Technical Report

From:	Robrecht Comeyne	***Doc ID:***	SLT.RJC02 252
To:	Acoustic Database Users	***Date:***	06/03/2009
Version:	3.2		

Re: **Acoustic Database Format Specification**

CONTENT

1. Introduction

Acoustic database acquisition outputs audio and logging files that adhere to certain rules described below. This does not mean that all possible information items are present in all databases but most of the applicable items are.

An utterance can be defined as a continuous recorded stream of human speech pronounced by a person at a certain moment in the time. All utterances are saved in separate **sample files** or **.WAV** files.

The record session is a set of more or less predefined utterances that are part of a predefined scenario, pronounced by a speaker in a certain environment under certain conditions. Because we gather all the utterances of a speaker in one record session, we can link the utterance data with more general information as recording circumstances and speaker typology. All these information items are stored in one structured text file called the **speech logging file** or **.SPL** file.

Both types of ADB output files have their own specific format that will be explained further.

Because we want to collect large amounts of acoustic speech data, we use a fixed common structure where it is possible to follow the genealogy of the recorded speech and to link the data information to the correct speech files. Therefore, an **ADB Tree Structure** is used that clearly defines how data emanating from the acquisition is stored.

2. The ADB Tree Structure

The ADB Tree Structure is based upon the following principles.

1. Store the Speech (.WAV) files and the information (.SPL) files on separate but complementary locations.
   Therefore, we need to separate the "data" and the "speech" part early in the tree structure. The reason for this is the difference in size between both output types: the audio files of a record session can easily take several megabytes of disk space while a data logging file takes only some ten thousands of bytes. We also have to consider the speech output as read-only data after acquisition. Utterance files contain a continuous stream of speech that cannot be altered or modified anymore. In many cases, the speech will end up to be stored on another site than the accompanying data files.

2. All the information of a Record session is logged in a single SPL file.
   All the utterances of a Record session are stored in separate audio files.

3. The tree structure must reflect the genealogy of the record session files.
   In this case, it is easy to link audio files to a SPL file or vice versa. This must be possible without the necessity to look into the file's contents.

The application of those principles leads to a deep tree structure that may look very complicated and redundant at the first sight, but it will prevent dramatic confusions once we deal with large amounts of speech data.

2.1 The Output Directory Path

The directory path ALWAYS has the following basic structure:

OUTPUTDIR=<root>\<data type>\<script>\<unit>\<group>\[<session>\]

2.1.1 Description

Directory	Format	Explanation
<root>	***<drive>[<dir>]***	The user of the acquisition tool specifies the <root>. It consists of a drive specification followed by a directory name
	<***drive***>=c:\ or d:\ or e:\ <***dir***>= root directory name	
<data type>	"***data***" or "***speech***"	Fixed directory name to indicate we are dealing with info data or speech
<script>	***"scr"<ssss>***	"scr" = fixed text indicating we are dealing with the 'script' subdirectory <ssss>= 0 aligned digit string containing the script ID
<unit>	***<uu>***	<uu>= 0 aligned digit string referring to the unit (workstation) number
<group>	***<ssss><uu><gg>***	<ssss> = 0 aligned digit string of the script ID <uu> = 0 aligned digit string of the unit number <gg> = 0 aligned digit string referring to a session subdivision in hundreds
[<session>]	***"r"<sss><rrrr>*** Only for the speech directories	"r" = fixed text: we deal with record session <sss> = 0 aligned digit string of the script ID <rrrr> = 0 aligned digit string of the record session number

2.1.2 Example

c:\db1\speech\scr0205\01\02050122\r2052130

<root>	c:\db1	Drive specification + directory name
<data type>	speech	Only speech data!
<script>	scr0250	Script 205 has been used
<unit>	01	We recorded using computer with unit number 1
<group>	02050122	Script 205, unit 1, group 22 (session 2101 - 2200)
<session>	r2052130	Session 2130 of script 205

