# ASCtoFMConverter User Guide

1. Introduction

Altered States of Consciousness (ASC) data is collected in a continuous fashion with "trial" lengths often measured in minutes. In order to extract features of the physiology from these states, it is necessary to split the continuous data into workable record lengths for processing by FILMAN (FM). In the following, the term <u>episode</u> is intended to generally indicate one of these "trial" segments. ASCtoFMConverter can be used to describe these episodes and the processing needed to create a FM file by "chopping" the episodes into shorter records.

2. Opening dialog

On starting the program, one is presented with the following dialog box for opening the header (HDR) file of the dataset to be processed.
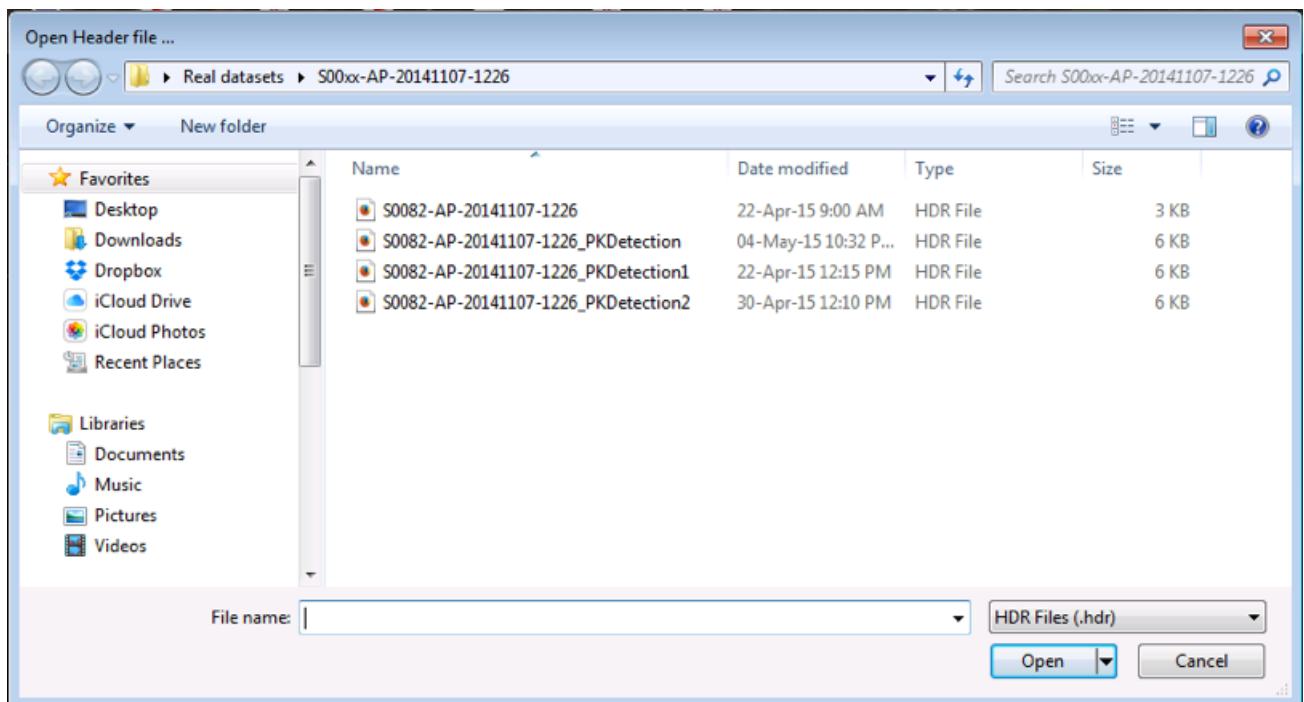
3. Main window

After the dataset is opened one is presented with this window which one uses to describe the processing to be performed. This window is divided into regions for describing the episodes, naming the group variables (GV) to be included in the FM file, the channels to be included, how the samples are to be processed, and the referencing to be used on the samples.

