Potential Arca Issue

Daniel B Davidson

21 December 2012

The following are a set of records asscociated with order 121967691 from the file `arcabookftp20070611.csv` which is in the range of problem dates.

```
xat:A,199999935,121967691,P,B,1000,XLI,38.7300,57687,522,E,ARCAX,E
xat:A,199999936,121967691,P,S,1000,XLI,38.7700,57687,522,E,ARCAX,E
xat:A,199999974,121967691,P,B,10000,SPY,151.3500,57687,542,E,ARCAX,E
xat:A,199999998,121967691,P,B,500,VXF,112.0400,57687,552,E,ARCAX,E
xau:A,200000015,121967691,P,B,25900,SPY,151.3100,57687,562,E,ARCAX,E
xau:A,200000025,121967691,P,S,2500,SPY,151.3800,57687,563,E,ARCAX,E
xau:A,200000027,121967691,P,S,500,SPY,151.4200,57687,563,E,ARCAX,E
xau:A,200000030,121967691,P,S,500,SPY,151.3900,57687,563,E,ARCAX,E
xau:A,200000033,121967691,P,S,1000,SPY,151.3800,57687,563,E,ARCAX,E
xau:D,200000173,121967691,57687,632,SPY,P,E,ARCAX,S,E
xau:D,200000180,121967691,57687,632,SPY,P,E,ARCAX,S,E
xau:D,200000185,121967691,57687,633,SPY,P,E,ARCAX,S,E
xau:D,200001013,121967691,57687,742,SPY,P,E,ARCAX,B,E
xau:D,200002990,121967691,57689,412,XLI,P,E,ARCAX,S,E
xau:D,200002991,121967691,57689,422,XLI,P,E,ARCAX,B,E
xau:D,200007523,121967691,57693,472,VXF,P,E,ARCAX,B,E
xau:D,200041138,121967691,57720,041,SPY,P,E,ARCAX,S,E
xau:D,200052720,121967691,57731,110,SPY,P,E,ARCAX,B,E
```

Notice there are 6 adds `(’A’)` and 6 deletes `(’D’)` associated with SPY. I think the way the `arca_depth_translate.pl` processes orders is it assumes there is only one `Add` (`(’A’)`) per order. I think this is the case because of the code below. Note: `handlemsg` is called on each record that is one of `(’A’)`, `(’M’)`, or `(’D’)`. So, if it is an add it calls `addaction` and the first thing that does is check if the order (e.g. 121967691) exists and if it does the app **dies** with the message *add for existing order*. When I run this script on that data I get that message, immediately. So the script is failing early on this, what I assume is, problem data.

```
sub handlemsg ()
{
    if (&numtime(&msgdatetime()) < &numtime(&bookdatetime()))
    {
        print STDERR &msgdatetime().’ ’.&bookdatetime()."\n";
        die "timestamp has gone backwards";
    }
    else
    {
        if    (&record(’MessageType’) =~ /^A$/) { &addaction();    }
        elsif (&record(’MessageType’) =~ /^M$/) { &changeaction(); }
        elsif (&record(’MessageType’) =~ /^D$/) { &deleteaction(); }
        else  { die "unknown message received"; };
        &set_bookdatetime(&msgdatetime());
    }
};

sub addaction ()
{
    &existsorder() and die "add for existing order";
    &addorder();
};
```

I took a file in the good range `arcabookftp20110722.csv` and ran the script `arca_depth_translate.pl` on it and there were no problems. This worked just fine. This means that for any given order id there is only a single `Add`. So, I have no idea what happened with the data in the problem case. If I were to guess, it looks like somehow their side did not disaggregate orders adequately. Perhaps batch orders are supported and the single orderId in the problem case is related to the batch instead of the specific order. One reason this might be reasonable is all adds listed as coming in within 40 milliseconds. I think each order at a different price should have a different order id, yet somehow they seem to have been aggregated. I'll look into a file prior to the date range and see if it works fine - which I expect it would.

An interesting question would be, if it is as it seems - a *garbage in* problem, then is there a way to compensate anyway. It is tough to know.

Do you have a contact for the data we could ask what the deal is?