

ObsTxtSds 4.0: Convert The Observer CSV-TXT to SDS Files for GSEQ

Roger Bakeman
Georgia State University

Vicenç Quera
Universitat de Barcelona

ObsTxtSds version 4.0 for CSV and tab-delimited .txt files

November 15, 2016
revised February 20, 2019

This is an update to the Appendix for: Bakeman, R., & Quera, V. (2008). ActSds and OdfSds: Programs for Converting INTERACT and The Observer Data Files into SDIS Timed-Event Sequential Data Files. *Behavior Research Methods*, 40, 869–872.

Correspondence should be addressed to Roger Bakeman, Department of Psychology, Georgia State University, Atlanta, GA 30303, USA, or Vicenç Quera, Departamento de Psicología Social y Psicología Cuantitativa, Sección de Psicología Cuantitativa, Facultad de Psicología, Universidad de Barcelona, Campus Mundet, Paseo Valle de Hebrón 171, 08035 Barcelona, Spain.

Electronic mail may be sent via Internet to bakeman@gsu.edu or vquera@ub.edu.

Web sites: bakeman.gsucreate.org/gseq/ and www.ub.edu/gcai/gseq/

ObsTxtSds 4.0: Convert The Observer CSV-TXT to SDS Files for GSEQ

This write-up describes a program that can convert comma-separated value (CSV) or tab-delimited text files into SDS files (Bakeman & Quera, 2009)—files formatted according to Sequential Data Interchange Standard conventions (SDIS; Bakeman & Quera, 1992, 1995). The resulting SDS files can then be used with our Generalized Sequential Querier analysis program (GSEQ; Bakeman & Quera, 1995, 2009, 2011).

All that is required is that the data files have at least three columns (more are optional): a time, an event code (behavior), and the event type which contains *Start* for the onset of a duration event, *Stop* for the offset of that duration event, and *Point* for the occurrence of a momentary event. Thus a duration event lasts from the time of a *Start* for that code to the next *Stop* for that code (i.e., duration = stop – start; if start = 3 and stop = 7, duration = 4)

ObsTxtSds was initially intended for users who wish to convert data collected with The Observer. Since Version 6 of The Observer, its data have been represented with XML (Extensible Markup Language), a textual data format widely used for the representation of arbitrary data structures that is machine-independent and well-suited for the internet and all languages of the world. However, XML files are difficult to read. As expected of any full-service user program, The Observer has an export capability. It can transform its XML files into CSV text files (usually with the extension .txt), which can be read by spreadsheet programs such as Excel, statistical packages such as SPSS, and other programs such as this one. If you have difficulty producing CSV text files for your data, contact your Observer representative.

ObsTxtSds allows you to read in as many CSV (or tab-delimited) text files as desired, where each file represents one observation session. You can then define factor and factor levels for each observation session, if you wish, and can edit the names you gave to your initial behavior codes and any actor and data modifier codes, shortening them so that the resulting SDIS codes are more readable.

Technical: Text files can be coded following ANSI (Windows default) or UNICODE conventions. The current version of **ObsTxtSds** (4.0) can read standard 8-bit ANSI, UTF-8, UTF-16 (used by later versions of The Observer), but not UTF-32.

1 Selecting Files to Convert

Each text file contains data for a single observational session. GSEQ expects that data for all the sessions you wish to analyze will be in a single file. Thus the first task is to select the files you want to convert into an SDS file. To open one or more text files, select **File > Open** in the main window (or select the open icon on the tool bar). All such files should have the same header line.

