# The Best Bid and Offer: A Short Note on Programs and Practices

**Joel Hasbrouck***

**First draft: August 4, 2010**

**This draft: October 14, 2010**

## Abstract

This note describes how to determine the best bid and offer (BBO) from the NYSE's monthly TAQ data, the source that underlies most academic research. At a given point in time the best bid is the maximum bid, taken over the set of current bids posted by all venues. This value persists until one of the bids posted by any of the venues changes. Then the maximum is recomputed. The best offer is computed in a similar fashion. This differs significantly, however, from the BBO defined and computed in Wharton Research Data System (WRDS) documentation and sample programs distributed prior to October 2010. Furthermore, the BBO calculation relies on correct ordering of the quote records. Incorrect sequencing within a reporting exchange's records is much more serious than incorrect sequencing between exchanges. This note explains these problems and makes some summary recommendations.

For comments and suggestions on an earlier draft I thank WRDS support personnel. Once the issues discussed herein were raised, they were very responsive in addressing them.

*Finance Department, Stern School of Business, Suite 9-190, 44 West 4th St., New York, NY 10012.
Email: jhasbrou@stern.nyu.edu

## 1. The best bid and offer

The best bid and offer (BBO) represent the best advertised terms of trade immediately available for a security. The best bid and offer are widely used to assess liquidity and evaluate trading cost, in real time and retrospectively, for practical, regulatory and research purposes.

"Best," with respect to the best bid and offer, presupposes a field of comparison. This field is usually clear from context. It might be as narrow as a single trading venue, but we are usually interested in a comprehensive set consisting of all venues where quotes are posted. In this connection, U.S. participants often refer to the *National* Best Bid and Offer (NBBO).

The terms are defined in the text of the U.S. Securities and Exchange Commission's National Market System regulation ("Reg NMS"):

> Best bid and best offer mean the highest priced bid and the lowest priced offer.
> …
> National best bid and national best offer means, with respect to quotations for an NMS [national market system] security, the best bid and best offer for such security that are calculated and disseminated on a current and continuing basis by a plan processor pursuant to an effective national market system plan …

(U.S. Securities and Exchange Commission (2005 )). The usage in practitioner- and investor-oriented settings is similar.

## 2. Derivation

TickData (a data vendor) states: "To derive the NBBO for an equity, you must evaluate each quote for that equity in chronological order. The first step is to determine whether the quote should be considered for NBBO … If a quote is deemed worthy for NBBO consideration, then the next step is to add the current quote to a list of prevailing quotes by exchange. The prevailing quote for an exchange is the most recent quote for on that exchange. From the list of prevailing quotes by exchange, you then determine the best (maximum) bid price, and the best (minimum) ask price" (TickData (2009 )).

That is, the algorithm works on an ordered (consecutive) set of quote records. The steps are:

1. Filter the quote record for eligibility and/or errors.
2. Construct the sequence of prevailing quotes by exchange.
3. Determine the maximum bid and minimum offer.

The present note does not fully address the preliminary eligibility/error filtering in step 1. The rigidity of the filtering depends on the ultimate purpose of the calculation. A bid, for example, might be economically meaningful, even though its mode flag suggests that it is not “firm”. It has also been common practice to exclude bids priced at $0.01 on the presumption that no trades could conceivably occur at such a price. The events of the May 6, 2010 “flash crash,” however, render the appropriateness of this exclusion debatable. Other issues are discussed in Section 5, below. Nevertheless, while we might want to approach step 1 with a modicum of flexibility, steps 2 and 3 are relatively clear.