2.2 The Speech or .WAV files

2.2.1 Examples

In most and certainly in recent databases, utterances are stored in separate sample files and the session logging file name is the same as the higher subdirectory:

d:\db1\speech\scr0205\03\02050322\r2052130**u2130001.wav**
d:\db1\speech\scr0205\03\02050322\r2052130**u2130002.wav**
d:\db1\speech\scr0205\03\02050322\r2052130**u2130003.wav**
d:\db1\speech\scr0205\03\02050322\r2052130**u2130004.wav**
…
d:\db1\speech\scr0205\03\02050322\r2052131**u2131001.wav**
d:\db1\speech\scr0205\03\02050322\r2052131**u2131002.wav**
d:\db1\speech\scr0205\03\02050322\r2052131**u2131003.wav**
d:\db1\speech\scr0205\03\02050322\r2052131**u2131004.wav**

For completeness the older way to store speech data in concatenated speech files is shown below (all utterances in one concatenated speech file):

d:\db1\speech\scr0205\03\02050322**r2052130.wav**
d:\db1\speech\scr0205\03\02050322**r2052131.wav**

2.2.2 The file name

SPEECHFILE=<name>.<extension>

<name>= *"u"<rrrr><uuu>* || *"r"<sss><rrrr>*
"u" = fixed character indicating we are dealing with one utterance per file
<rrrr> = 0 aligned digit string of the record session number
<uuu> = 0 aligned digit string of the utterance number

"r" = fixed character indicating we are dealing with an old concatenated file
<sss> = 0 aligned digit string of the script ID
<rrrr> = 0 aligned digit string of the record session number

<extension> = *"wav"*
"wav" = fixed extension name to designate NIST headed speech files (See further)

2.2.3 The file format

The ADB speech files carry a NIST header. This is an ASCII header elaborated by the *National Institute for Standardisation and Technology* in the United States and is commonly used in many speech laboratories all over the world.

The NIST header is 1024-byte blocked and composed of a fixed-format portion followed by an object-oriented variable portion.
The fixed portion is as follows:

NIST_1A<new-line>
1024<new-line>

The first line specifies the header type, the second line the header length in bytes. Each line is 8 bytes long, the <new-line> character included.

The variable portion is composed of object-type-value "triple" lines, which are always delimited by a new line character:

LINE = <OBJECT> <space> **<TYPE>** <space> **<VALUE>** [<space>]<new-line>

	FORMAT	COMMENTS
OBJECT	<STRING>	Use underscores instead of spaces
TYPE	-i	VALUE is integer
	-s<DIGITSTRING>	VALUE is string, the digit string represents the string length
VALUE	<STRING> |	If object type is -i
	<DIGITSTRING>	If object type is -s

The following fields are required for each NIST header

FIELD NAME	TYPE	DESCRIPTION
sample_count	integer	Number of samples per channel
sample_n_bytes	integer	Number of bytes per sample
channel_count	integer	Number of channels

The next fields are recommended or conditionally required. They are present in each ADB WAV file:

FIELD NAME	TYPE	DESCRIPTION
sample_rate	integer	Sampling rate in Hz
sample_coding	string	"pcm" | "mulaw" | "alaw"
sample_byte_format	string	"01"LSB, MSB (PC, Intel) | "10" MSB, LSB (Sun, Motorola) | "1" (1 byte samples, e.g. alaw, mulaw)

The following fields are optional but can be found in a ADB WAV file:

FIELD NAME	TYPE	DESCRIPTION
sample_checksum	integer	Checksum of sample portion
sample_sig_bits	integer	Number of significant bits
sample_min	integer	Lowest sample value in utterance (split file)
sample_max	integer	Highest sample value in utterance (split file)
speech_file_name	string	Name of speech file
database_id	integer	Script ID
speaker_session_number	integer	Record session number
session_utterance_number	integer	Utterance number
utterance_id	string	Transcription of utterance (split file)
utterance_count	integer	Number of recorded utterances
recording_date	string	Date of recording
recording_time	string	Time of recording
channel_<x>	string	Channel description. <x> refers to the channel number starting with 1. The field content is a copy of the corresponding SPL [Channels] section line