2 The Observer CSV Text File Format

- 2.1 The exact format of the export or data text files produced by The Observer have changed with successive versions, but certain essential elements are common to all. As is true of CSV files generally (files that define a rows by columns data grid), all lines (rows) consist of fields (columns) separated by commas with fields enclosed in double quotes if desired.
- 2.2 **Header line.** The header line gives the names for the columns (fields). Often it is the first line in the file, but may be preceded by a few lines that give additional information such as a session date or start time; if present, such lines are ignored. Lines containing data follow the header line. **ObsTxtSds** defines the header as being the first line with three or more fields and with the words *Time*, *Behavior*, and *Type* in the line (optionally, you may define different words).
- 2.3 **Required fields.** **ObsTxtSds** requires three kinds of fields. Required fields are:
- Time: the time the event was coded.
 - Behavior: the event code.
 - Event Type: whether the time is for an onset of a duration event (Start), its offset (Stop), or simply the time of a momentary event (Point).
- 2.4 **Optional fields.** Optional fields are:
- Session: the name for the session (or observation).
 - Actor: who performed the behavior (or subject).
 - Modifier(s): Any additional codes that modify the behavior.
 - Comment: Any comment about a particular event that you wish.

3 Identifying Columns for Conversion, the Header Line, and Event Types

- 3.1 **Key words.** Once you have selected files to convert, the window shown on the next page opens. In order to connect the fields that **ObsTxtSds** needs with the appropriate columns in an OBS data file, the program searches the header line of each file for key words, as shown. The window shows the default words, but you may change them if the header lines in your data files use other words.

Similarly, you can change the key words **ObsTxtSds** uses to define the header line and the words used to indicate the Event Type.

- 3.2 **Marching.** **ObsTxtSds** declares a match (case insensitive) for the first field containing the key word. The key word can appear anywhere in the field, thus “Start” matches “State start”.

If multiple fields contain the same word—“Time” for example—and you want a later one, then you should change the key word to match exactly the heading of the desired column. Because OBS files often contain time in many formats—absolute and relative, with and without milliseconds—this could prove useful.

- 3.3 Often entries in the first column with “Time” in the heading also contain a date: e.g., dd-mm-yyyy hh:mm:ss.ddd. Dates in this format are ignored. **ObsTxtSds** can deal with time in several formats, e.g., hh:mm:ss, mm:ss, sss, hh:mm:ss.ddd, mm:ss.ddd, and sss.ddd where ddd is decimal time (with three digits it would indicate milliseconds) and where hh indicates hours (0–n), mm and ss minutes (0–59), and sss seconds (0–n).

OBS data-file headings, and header-line and event-type text

ObsTxtSds needs to connect SDS roles with the appropriate columns in The Observer text files you want to convert.

For each SDS role you use--the SDS roles of Time, Behavior, and Event Type are required; the other SDS roles are optional--ObsTxtSds expects to find a column heading in your OBS files that contains exactly the text shown in the box next to each SDS role.

For each file, ObsTxtSds need to find a line containing column headers, if not the first line in the file, one shortly thereafter. ObsTxtSds defines the header as the first line that contains the words Time, Behavior, and Type.

For each event, ObsTxtSds needs to know whether the Event Type column contains Start, Stop, or Point.

In all cases, if you want other text used, you can edit the defaults shown. Matches need only contain the text, thus 'Start' matches 'State Start'.

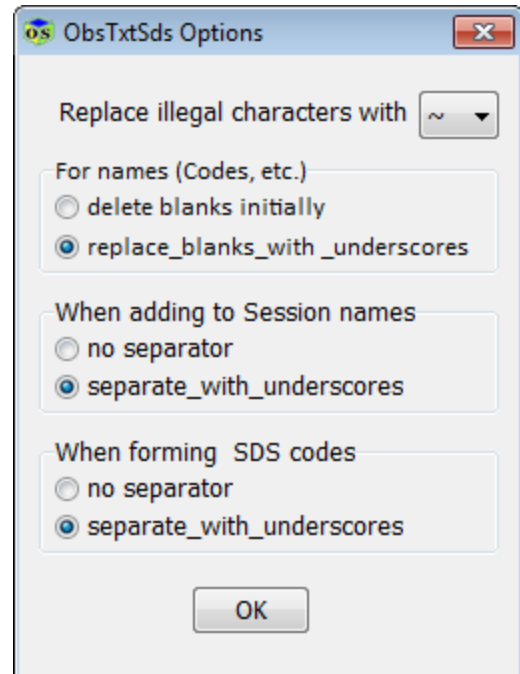
<u>SDS role</u>	<u>text for OBS heading</u>	<u>text for header line</u>
Time	Time	Time
Session	Observation	Behavior
Actor	Subject	Event type
Behavior	Behavior	
Modifier	Modifier	
Event type	Type	
Comment	Comment	