Convert S0082-AP-20141107-1226_PKDetection

Generic Healing - 11-07-2014 12:26 S=0082

Episode descriptions

Episode description +

New GV	From Event	Any Event ▼	with GV	*None* ▼	= ▼		offset	0
1	To Event	Same Event ▼	with GV	*None* ▼	= ▼		offset	0
value	Exclude Event	None ▼	to	Same Event ▼		☐ May use EOF as last "To Event"		

+ −

Sync clock using: ◉ first covered Event ○ "middle" covered Event ○ none

Group variables

Group Variable	Description
Intention	Target selected by computer.
Feedback	Feedback to the subject.
Milliseconds	Milliseconds since start of the trials.
RunNumber	Run number within this experimental session.

Select All   Select None

Total selected = 0

Channels

1-55	55 channels

Samples

Decimation factor	1	512	samples/sec
Record length(sec)	4	2048	points
☐ Radin reference: from	0		points
to	4		points

◉ None   ○ Remove channel offsets   ○ Remove channel trends

Reference

○ Selected channel(s)->	1-55	55 channels
○ Reference expression->	(1-55)~{1-55}	1 reference set

◉ No correction (CMS/DRL)

Exit   Convert

Status: preliminary

4. Episode descriptions

Episode descriptions is the most complicated of these segments. Multiple descriptions may be used for a given FM file and each may be labeled by a GV value, entered on the left of each description panel. This GV is named "NewGroupVariable" in the FM file created. Each episode type is described by a panel:

To describe an episode one must indicate an Event at the beginning of each episode ("From Event") and an Event at the end of the episode ("To Event").

In the drop-down are listed all the Events in the underlying dataset and two "anonymous" Events: Any Event and Beginning of file. These are generally self-descriptive.

A similar list is presented for the "To Event" row. Again, all the available GVs from the input dataset are included as well as some "anonymous" Events. Same Event is the same Event that started the episode. Next Event is the next chronological Event in the Event file, qualified by any to mean both covered and naked Events, and covered, only covered Events. Recall that "covered" implies that there is a corresponding change in the Status channel to the value of the Grey-code which identifies the Event in the Event file. This implies that "naked" Events have no such marker in the Status channel and that the timing of these Events is wholly dependent on the time recorded in the Event file record. This will be discussed further below when reviewing "clock synchronization".

The Events selected for the beginning and end markers may be further refined by adding a GV criterion. Thus, for instance, in the example below, the beginning of these episodes would be marked by an "IntentionSelected" Event with the GV "Feedback" set to "NEUTRAL". "IntentionSelected" Events with other values for "Feedback" would be ignored and not used to start data collection episodes.

So let's look at some examples of episode descriptions. In the first, "IntentionSelected" with a "Feedback" GV set to "NEUTRAL" would be sought to begin an episode, ending at the same Event, 24 seconds later, independent of any intervening Events. Note that "offset" refers to the

number of seconds displaced from the nominal Event that the actual data collection begins or ends. This value is positive for later or delayed, or negative for earlier or anticipatory timing.

Here we've used the same initiating Event, but the end of the episode is 4 seconds before the next "EndTrial" Event. If each "IntentionSelected" Event is paired with an "EndTrial" Event , this will work as (likely) intended. Of course, the trials need to be at least 4 seconds long; otherwise the episode will be skipped. Note that this means the offset is applied after the two Events are selected.

Here's an example which might be used to collect baseline data. 60 seconds of data would be collected from the beginning of the BDF file. Note the *per force*, only a single episode of data is collected when using a "Beginning" of file "pseudo-Event".

If a "StartExperiment" Event is available, one could use this example to collect data from the beginning of the file to 15 seconds before "StartExperiment".

In this final example (below), we have used a simple "IntentionSelected" to "EndTrial" episode description, but used the "Exclude Event" feature to indicate that segments of data between Event "**ArtifactBegin" and "**ArtifactEnd" should be excluded from data collection. If any

portion of a collected record includes data from the excluded segment, the record will no be created.

In addition, note that the "May use EOF as last 'To Event'" checkbox has been marked. This means that if there is no terminating "EndTrial" Event after the last "IntentionSelected" Event, the end of file may be used as the terminating (pseudo-) Event. This should be used with caution as it indicates that the last trial may not have ended appropriately.