This note will focus on derivation of the best bid. The modifications for best offer are straightforward. Suppose that for a particular trading venue, *m* bid quotations in a particular trading day are posted at times $0 < t_1 < t_2 < \cdots < t_m$. The bids associated with these times are denoted $b(t_1), b(t_2), \ldots, b(t_m)$. The prevailing bid is a step function defined as

$b(t) = b(t_k)$ where $t_k \le t < t_{t+1}$. That is, the bid for the venue is assumed to persist until it is explicitly changed (or the market closes). To determine the prevailing bid, we have to keep a record of the most recently posted bid. Now denote the set of prevailing bids for the *n* venues (trading centers) comprising the market as $b_i(t),\ i = 1, \ldots, n$. The market-wide (“national”) best bid at time *t* is

$$B(t) = \max_{i=1,\ldots,n} b_i(t). \qquad (1)$$

It is emphasized that the max is taken over venues, not time.

As an illustration, Table 1 contains a partial quote record for ticker symbol “A” on June 2, 2008. For clarity, I only consider exchanges N and P (the NYSE and Arca). The source is the monthly TAQ consolidated quote file, which is the file used most commonly in academic research. The time stamps on the data are hours:minutes:seconds. There are often multiple quote records in a given second. The following discussion assumes that the quote records are correctly sequenced within the second. The time stamps and sequencing assumptions are discussed in greater depth later.

I focus on two groups of observations. The first group consists of all N and P quote records posted at 9:30:20. The best bid established prior to 9:30:20 is 37.00. Within the 9:30:20 set, the first record (33) is a bid of 37.00 from N. This matches the prevailing best bid. Next (at record 34) P bids 36.90, but since this is inferior to 37.00, the best bid remains at 37.00. Records 35 and 36 repeat N’s bid of 37.00. In this context, these records are redundant, but I include them for the sake of completeness. At record 37, P increases its bid to 36.94. This doesn’t top or even match N’s bid of 37.00, so the best bid remains at 37.00. But even though P’s bid is not immediately the best, it will become the best in a moment. At record 39, N drops its bid to 36.93. This establishes P’s earlier bid of 36.94 as the new best bid. Note that in order to make this inference, the algorithm has to remember P’s prevailing bid. Record 45 is the last observation with time stamp 9:30:20. In this record, N bids 36.95. This is the new best bid. It is the best bid as of the end of 9:30:20. A SAS macro to compute the NBBO following these principles is presented in Appendix A. (This code is presented solely for academic research purposes, without admission of liability for any errors.)

The Wharton Research Data System (WRDS) website formerly posted a SAS program to compute a national best bid and offer. The version dated October 9, 2009 is reproduced in Appendix B. The algorithm defines the best bid established at the end of a given second as the maximum bid over all quote records within the second. This maximum is labeled in Table 1 as the “WRDS best bid”. It is 37.00. Looking back over the observations, we can see that this most recently occurred as an N bid at record 38. Exchange N superseded this bid, however, with multiple subsequent bids at lower prices. Thus, the 37.00 bid represents the highest bid that was observed during the second interval, but it can't be assumed current or prevailing at the end of the interval.

The second group of observations in Table 1 consists of all records between 9:30:31 and 9:30:34. The best bid set previously, i.e. prevailing as of 9:30:30, is 36.97. At 9:30:31, P matches the best bid (record 85), but then bids 36.98 (at record 86) which becomes the new best bid. N's bid of 36.97 (at record 87) is inferior. Since 36.98 is also the maximum bid (among all quote records time-stamped 9:30:31), the WRDS result is in agreement. There are two records with the time stamp 9:30:32: P bids 36.98; N bids 97. Again, the WRDS result agrees with the best bid.

At 9:30:34, however, N repeats its bid of 36.97 (three times). The WRDS algorithm identifies the maximum of all bids posted in this interval as 36.97, and considers the best bid to this value. In fact, however, P's bid of 36.98 established at 9:30:32 is still prevailing and valid. This demonstrates that in determining the best bid, it does not suffice to consider all records in a particular second. The best bid might actually have been posted in a previous second.

The WRDS algorithm is correctly computing the quantity that it claims to compute. The program is not, in this sense, erroneous. The problem is that this definition of the best bid is neither generally accepted nor (in most contexts) economically useful.