	<u>text for event type</u>
Start	Start
Stop	Stop
Point	Point

OK, proceed

4 ObsTxtSds Options for Forming SDIS Codes

- 4.1 **File > Options** opens an options window. By default, any characters not allowed in SDIS codes are replaced with a tilde (~). The first option lets you pick another character.
- 4.2 The second option lets you specify whether any blanks in names (e.g., for Codes or Sessions) are deleted or replaced with an underscore.
- 4.3 If you opt to form new sessions names (e.g., by combining the file name with the initial session name), the third option lets you specify whether the names are separated with nothing or an underscore.



- 4.4 **Behavior.** In the simplest case, the SDS code is simply the behavioral code from the OBS file. If an actor (subject) code is present, the behavioral code is appended to it. And if one or more modifiers are present, any modifiers are appended to the behavioral codes. The fourth option lets you specify how any concatenated codes should be formed: with no space or with an underscore.

Note. If you want to change any of these options, do so before you select the files to convert.

- 4.5 **Session.** All observation names in a text file should be the same. If present, the program uses the one on the first data line for the session name in the SDS file.
- 4.6 **Comment.** If present, comments are included in the SDS file. If you don't want comments included, simply delete the key word "comment" from the window shown on the previous page.

5 Sessions and Research Factors

- 5.1 Once you have selected your OBS text files, and after you have identified columns for conversion, a new window opens. It lists the file names, their session (observation) names, and the number of data lines for each session (i.e., # of events). You can replace a session name with its file name or add its file name to it or edit it as you wish. The session names are used to label session in the SDS file.
- 5.2 Often sessions (represented by the separate data files) are embedded in a research design consisting of factors (either between- or within-subjects), each with two or more levels (e.g., sex with two levels, male and female, or age with three levels, 3-, 4- and 5-years). SDIS conventions let you declare up to 7 factors. For example, the SDIS declaration, which gives the data type and names of codes, could end with:

... * sex (male female) age (1year 2year 3year);

indicating two factors, one with two and one with three levels. Then, the following SDIS representation for a session:

```
<subject #24> (female 2year) ... /
```

would indicate that the session with the identifier *subject #24* was for a 2-year-old female.

- 5.3 If you want sessions to be identified with their factor levels, indicate the number of factors using the up-down buttons. A new row opens at the top of the grid with default factor names (Factor#1, etc.) and space is made in the table for each session's levels names. You can edit factor names and add level names for each session. The declaration in the SDS file will then contain the factor names and their levels and the sessions in the SDS file will indicate their levels.

If several sessions have the same factor level, you need only enter the level name for the first. **ObsTxtSds** will automatically fill in subsequent level names for that factor. This can be useful when sessions are ordered by factor levels.

6 Forming SDIS Codes

- 6.1 Selecting *Next* at the bottom of the window displays a two column grid. The first column contains the codes in your data file and a second column displays those same codes, but in a form you can edit. If you have actor codes, first actor codes are displayed and selecting *Next* will display behavior codes. If you have no actor codes, behavior codes are displayed immediately. Then, if you have modifier codes, selecting *Next* will display them. If you do not have any modifier codes, you can select *Convert* immediately.

ObsTxtSds forms SDIS codes by concatenating your initial observer codes. For example, if Actor = Mother, Behavior = Soothes, and Modifier = Vocalizes, the SDIS code would be either MotherSoothesVocalizes or Mother_Soothes_Vocalizes, depending on which separator option for forming SDS codes you chose.