5. Counting PK events

In order to create new GVs based on the counting of PK detector events, click on the "+" tab. One will then see a panel looking like this:

Here the idea is to count the number of PK events in each FM record, creating a new GV with this count. Of course, the GV value may be 0 if no events occur during a give record segment. Some routines in FILMAN may not work correctly for GVs with a zero value, so be careful.

This counting only works on datasets that have been pre-processed by the PKDetectorAnalyzer, which creates additional, naked, Events in a new dataset marking the points that it finds PK events. The new Events are named "PK detector event" and you will see this name in the list of Events on the Episode description panel (though you probably don't want to select it). There are a number of GVs associated with this Event type and one may use these GVs to select a subset of the PK detector Events for counting.

First, enter a name for the new FM GV. Think of this as a "bin" in which the counting occurs. Since more than one of these panels may be opened, if several of them name the same GV "counting bin", this GV will contain the sum of these counts. This is particularly useful when accumulating PK events from several detector channels. Then select the detector channel of interest. This drop-down will only display the channels on which PKDetectorAnalyzer has performed Event creation.

These are the only two entries that are required. A subset of PK events may be selected however by using the other controls on the panel. "Found" indicates whether the events should have a valid non-linear signal fit found by PKDetectorAnalyzer with "Either" indicating either a found fit or not. "Direction" refers to either positive- or negative-going PK events. One can choose a χ^2 selection criteria by checking the checkbox, choosing a comparison relation, and entering a value in the text box. Similarly, a "Magnitude" criterion may be selected. This refers to the magnitude estimate may by the non-linear fit and probably should be used in conjunction with found fit criterion.

Finally, one may choose only PK events detected by a particular "filter" in PKDetectorAnalyzer. This requires entry of values for the length of the filter in points, the threshold value in microvolts per point and the minimum number of points above threshold (see PKDetectorAnalyzer for more information).

In this example, note that there are three different PK detector panels opened, each using the same GV name of "QCChannels". Assuming that they refer to channels QC1, QC2, and QC3, the result would be a new GV of that name with the sum of the QC1, QC2, and QC3 PK events. No other sub-selection criteria were set.

6. Clock synchronization

One further feature of the Episode description block has to do with <u>clock synchronization</u>. Recall that there are two different clocks involved in data collection: one is the clock implied by the collected data (and Status channel) on the BioSemi machine and the second is the clock in the Presentation machine that is used to record times in the Event records in the Event file. The link between these two clocks is the Status channel, used by Presentation to mark the times of covered Events in the data record. In order to synchronize these clocks, the time recorded in an Event record needs to be correlated with its corresponding mark in the Status channel and used to establish a "beginning of file" time for the BDF data file. In theory, from that point on one can use the times in the Event records to find a corresponding point in the BDF file.

There are potentially two problems with this: the first is that there is an unknown (and perhaps variable) latency between the time that Presentation obtains the clock data to be placed in the Event record and the corresponding marking of the Status channel through setting of the Grey-code DIO value; and the second problem is that we have really only

"synchronized" the two clocks at the Event used for the synchronization, and from that point before and after, the clocks may be running at different tempos and can drift apart. The first of these problems can be minimized by careful Presentation program logic, but the second is more difficult to solve. We have previously noted significant drifts in the clock times. The clock on Presentation is software-based and may run slow if the CPU is under any computational pressure. There is supposed to be a clock correction routine in Presentation that periodically synchronizes its clock to an external standard using the so-called Network Time Protocol (NTP). The so-called BioSemi clock is in fact a hardware clock associated with the data collection hardware itself, and is thus not subject to software load exigencies.

Creation of "naked" Events is particularly susceptible to various clocking issues, despite the fact that they are based on the BDF data and are thus directly tied to the physiology. This is because we are forced to locate these Events on the basis of the Presentation clock only, which is the time saved in the Event record. If, for instance, a different synchronization Event is used for the creation of the naked Events than is used for the synchronization during ASC to FM conversion, there can be an offset between where the basis events are located in BDF and where they are counted in the conversion process. PK events may not be counted in the correct FM record! This would be less of a problem if the original Event markers all were carefully crafted to minimize the latency (or at least have a consistent latency) between marking the Status channel (writing the Event's Grey code) and querying the clock of the computer creating the Event record (the time recorded with the Event). This also assumes that relative clock drift is minimal (which may not be the case with most Windows-based computers).