The difference might be of little importance if the frequency of disagreement were small. To investigate whether this might be the case, I applied both computations to the quote data for ticker symbol "A" on June 2, 2008, for all quotes from exchanges N and P posted between 9:30 and 16:00. The WRDS best bid differed from the standard/accepted computation in over 11% of the computations. Many studies rely on the bid-ask spread or bid-ask midpoint. These variables depend on the best offer (as well as the best bid). It is therefore likely that for these quantities the frequency of disagreement would be even higher.[1]

[1] These issues with the WRDS algorithm came to my attention when I assigned empirical projects in a PhD microstructure course in the Spring of 2010. There is no easy way to determine which published studies (if any) relied on the WRDS program. The calculation of the NBBO is usually regarded as so straightforward as to require no further documentation in academic research papers. Given the increasing extent to which liquidity measures have

## 3. Sequencing

### *A. General considerations*

Because the most recent bid for a venue is presumed to prevail until a new one is posted, determination of the best bid is sensitive to sequencing. The TAQ records contain one-second time-stamps, but what can be assumed about the correct sequencing of records within a second? Since the data contain no sub-second sequencing fields, we can't easily determine whether an observed sequencing corresponds to what actually occurred in the market. Within the second, there are two kinds of potential sequencing problems: mis-sequencing within a particular venue's records, and mis-sequencing between venues.

Mis-sequencing between venues is the lesser concern. For example, suppose that the first group of quote records in Table 1 is reordered as:

| *Record* | *Time* | *Bid* | *Exchange* |
|---|---|---|---|
| 33 | 9:30:20 | 37.00 | N |
| 35 | 9:30:20 | 37.00 | N |
| 36 | 9:30:20 | 37.00 | N |
| 38 | 9:30:20 | 37.00 | N |
| 39 | 9:30:20 | 36.93 | N |
| 40 | 9:30:20 | 36.93 | N |
| 41 | 9:30:20 | 36.89 | N |
| 42 | 9:30:20 | 36.89 | N |
| 43 | 9:30:20 | 36.94 | N |
| 44 | 9:30:20 | 36.94 | N |
| 45 | 9:30:20 | 36.95 | N |
| *34* | *9:30:20* | *36.90* | *P* |
| *37* | *9:30:20* | *36.94* | *P* |

This is a reordering between venues. All of the P quotes (records 34 and 37) are placed after all of the N records, but within N and P groups the relative ordering is unchanged. At the end of 9:30:20, the N's prevailing bid is 36.95 and P's is 36.94. The best prevailing bid is 36.95, the same as in the original data.

been incorporated into analyses in areas outside of microstructure, however, it seems reasonable to suppose that at least a few studies might have been guided by the WRDS code.

Mis-sequencing within venues is more serious. Suppose that the data are reordered as:

| *Record* | *Time* | *Bid* | *Exchange* |
|---|---|---|---|
| 33 | 9:30:20 | 37.00 | N |
| 34 | 9:30:20 | 36.90 | P |
| 35 | 9:30:20 | 37.00 | N |
| 36 | 9:30:20 | 37.00 | N |
| 37 | 9:30:20 | 36.94 | P |
| 39 | 9:30:20 | 36.93 | N |
| 40 | 9:30:20 | 36.93 | N |
| 41 | 9:30:20 | 36.89 | N |
| 42 | 9:30:20 | 36.89 | N |
| 43 | 9:30:20 | 36.94 | N |
| 44 | 9:30:20 | 36.94 | N |
| 45 | 9:30:20 | 36.95 | N |
| *38* | *9:30:20* | *37.00* | *N* |

The positions of N and P records are the same, but in the last run of eight N records, the first record (38) is moved to the last position. With this reordering, N's apparent prevailing bid at the end of the second is 37.00. This would top P's prevailing bid of 36.94, and would become the apparent best bid. Were the data to be reordered in this fashion, there would be no obvious ex post indications of the problem. No sorting procedures could recover the original order.

### *B. Programming*

Most WRDS users will access the data using SAS. A SAS data set is an ordered rectangular array. Most SAS processing in data steps or transformation procedures leaves the ordering unchanged.