Some of your initial codes may be quite lengthy and concatenating makes the SDIS codes even more so. For that reason, it is often advantageous to edit your initial codes (the second column) before converting. Remember that by default SDIS codes are case sensitive, so *mother* and *Mother* are two different codes.

To avoid ambiguous SDIS codes, the codes in each edited list usually are, but are not required to be, unique; if they are not unique, you are warned and given the option of changing the edited list before proceeding.

- 6.2 Later, if you wish, you can form additional codes using GSEQ's flexible data modification commands. For example, the following GSEQ command:

```
OR MomActs = MotherSoothesVocalizes MotherIgnoresQuiet ... ;
```

would let you create a new code representing several different mother behaviors. For further details see Bakeman and Quera (2009, 2011).

- 6.3 SDIS codes are formed from letters, digits, and special characters, for example # ^ _ [] ? and |. Specifically, 19 characters used for SDIS syntax may not be used; they are ! " \$ % & ' () * + , - . / : ; < = >. By default, **ObsTxtSds** replaces any of these 19 characters with a tilde (~) but you can change this to _ # ? or | if you wish. **ObsTxtSds** also replaces any blanks with either nothing or an underscore (to change these defaults see #4, Options).

7 Saving and Restoring Edited Versions of Codes

- 7.1 Like data cleaning in general, data conversion is almost always an iterative process. Likely you will run the conversion program not once and be done, but multiple times as you find and fix errors and other anomalies. You will edit most of the lists the conversion program prepares and understandably you won't want to redo the same edits every time you run the conversion program with the same or similar Observer data files.
- 7.2 If you want to save the edited versions of your Behavior and any Actor or Modifier codes, select **File > Save** in the code editing window. Next time you run **ObsTxtSds** you can select **File > Open** to restore your previous edits. The extension used for the save file is *tdf*, for tab-delimited file. This tab-delimited text file can be edited with any word processor or Excel. In fact, you may find it easier to save and then edit this file with a word processor than to edit the tables within the conversion programs.

8 Time Rounding and Other Conversion Options

Before converting your file, you can specify whether to round time to the second or to the tenth, hundredth, or thousandth (i.e., millisecond) of a second, the line length for the SDS file produced, and whether to display the converted lines in the main window.

- 8.1 All times in the SDS file are represented as MM:SS, MM:SS.d, MM:SS.dd, or MM:SS.ddd, depending on which rounding option you selected, where *d* is a decimal digit, SS is second (0–59), and *MM* can be any number, including numbers larger than 59 (i.e., hours are not represented separately).
- 8.2 Your data files may contain time represented as milliseconds (three digits after the decimal point). Nonetheless, select the rounding option you think most reasonable.
- 8.3 In many cases, rounding to seconds may be reasonable. However, if similar events occur only tenths of a second apart, the SDIS compiler may issue an error message like "code precedes or overlaps itself" when rounding with second precision. In such cases, select a rounding option with greater precision—one that doesn't result in SDIS compiler errors. After compiling the SDS file produced by **ObsTxtSds**, you may find it convenient to round times in that file to lesser (and more reasonable) precision using the rounding utility in GSEQ. See the GSEQ help file for details.

The advantage of rounding to a second is that the printer-style plots in the GSEQ Plot and Kappa routines are easier to read. The disadvantage of rounding to a second or a tenth of a second—which is less precise than the 1/25 or ~1/30 second precision that video recordings provide—is that rounding errors may be introduced. Percentages for events that are mutually exclusive and exhaustive may not sum exactly to 100%. If you want to maintain the full precision provided by video recording, round to a hundredth of a second, but rounding to a millisecond does not seem justified by the precision afforded by video recordings.

9 SDIS Declaration

- 8.1 The first line of an SDS file begins with a type declaration, which in this case is *Timed* (SDIS allows for other possibilities). It is followed by a list of codes.
- 8.2 The conversion program lists codes alphabetically (or alphabetically within actor, behavior, and modifier codes). GSEQ output will order analyses for codes as they appear in the declaration. If you want a different order, edit the declaration in the SDS file to reflect the order you wish.

