Troubleshooting Tips and Tricks

Below are some troubleshooting tips and tricks when working with Servoy.

General Log File

The output of Servoy Developer/Eclipse is written in currentWorkspaceDirectory/.metadata/.log. Note that on some operating systems the .XXX directories /files are hidden by default.

Improving Slow Performance/Out of Memory Errors

When experiencing slow performance or out of memory errors, the maximum value for the memory that Servoy Developer is allowed to use can be increased in *servoy.ini* by changin the -Xmx argument from "-Xmx512m" to "-Xmx1024m", for example. (Mac users can expose the servoy.ini file by showing the Servoy application package contents and browsing to the MacOS folder).

Manual Cleanup of Memory

At the bottom of the Servoy Developer window is a indicator that says how much memory is used. The first number is what is actually used by the active process, and the second number is how much of the maximum value is currently allocated to Servoy Developer.

Users can click the Garbage can icon clicked to trigger a cleanup of the memory. This action is typically not necessary, because Java does this automatically, but it may be performed on command as needed.

Gray Screens on Startup

Mac users sometimes experience gray screens on startup. This is due to issues in the Java Runtime environment on OSX, which cannot be worked around. The best remedy is to close all editors before shutting down Servoy/Eclipse.

Crashes in Linux

Linux users have been experiencing complete freezes or crashes of Servoy Developer when using Java 6. Reverting to Java 5 fixes this problem.

Debugging & Dialogs on OSX

Due to an incorrect implementation inside Java on OSX, it it not possible to debug when dialogs are being shown in the Debug Client. Any breakpoint set will be ignored. Unfortunatly, there is no workaround, except using a different platform.