There are, however, some significant exceptions. SAS has an implementation of Structured Query Language (SQL), which is invoked by PROC SQL. SQL sometimes offers a concise way to specify merging and aggregation procedures, and has sometimes appeared in SAS code posted on WRDS for TAQ (and other databases).

The SAS SQL User's Guide notes:

> Unless an ORDER BY clause is included in the SELECT statement, then a particular order to the output rows, such as the order in which the rows are encountered in the queried table, cannot be guaranteed, even if an index is present. Without an ORDER BY clause, the order of the output rows is determined by the internal processing of PROC SQL, the default collating sequence of SAS, and your operating environment. Therefore, if you want your result table to appear in a particular order, then use the ORDER BY clause (SAS Institute Inc. (2009 )).

The ORDER BY clause specifies a sort order. The force of the passage is that if we don't explicitly sort, resequencing relative to the original dataset may well occur. This possibility is a general feature of the SQL specification, not just the SAS implementation.

The passage also suggests that use of an ORDER BY clause can guarantee a particular ordering. With respect to the TAQ data, however, this is in fact likely to aggravate the problem. That is, instead of restoring correct sequencing, it is likely to produce incorrect sequencing. In particular, the principal ordering fields in the CQ file are symbol, date and time. It might therefore seem reasonable to use these as ORDER BY fields, as in:

```
proc sql;
create table myQuotes as select symbol, date, time, ex, bid, ofr from
taq.cq_20080602 order by symbol, date, time;
quit;
```

The ORDER BY clause induces a sort. Sorting algorithms may be classified as stable or unstable. A stable sort maintains the relative order of records that have identical sorting keys. While this might seem to be a feature that one would always want, it imposes a cost in computational efficiency. Probably for this reason SQL and PROC SQL use an unstable sort.

It is easy to generate examples where a dataset containing records with duplicate sorting keys is reordered. Surprisingly, some commonly-used unstable sorting algorithms will reorder a dataset even if the input is already sorted according to the specified keys. In the present application, sorting TAQ data by one-second time stamps, using an SQL ORDER BY clause will usually reorder records with the same time stamp.

As an example, I formed a small dataset ("a") consisting of the quote records discussed previously, with the time stamp of 9:30:20. This data, in its entirety, is:

| *Record* | *Symbol* | *Date* | *Time* | *Bid* | *Exchange* |
|---|---|---|---|---|---|
| 33 | A | 20080602 | 9:30:20 | 37.00 | N |
| 34 | A | 20080602 | 9:30:20 | 36.90 | P |
| 35 | A | 20080602 | 9:30:20 | 37.00 | N |
| 36 | A | 20080602 | 9:30:20 | 37.00 | N |
| 37 | A | 20080602 | 9:30:20 | 36.94 | P |
| 38 | A | 20080602 | 9:30:20 | 37.00 | N |
| 39 | A | 20080602 | 9:30:20 | 36.93 | N |
| 40 | A | 20080602 | 9:30:20 | 36.93 | N |
| 41 | A | 20080602 | 9:30:20 | 36.89 | N |
| 42 | A | 20080602 | 9:30:20 | 36.89 | N |
| 43 | A | 20080602 | 9:30:20 | 36.94 | N |
| 44 | A | 20080602 | 9:30:20 | 36.94 | N |
| 45 | A | 20080602 | 9:30:20 | 36.95 | N |

I then created a new dataset ("b") as:

```
proc sql;
create table b as select * from a order by symbol, date, time;
quit;
```

