Actually, you could use ANY Java Integrated Development Environment (IDE)
  - The Two-Headed Beast is friendly, works everywhere!

What We’ll Cover

- Introduction
- Java in Domino — why? (Reminder)
- Eclipse — it’s not just about astronomy anymore
- The "Two-Headed Beast"
  - And why you will love him/her/it
- Demos: Simple and less-simple agents
- Demo: Building a real Java agent in Eclipse
- Wrap-up

The Two-Headed Beast Domino Agent

- Head 1:
  - A regular Java agent
  - Extends lotus.domino.AgentBase
  - Logic implemented in NotesMain()
  - Get Session from AgentBase
  - Get AgentContext from Session
  - Does its normal thing

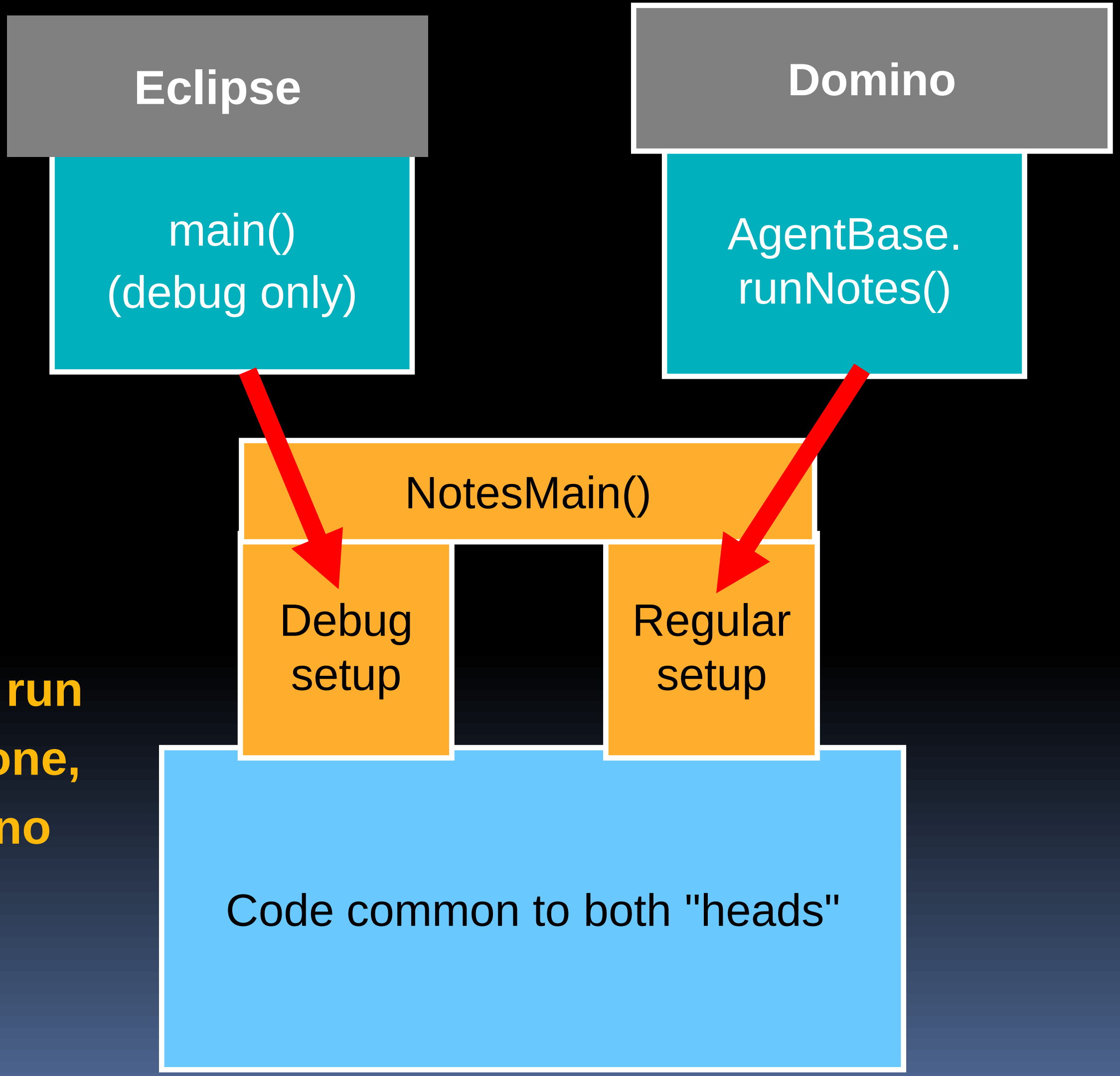
Two-Headed Beast

- Head 2:
  - A standalone Java program
    - So we can debug it!
    - Has a main()
  - The main() function does all debugging setup:
    - Creates a Session
    - Points to "current" database, etc.
    - Invokes NotesMain()
    - Cleans up

Two-Headed Beast, Diagram

Standalone, debuggable

"Real" agent



The agent can run either standalone, or in Domino, no Code changes needed

Notes on the Two-Headed-Beast Technique

- Works for any agent
- You might have to do extra setup
  - (We'll get to that in the second demo)
- You can remove the "debug head" for production
  - But why bother? You might have to put it back later
- You want as much of the code to be common to both heads as possible

Setting up Eclipse to Use Notes JRE

- Needs to be done for compile-time (project properties)
- And for run-time (run configuration)
- Caveat: Varies from Notes version to Notes Version!!