That being said, in the short-trial (single FM record per trial) scenario, where timing is particularly critical, we have established a standard using the so-called "extrinsic Event" concept, where the true, real-world, Event is marked a second time in the BDF record to indicate the time of the corresponding Event on the BDF clock-schedule. Fortunately clock synchronization is less critical in ASC experiments as millisecond timing is not really required, and thus, generally, using the covered Event synchronization is adequate, if carefully implemented. The exception to this is noted in the previous paragraph where Events may be incorrectly accounted in the acquired FM records.

One of three (actually four) choices can be selected to handle the synchronization:
1. Synchronization to the first covered Event in the Event file;
2. Synchronization to an Event selected near the middle of the overall time of data collection (the so-called middle Event choice);
3. Use estimated synchronization only (none):
   a. Estimate the time between the beginning of BDF data collection and the first recorded Event in the Event file; or
   b. Use the times recorded in the Event file, which are thus assumed to be the actual times from the beginning of BDF data recording.

The advantage of option 1 is that a special Event might be created to provide synchronization, the creation of which could be carefully controlled. Option 2 has the advantage that it may minimize the offset caused by relative clock drift. Options 3 should be used only when no properly covered Events are available for synchronization. The three main options are indicated by this control.

Sync clock using: ◉ first covered Event   ◯ "middle" covered Event   ◯ none

Options 3a and 3b are selected in a special dialog box presented as the conversion actually begins.
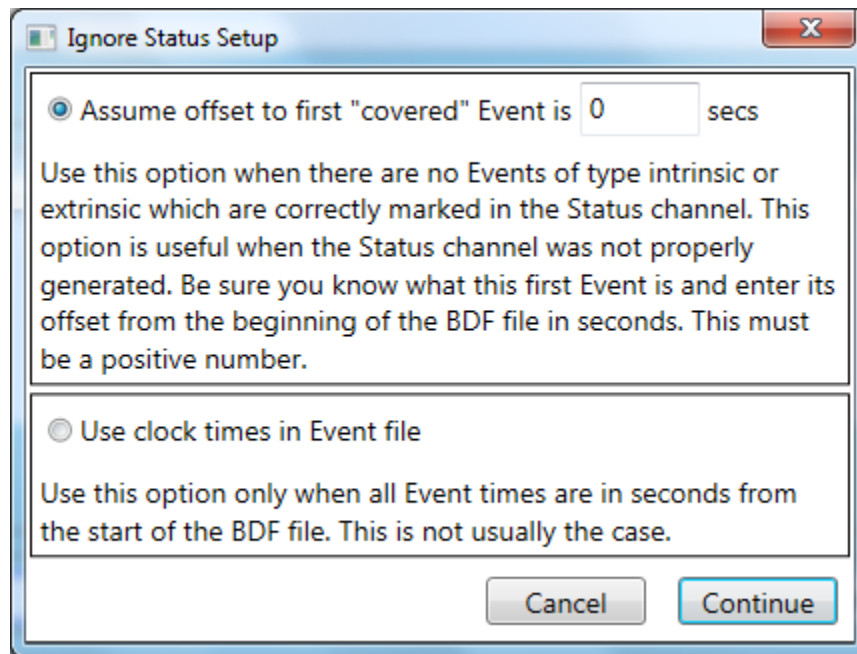
7. Group variable selection

The next panel in the main window is labeled "Group variables". Here you select the GVs to be included in the FM output file (so-called pass-through GVs). The values placed in the FM GVs are copied from the Event at the <u>beginning</u> of the episode, the "From Event". If this Event does not have a selected GV, a value of zero is placed in the GV in the FM file. Multiple GV may be selected. NB: portions of the current version of FILMAN may not work correctly with GVs equal to zero.

8. Other fields

The remainder of the main window is similar to the corresponding fields in FileConverter and use standard channel selection, sampling selection/detrending, and referencing selection that are used in that application. See FileConverter documentation and the UVaCollab wiki site for more information.

9. Performing conversion

Once all the fields have been correctly entered, conversion is initiated with the "Convert" button and progress will be displayed on the bottom status line. The conversion may be halted using the "Cancel" button with some portion of the FILMAN file, up to the point of interruption, intact for inspection.