Since the input dataset is already sorted by symbol, date, and time, we might hope that the ordering of records in the output dataset would be unchanged. In fact, dataset "b" is:

| *Record* | *Symbol* | *Date* | *Time* | *Bid* | *Exchange* |
|---|---|---|---|---|---|
| 44 | A | 20080602 | 9:30:20 | 36.94 | N |
| 42 | A | 20080602 | 9:30:20 | 36.89 | N |
| 38 | A | 20080602 | 9:30:20 | 37.00 | N |
| 39 | A | 20080602 | 9:30:20 | 36.93 | N |
| 41 | A | 20080602 | 9:30:20 | 36.89 | N |
| 37 | A | 20080602 | 9:30:20 | 36.94 | P |
| 43 | A | 20080602 | 9:30:20 | 36.94 | N |
| 35 | A | 20080602 | 9:30:20 | 37.00 | N |
| 45 | A | 20080602 | 9:30:20 | 36.95 | N |
| 33 | A | 20080602 | 9:30:20 | 37.00 | N |
| 34 | A | 20080602 | 9:30:20 | 36.90 | P |
| 36 | A | 20080602 | 9:30:20 | 37.00 | N |
| 40 | A | 20080602 | 9:30:20 | 36.93 | N |

The ordering of the records is scrambled, and the reordering changes the implications for the best bid. Under the correct ordering, the prevailing bid for exchange P going forward is 36.94 and that for N is 36.95. After the reordering, P's prevailing bid is 36.90 and N's is 36.93.

In these examples I added a record number field in the data step that accessed the original CQ file. I did this for expositional clarity, to explicitly label record positions. This field is incidentally a unique record identifier, and so could be used as a sort key to ensure proper ordering.

It bears emphasis that the ordering problems arise entirely from the use of SQL. Normal SAS data step processing accesses records in sequential order, and the sorting algorithm invoked by SAS's PROC SORT is (by default) a stable one. WRDS personnel have stated that SQL was not used at any point in constructing the WRDS TAQ SAS databases. It is also important to note that the properties of SQL noted here are not peculiar to the SAS implementation. "Standard" SQL can't be assumed to use stable sorts or to return records in any particular order absent a sort.

## 4. Recommendations

The lessons drawn from this analysis are simples ones:

- Best bid and offer computations should use an algorithm that maintains (as state variables) the prevailing bids and offers from all venues.
- Use of SQL should be avoided, or at least limited to situations where the possibility of reordering is not an issue.
- If SQL must be used, record counters should be added to all data when they are first accessed. These counters can then be used as sort keys to verify proper ordering.

## 5. Some further notes on the TAQ data

### *A. Error filtering*

As noted in Section 2, the first step in using the consolidated quote data usually involves error filtering. This is a complex issue because any filtering involves a trade-off between type-I and type-II errors, and this trade-off depends on the sample and ultimate purpose of the calculation. For this reason, I limit my discussion to a few general considerations.

It is not clear that all exchanges should be included in the NBBO calculation. For example, the quotes currently posted in some venues aren't fast and accessible in the Reg NMS sense. FINRA's Alternative Display Facility (which reports under exchange code "D"), for example, does not provide for automated execution or routing. Furthermore, prior to Reg NMS, trading in NYSE-listed issues was often highly concentrated at the NYSE (the primary market). For many purposes, therefore, restricting consideration to the primary market quotes might serve as the cleanest way to identify reliable bids and offers.

Quotes are sometimes distributed with an indication that they are "not firm" (perhaps, for example, because the market is moving too fast to keep them current). By a narrow definition, they should be excluded from the sample. However, they may still have considerable informational value, and so be useful in projecting future prices or volatility. Similar considerations apply with respect to indicative quotes posted prior to an opening or re-opening, or quotes that lock or cross the market.

It is also common to subject quotes to a spread price filter, removing records with spreads greater than a dollar, for example. This makes sense if the intent is to compute an average spread over some interval (say, a day). Averages are sensitive to outliers, and since the spread is positive, the outliers are not symmetric. The exclusion is less useful, though, if the aim is to locate periods of low (or essentially non-existent) immediacy.

Finally, when a quote record is (by whatever means) declared to be in error, one must decide whether to treat it as an event that removes the previously posted quote. Suppose, for example, that exchange Λ is bidding 100 at 10:00, and that the next record for Λ is an "error" at 10:01. After that, though, should Λ be presumed to have withdrawn from the market, or should the earlier bid of 100 be considered valid?

