

## **Debugging a CDC Java User Exit in Eclipse**

InfoSphere Change Data Capture supports Java User Exits to allow user code to be executed in response to certain CDC events.

Debugging user exits in the past has relied upon the time-honored method printing output to stdout. User exits can also use CDC tracing to write user exit output to CDC trace files, which can also be very useful for debugging.

However there is a way to use the JVM's Java Debugger tool (jdb) to enable remote debugging of a user exit via Eclipse.

There are a few simple steps to enable this.

### **Step 1**

First, enable remote debugging in the JVM via jdb. This can be done by editing <CDC Install dir>/conf/dmts64.vargs (or dmts32.vargs if your engine is 32-bit) to add a couple of -X jdb arguments. The file content would look like this once those are added (ignore any line wrapping here, as this would all be on one line in the file):

```
-Xdebug  
-Xrunjdbwp:transport=dt_socket,server=y,suspend=n,address=8765  
-Dcom.datamirror.ts.instance=%TSINSTANCE%  
com.datamirror.ts.commandlinetools.script.Startup
```

The value for address is the port that will be used by the debugger.

Start the instance running. You should see output like the following indicating that CDC is running and the debugger is listening for connections:

```
dmts64 -I CDCTEST  
Listening for transport dt_socket at address: 8765  
IBM InfoSphere Change Data Capture is running.
```

### **Step 2**

In Eclipse, click Run -> Debug Configurations. At the bottom of the configurations list, click Remote Java Application, and click the New Launch Configuration button.

Type in or browse to the project name for your user exit code. For connection type, use Standard Socket Attach and give the host for the CDC engine which will execute the ue, and the port for remote debugging as in step 1.

### **Step 3**

Set a break point in your code in Eclipse and run the configuration in debug mode.

When replication causes the user exit to be invoked, execution of your code will stop at the first breakpoint and you can step through it in Eclipse.

Author: abeaton

Date: 6/20/2012