  - YMMV, and UI may be slightly different in different versions of Eclipse

Edit Project Properties

Where to Find It

- Libraries tab
- "Add External Jar", browse to notes\jvm\lib\ext\Notes.jar
- Or…. Set Notes as the default JRE:
  - Window menu ➞ Preferences ➞ Java ➞ Installed JREs
  - Search, or add the Notes JRE
  - In 8.5.x:
    - …\Notes\jvm\
    - …\Domino\jvm
  - You can create an "Installed JRE" for each version of N/D you have installed, then assign 1 of them to each project
  - Select its checkbox

Create a “Run Configuration”

- This is used at runtime, when you execute your program
  - You want Eclipse to use the Notes JRE for this too
- Run menu, “Run…” for EACH JAVA main program:
  - On the Arguments tab, add the following:
    - -Dsun.boot.library.path=<c:\\<notes>;c:\\<notes>\\jvm\\bin
    - -Djava.library.path=c:\\<notes>;c:\\<notes>\\jvm\\bin
    - NOT NEEDED For Notes 8 and later!
  - On the JRE tab, select Project JRE (Notes)
  - set “working directory” to point to the N/D executable directory

Things to Remember

- Run configuration has to be consistent:
  - JRE same version as “working directory”
  - Data tree same as “working directory”
- Not really an issue if you only have 1 version of Notes/Domino installed
- Best to create a Run/Debug configuration for each application in your project

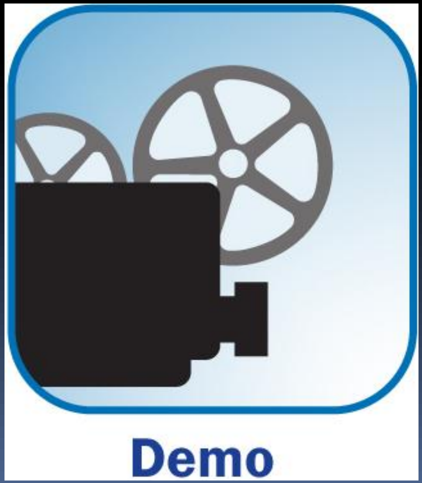
What We’ll Cover

- Introduction
- Java in Domino — why? (Reminder)
- Eclipse — it’s not just about astronomy anymore
- The "Two-Headed Beast"
    - And why you will love him/her/it
- Demos: Simple and less-simple agents
- Demo: Building a real Java agent in Eclipse
- Wrap-up