For these reasons, an error filter that can be deemed universally applicable in all academic research is probably an impossibility.

### *B. Time-stamps*

The time-stamps on the consolidated data used by practitioners are given to the millisecond. These data are described on the NYSE's website as the Daily TAQ data, available via ftp. Most academic researchers, however, will use the Monthly TAQ data, which are time-stamped only to the second.

By matching recent Inet data to corresponding consolidated data, I've tentatively concluded that the TAQ time-stamps are those reported by the venue. In principle this is the time when the trade occurred, as opposed to the time when the report was received by the consolidator.

Since the data channel from a reporting venue to the consolidator is a serial mechanism, it seems reasonable to assume that quote data from a single venue are correctly sequenced. Different venues, however, will use different channels, with varying degrees of latency. It therefore seems unlikely that the relative ordering of quote records between venues is reliable.

The SEC's recent concept release on market structure states: "The average latencies of the consolidation function at plan processors (from the time the processor receives information from the SROs [trading venues] to the time it distributes consolidated information to the public) are as follows: (1) Network A and Network B – less than 5 milliseconds for quotation data and less than 10 milliseconds for trade data; and (2) Network C – 5.892 milliseconds for quotation data and 6.680 milliseconds for trade data" (U.S. Securities and Exchange Commission (2010 )).  It bears emphasis that the latencies referred to in this passage are narrowly defined. They reflect neither the transmission delays between the trading venue and the consolidator, nor those between the consolidator and the ultimate user.

### *C. Size*

The number of shares in a "round lot" is one hundred. Smaller quantities are sometimes referred to as "odd lots".  The consolidated system was set up to report round lot information. When odd lot trades and orders were primarily used by small retail traders, it could be argued that a round-lot reporting system would capture the bulk of activity. Current developments, however, render the consolidated systems less comprehensive.

The most aggressively-priced orders in an electronic limit order book are often for sizes smaller than 100 shares. A bid or offer is not consolidated unless the total size of all the orders at that price meets the 100 share threshold. Thus there may be visible bids and offers that are more aggressive than the bid and offer posted to the consolidated system.

Trade reporting is more complicated. The trade sizes reported by electronic communications networks (ECNs) are usually determined as the executed portion of the standing limit order. Suppose that at a market's best bid price there are two limit buy orders of 50 and 100 shares, and that the 50-share order has time priority. An incoming marketable sell order for 100 shares will execute 50 shares against each limit buy order. As neither execution meets the 100-share threshold, nothing is reported to the consolidated system. Thus, the consolidated tape can only be considered a partial record of trade executions.