9 Compile Errors

The SDIS format is precise and attempting to compile in GSEQ an SDS file produced by the conversion program may reveal anomalies and errors that previously went undetected. SDIS compile error message that can occur include:

- a. Error 51: Wrong time. In SDIS events must be time-ordered. This error occurs when a code begins earlier than a preceding code.
- b. Error 56: Wrong session offset time. The last code cannot have the same time as the offset time. To fix, increment the offset time.
- c. Error 60: Code precedes or overlaps itself. In SDIS the times for the same code may not overlap.

In particular, the SDIS compiler expects events to be time ordered. Usually OBS text files are ordered by onset time, so this should not be a problem.

10 Problems

Please let us know of any errors or problems you encounter. If you encounter a problem, first try to get help locally; often a second pair of eyes proves beneficial. Otherwise, try to create a very small test file that illustrates the program and e-mail it to us, along with a description of the problem—we will try to find a solution.

References

- Bakeman, R., & Quera, V. (1992). SDIS: A sequential data interchange standard. *Behavior Research Methods, Instruments, and Computers*, 24, 554–559.
- Bakeman, R., & Quera, V. (1995). *Analyzing Interaction: Sequential Analysis with SDIS and GSEQ*. Cambridge, UK: Cambridge University Press
- Bakeman, R., & Quera, V. (2008). ActSds and OdfSds: Programs for Converting INTERACT and The Observer Data Files into SDIS Timed-Event Sequential Data Files. *Behavior Research Methods*, 40, 869–872.
- Bakeman, R., & Quera, V. (2011). *Sequential analysis and observational methods for the behavioral sciences*. Cambridge, UK: Cambridge University Press.

Perhaps the best resource for SDIS syntax and information about the capabilities of the GSEQ program is the help file that comes with GSEQ: **Help > Show GSEQ Help** in GSEQ.

Visit our websites for more information and to download GSEQ:

<http://bakeman.gsucreate.org/gseq>

<http://www.ub.edu/gcai/gseq/Download.html>

Example

1. This CSV text data file (aligned and shown without the double-quotes for readability):

Start Date:	4/25/2011					
Start Time:	0:00:00					
Start Time (ms):	0					
Header Lines:	5					
Relative Time (seconds)	Observation Name	Event Log File Name	Subject	Behavior	Event Type	Comment
0.000	Dyad5	Event log0001	Exp	Present	State start	
12.267	Dyad5	Event log0001	Kid	Cry	State start	
15.467	Dyad5	Event log0001	Mom	Look	State start	
17.500	Dyad5	Event log0001	Mom	Sooth	State start	
21.967	Dyad5	Event log0001	Mom	Sooth	State stop	
27.667	Dyad5	Event log0001	Mom	Look	State stop	
42.400	Dyad5	Event log0001	Kid	Cry	State stop	
63.733	Dyad5	Event log0001	Exp	Present	State stop	
63.733	Dyad5	Event log0001	Exp	Absent	State start	
92.841	Dyad5	Event log0001	Kid	Burp	Point	
113.533	Dyad5	Event log0001	Kid	Look	State start	
129.300	Dyad5	Event log0001	Kid	Voc	State start	
131.633	Dyad5	Event log0001	Mom	Voc	State start	
132.267	Dyad5	Event log0001	Kid	Look	State stop	
132.700	Dyad5	Event log0001	Kid	Voc	State stop	
139.433	Dyad5	Event log0001	Mom	Voc	State stop	
179.000	Dyad5	Event log0001	Exp	Absent	State stop	

specifying no separator when forming SDS codes, rounding to a tenth of a second, and requesting one item/line, produces this SDS file:

<pre>% File converted by ObsTxtSds ... Timed <tenths> ExpAbsent ExpPresent KidBurp KidCry KidLook KidVoc MomLook MomSooth MomVoc ;</pre>	<pre><Dyad5> % "Start Date:", "4/25/2011" % "Start Time:", "0:00:00" % "Start Time (ms):", "0" ExpPresent, 0:0.0-1:3.7 KidCry, 0:12.3-0:42.4 MomLook, 0:15.5-0:27.7 MomSooth, 0:17.5-0:22.0 ExpAbsent, 1:3.7-2:59.0 KidBurp, 1:32.8 KidLook, 1:53.5-2:12.3 KidVoc, 2:9.3-2:12.7 MomVoc, 2:11.6-2:19.4 , 2:59.0 /</pre>
--	--

2. This CSV text data file (aligned and shown without the double-quotes for readability):

Start Date:	4/25/2011					
Start Time:	0:00:00					
Start Time (ms):	0					
Header Lines:	5					
Relative Time (seconds)	Observation Name	Event Log File Name	Subject	Behavior	Event Type	Comment
0.000	Dyad5	Event log0001	Exp	Present	State start	
12.267	Dyad5	Event log0001	Kid	Cry	State start	
15.467	Dyad5	Event log0001	Mom	MLook	State start	
27.667	Dyad5	Event log0001	Mom	MLook	State stop	
42.400	Dyad5	Event log0001	Kid	Cry	State stop	
63.733	Dyad5	Event log0001	Exp	Present	State stop	
63.733	Dyad5	Event log0001	Exp	Absent	State start	
92.841	Dyad5	Event log0001	Kid	Burp	Point	
113.533	Dyad5	Event log0001	Kid	KLook	State start	
129.300	Dyad5	Event log0001	Kid	KVoc	State start	
131.633	Dyad5	Event log0001	Mom	MVoc	State start	
132.267	Dyad5	Event log0001	Kid	KLook	State stop	
132.700	Dyad5	Event log0001	Kid	KVoc	State stop	
139.433	Dyad5	Event log0001	Mom	MVoc	State stop	
179.000	Dyad5	Event log0001	Exp	Absent	State stop	

specifying no separator when forming SDS codes, editing actor codes to nil (makes actor codes not unique, but this is OK because behavior codes are unique), and then rounding to a tenth of a second and requesting one item/line produces this SDS file:

% File converted by ObsTxtSds ... Timed <tenths> Absent Burp Cry KLook KVoc MLook MVoc Present ;	<Dyad5> % "Start Date:", " 4/25/2011" % "Start Time:", " 0:00:00" % "Start Time (ms):", " 0" Present, 0:0.0-1:3.7 Cry, 0:12.3-0:42.4 MLook, 0:15.5-0:27.7 Absent, 1:3.7-2:59.0 Burp, 1:32.8 KLook, 1:53.5-2:12.3 KVoc, 2:9.3-2:12.7 MVoc, 2:11.6-2:19.4 , 2:59.0 /
--	---

Editing all actor codes to nil will result in the message, "Edited actor names are not unique. Fix (yes) or ignore (no). In this case, select "no."

3. This CSV text data file (aligned and shown without the double-quotes for readability):

Relative Time (seconds)	Observation Name	Event Log File Name	Subject	Behavior	Event Type
3835.89	2	Event log0001	Kid	Passive Noncompliance	Point
3836.979	2	Event log0001	Mom	Reason	Point
3838.927	2	Event log0001	Mom	Physical Power Assert	Point
3840.902	2	Event log0001	Mom	Alternative offer	Point
3840.902	2	Event log0001	Kid	Passive Noncompliance	Point

specifying an underscore when forming SDS codes, rounding to a second, and requesting one item/line, produces this SDS file:

```
% File converted by ObsTxtSds
11/1/2016 1:38:05 PM
Timed <seconds>
Kid_Passive_Noncompliance
Mom_Alternative_offer
Mom_Physical_Power_Assert
Mom_Reason
;

<2>
Kid_Passive_Noncompliance,63:56
Mom_Reason,63:57
Mom_Physical_Power_Assert,63:59
Mom_Alternative_offer,64:1
Kid_Passive_Noncompliance,64:1
,64:1
/
```