Demo 1: Simple (Debuggable) Agent

- Basic two-headed beast
- Really just a template
  - Only prints the file path of the "current" database

Agent1.java



Demo 2: Two-Headed Agent With Extra Setup

- Shows extended customization technique:
  - We want the agent to work on the UnprocessedDocuments collection
  - We write a class that extends AgentContext, ONLY used in debug case
  - We fake a DocumentCollection for testing
  - Custom AgentConext delivers the DocumentCollection to the Agent

Agent2.java

DebugAgentContext.java

What We'll Cover

- Introduction
- Java in Domino — why? (Reminder)
- Eclipse — it's not just about astronomy anymore
- The "Two-Headed Beast"
  - And why you will love him/her/it
- Demos: Simple and less-simple agents
- Demo: Building a real Java agent in Eclipse
- Wrap-up

A Real Demo: Create a Domino Agent for a Web Service

- Use RAD to automatically generate a Web Services client for Java
  - Provide URL to the Web Service Definition Language (WSDL)
  - Use the client proxy class in your agent to invoke the service
  - Paste the proxy (and supporting JARs!) into a Domino agent

~~A Real Demo: Create a Domino Agent for a Web Service~~

- ~~Use RAD to automatically generate a Web Services client for Java~~
  - ~~Provide URL to the Web Service Definition Language (WSDL)~~
  - ~~Use the client proxy class in your agent to invoke the service~~
  - ~~Paste the proxy (and supporting JARs!) into a Domino agent~~

TOO MUCH TYPING!!!

Use Notes 8 Instead

- No longer needed after Notes 8.0:
  - Web Services consumer!
  - Can be written in Java or LotusScript (whoo hoo!)
- BUT, you still need an agent to invoke the Web Service consumer
  - And a “2-headed” agent will not be able to “link” to a Notes/Domino WS from Eclipse
- So, what can you do?
  - Write the invoking agent in LotusScript!

Debuggable JDBC Agent

- Java agent uses JDBC to access a relational db
  - SQLServer in this case (hey, it was free)
  - Could also be Db2, Oracle, anything
- Setup steps:
  - Install JDBC driver for vendor’s db
  - Put JAR in Notes or Domino \jvm\lib\ext
  - (Do NOT bother trying to use JavaUserClasses in notes.ini)
  - Write 2-Headed Java agent code
  - NOTE: Agent MUST run with “unrestricted” privileges!

Tip

Demo: JDBC Agent

JDBCAgent.java

Special Bonus Stuff

- You can (sorta/kinda) do some Java debugging inside Designer
  - V8.x, that is (not 7)
- VERY not-supported
- There’s a post on this on my blog

What We'll Cover

- Introduction
- Java in Domino — why? (Reminder)
- Eclipse — it's not just about astronomy anymore
- The "Two-Headed Beast"
  - And why you will love him/her/it
- Demos: Simple and less-simple agents
- Demo: Building a real Java agent in Eclipse
- Wrap-up

Wrap-Up

- Eclipse is free, comes with a good Java debugger
- But does not work in Domino Designer
- You can (with a little bit of work) create Java code that works BOTH in Designer AND in standalone Eclipse

7 Key Points to Take Home

- Two-Headed-Beast technique makes it easy to debug
- Always provide a default constructor
- Keep as much of the code as common as possible
- Create your own Domino-extended objects to customize the debug case
- Use real data whenever possible
- Keep It Simple (except when you can't)
- It's a TOOL, not a RELIGION
  - If it solves more problems than it creates, use it
  - Otherwise, don't

Your Turn!

Q & A

How to contact me:
Bob Balaban
Email: bbalaban@gmail.com
Blog: http://www.bobzblog.com