## Appendix A: A SAS macro to compute the NBBO

```
/*     nbbo macro

 *     Joel Hasbrouck, August 4, 2010
 *
 *     Usage:
 *     nbbo(cqIn, cqOut, exString, last, ComputeExAtBestBidOfr, keepExBest)
 *
 *     Parameters:
 *     cqIn     input dataset (from TAQ)
 *     cqOut    output dataset (containing the NBBO)
 *     exString A string containing the exchange codes that should be considered. I.e., the nbbo will be constructed
 *         using only these exchanges. The default (exString=ABCDEIMNPQSTWXZ) is essentially all exchanges
 *     last     yes/no (default=no) If 'yes', only the NBBO prevailing at the end of each second is output. If 'no',
 *         then a BBO record is output for each record in the input file.
 *     Infrequently used parameters:
 *     ComputeExAtBestBidOfr  yes/no, default=yes. Includes in the output a string listing all exchanges at the best
 *              offer, and another string of exchanges at the best bid.
 *     keepExBest  yes/no, default=no. If 'yes', then the output dataset will include the full array of bids and offers
 *         prevailing at each exchange.
 *
 */

%macro nbbo(cqIn, cqOut, exString=ABCDEIMNPQSTWXZ, last=no, ComputeExAtBestBidOfr=yes, keepExBest=no);
%global nEx;
%let nEx=%length(&exString);
data &cqOut;
       set &cqIn (drop=mmid);
       by notsorted symbol notsorted date time;
       array exBid(&nEx);
       array exOfr(&nEx);
       retain exBid1-exBid&nEx exOfr1-exOfr&nEx BBid BOfr spread;
       %if %upcase(&keepExBest)=YES %then %do;
        *  The following snippet renames the exBid and exOfr variables to include the exchange symbol
        *  (This is not necessary, but makes for easier reading when debugging;
        %do i=1 %to &nEx;
           %let ex=%substr(&exString,&i,1);
           rename exBid&i=exBid&ex exOfr&i=exOfr&ex;
        %end;
       %end;
       %else drop exBid1-exBid&nEx exOfr1-exOfr&nEx %str(;);

       if first.date then do;
        do i=1 to &nEx;
```

```
           exBid(i) = .;
           exOfr(i) = .;
        end;
        drop i;
       end;
       if ofr=0 or ofr>1999 then ofr=.;
       if bid=0 then bid=.;
       kEx = index("&exString",ex);   *  kEx is used to index the bid and offer arrays;
       if kEx>0 then do;
        exBid(kEx) = bid;
        exOfr(kEx) = ofr;
        BBid = max(of exBid1-exBid&nEx);
        BOfr = min(of exOfr1-exOfr&nEx);
       end;

       %if %upcase(&ComputeExAtBestBidOfr)=YES %then %do;
        length exAtBBid $ %eval(&nEx+1);
        length exAtBOfr $ %eval(&nEx+1);
        exAtBOfr = ' ';
        exAtBBid = ' ';
        do i=1 to &nEx;
           if BBid^=. and exBid(i)^=. and exBid(i)>=BBid then exAtBBid=trim(exAtBBid) || substr("&exString",i,1);
           if BOfr^=. and exOfr(i)^=. and exOfr(i)<=BOfr then exAtBOfr=trim(exAtBOfr) || substr("&exString",i,1);
        end;
       %end;
       %if %upcase(&last)=YES %then %do;
        drop ex bid ofr kex bidsiz ofrsiz mode;
        %str(if last.time then output;);
       %end;
       %else %str(output;);

run;
%mend;
```

## Appendix B: The WRDS NBBO SAS macro (prior to October, 2010)

```
/**********************************************************************/
/* Program : NBBO Derivation using SAS Views                          */
/* Author : ***********, WRDS                                         */
/* Date : October 9, 2009                                             */
/* Description :                                                      */
/* - Calculates the NBBO Quotes for a subset of TAQ Quotes            */
/* - Retrieves the data, applies filters                              */
/* - Calculates the Best Bid and Best Offer Prices, BBO Depth         */
/* - Produces an output in SAS View format with a date stamp          */
/**********************************************************************/
%let cqtime1="09:30:00"t; * First Quote Time;
%let cqtime2="16:00:00"t; * Last Quote Time;
/* Quote Variables, from CQ Datasets */
%let vars_in_cq=symbol date time ofr bid mode bidsiz ofrsiz;
%let stocks='IBM' 'DELL';
%let period=20090120; options msglevel=i fullstimer;
/*--------------------------------------------------------------------*/
/* Calculate the inside quotes at every point in time and the depth   */
/* at the inside quotes from all venues and from all market makers    */
/* As a new time is identified, the algorithm finds the inside bid    */
/* and the inside offer at that time and then adds the depth from     */
/* every venue and every market maker that is at that inside quote    */
/* to the inside quotes. If a better quote is found the previous      */
/* inside is abandoned and the addition procedure is restarted        */
/*--------------------------------------------------------------------*/

data NBBO_&period  (sortedby= symbol date time) / view=NBBO_&period;
set taq.cq_&period (keep=&vars_in_cq);
by symbol date time;
where symbol in (&stocks) and time between &cqtime1 and &cqtime2
      and bid>0 and ofr>bid and bidsiz>0 and ofrsiz>0
      and (ofr-bid)/((ofr+bid)/2) <= 0.10
      and mode not in (4, 7, 9, 11, 13, 14, 15, 19, 20, 27, 28);
if first.time then
  do;
   BB = bid;
   BBIDSIZ = bidsiz;
   BO = ofr;
   BOFRSIZ = ofrsiz;
  end;
else
  do;
   if bid > BB then
```

```
    do;
     BB = bid;
     BBIDSIZ = bidsiz;
    end;
   else if bid = BB then BBIDSIZ = BBIDSIZ + bidsiz;
   if ofr < BO then
    do;
     BO = ofr;
     BOFRSIZ = ofrsiz;
   end;
   else if ofr = BO then BOFRSIZ = BOFRSIZ + ofrsiz;
  end;

/* Retain Observations within Each Time Block */
retain bb bo bbidsiz bofrsiz;
/* Keep Only Last Observation with NBBO Data */
if last.time then output;
label BB = 'Best Bid';
label BBIDSIZ = 'Sum of all the Depths at the Best Bid';
label BO = 'Best Offer';
label BOFRSIZ = 'Sum of all the Depths at the Best Offer';
keep symbol date time BB BO BBIDSIZ BOFRSIZ;
run;
/* END */
/**********************************************************************/
```

Table 1.
Partial quote record for ticker symbol “A” on June 2, 2008.

The sample consists of quote records from the monthly TAQ file with exchange code N (NYSE) or P (ARCA). *Time, Bid*, and *Exchange* are directly from the TAQ file. *Record* is a constructed sequencing field; The *Best Bid* is computed as described in the paper; *WRDS Best Bid* is computed using the algorithm in WRDS code posted prior to October 2010.

| *Record* | *Time* | *Bid* | *Exchange* | *Best Bid, B(t)* | *WRDS Best Bid* |
|---|---|---|---|---|---|
| | | | | 37.00 | |
| 33 | 9:30:20 | 37.00 | N | 37.00 | |
| 34 | 9:30:20 | 36.90 | P | 37.00 | |
| 35 | 9:30:20 | 37.00 | N | 37.00 | |
| 36 | 9:30:20 | 37.00 | N | 37.00 | |
| 37 | 9:30:20 | 36.94 | P | 37.00 | |
| 38 | 9:30:20 | 37.00 | N | 37.00 | |
| 39 | 9:30:20 | 36.93 | N | 36.94 | |
| 40 | 9:30:20 | 36.93 | N | 36.94 | |
| 41 | 9:30:20 | 36.89 | N | 36.94 | |
| 42 | 9:30:20 | 36.89 | N | 36.94 | |
| 43 | 9:30:20 | 36.94 | N | 36.94 | |
| 44 | 9:30:20 | 36.94 | N | 36.94 | |
| 45 | 9:30:20 | 36.95 | N | 36.95 | 37.00 |
| … | | | | | |
| | | | | 36.97 | |
| 85 | 9:30:31 | 36.97 | P | 36.97 | |
| 86 | 9:30:31 | 36.98 | P | 36.98 | |
| 87 | 9:30:31 | 36.97 | N | 36.98 | 36.98 |
| 88 | 9:30:32 | 36.98 | P | 36.98 | |
| 89 | 9:30:32 | 36.97 | N | 36.98 | 36.98 |
| 90 | 9:30:34 | 36.97 | N | 36.98 | |
| 91 | 9:30:34 | 36.97 | N | 36.98 | |
| 92 | 9:30:34 | 36.97 | N | 36.98 | 36.97 |
| … | | | | | |

## References

SAS Institute Inc., 2009. *SAS 9.2 SQL Procedure User's Guide* (SAS Institute Inc., Cary, NC).

TickData, 2009, Deriving the NBBO from TAQ level 1 quotes.

U.S. Securities and Exchange Commission, 2005, Regulation NMS (Final Rule Release No. 34-51808; June 9, 2005).

U.S. Securities and Exchange Commission, 2010, Concept release on equity market structure.