2.2.4 Example

NIST_1A □ 1024 □ sample_n_bytes -I 2 □ channel_count -I 1 □ sample_rate -I 16000 □ sample_coding -s3 pcm □ sample_byte_format -s2 01 □ speech_file_name -s39 u2130001.wav □speaker_session_number -s2 12 □ database_id -s1 5 □ sample_count -I 320000 □ utterance_count -I 10 □recording_date _s8 22/04/95 □ recording_time -s8 16:38:54 □ sample_checksum -I 25448 □ end_head □

2.3 The Speech Logging or .SPL files

The Speech Logging files are ASCII files that group all the information concerning a Record session.

2.3.1 Examples

d:\db1\data\scr0205\03\02050322**r2052130.spl**
d:\db1\data\scr0205\03\02050322**r2052131.spl**
d:\db1\data\scr0205\03\02050322**r2052132.spl**
d:\db1\data\scr0205\03\02050322**r2052133.spl**

2.3.2 The file name

INFOFILE=<name>.<extension>

<name>= *"r"*<*sss*><*rrrr*>
"r" = fixed character indicating we are dealing with a complete record session
<sss> = 0 aligned digit string of the script ID
<rrrr> = 0 aligned digit string of the record session number

<extension> = *"spl"*
"spl" = fixed extension name to designate a SPEECH LOGGING INFO file

2.3.3 The file format

2.3.3.1 The SPL sections

SPL files are ASCII files that contain all the information about a Record session collection during acquisition and validation. Those information data are included in several information blocks we call ***sections***. Each section contains specific information about one aspect of the collected speech data and is introduced with a title embraced by square brackets (**[...]**). Each section ends there where a new section title is found or at EOF. In more recent SPL's, the dummy section [End] is added at the end of the file.

The **SPL** file contains the following sections:

Section Title	Information type
[System]	Configuration of the acquisition
[Channels]*	Acquisition channels description
[Session]	Session related general information
[Record states]	Information about recorded utterances after acquisition
[Info states]*	Supplementary information entered in the info sheet
[Validation states]	Information about recorded utterances after validation

2.3.3.2 The Section lines

Each section is subdivided into several lines delimited with a *new line* character. Lines are introduced a ***key***. Keys are always separated from the rest of the line by an equal sign ('**=**') and specify the kind of information the line contains. We distinguish two types of keys:

1. ***Alphanumeric keys*** :
   Always introduce a unique information line
2. ***Numeric keys***:
   Represent the sequential number of a line within the section without further specification of the line content. In this case, the *section title* suffices to know about the kind of information we can expect.

A *section* can never mix the two types of information lines in its body. Or we deal with a section that contains only specific information lines, or we deal with a section that lists several lines containing the same type of information. In the last case, we also speak about ***records***.

Follows a list of the SPL sections together with the line types they contain and a brief description of the line contents:

Section	Type	Description
[System]	Info	Configuration information
[Channels]*	Record	Acquisition channel description
[Session]	Info	Session related information
[Record states]	Record	Description of the acquisition of one utterance
[Info states]*	Record	Description of one information item
[Validation states]	Record	Description of the validation of one utterance

2.3.3.3 The line fields

The *lines* within a section can be divided into one or more ***fields***, so that we can insert multiple information items in one line. Especially in case of *record* lines, this can be very useful. A delimiter string separates the fields. By default, this is a one-character string containing the semi-colon (**";"**), except in case a customized delimiter string is specified in the [System] section.

2.3.3.4 The field lists

Fields can contain ***lists.*** This means that a field can enumerate several strings. Those strings are separated by the vertical bar ('**|**') character. If no vertical bar character is present, the field is considered to carry a single string. Lists are used in case the number of elements is unknown or variable.

3. The Speech Logging file sections

3.1 The [System] section

The [System] section contains configuration information used to record the Record session utterances. Except from the *memo* field, the SPL [System] section is a copy of the SCRIPT [System] section.

3.1.1 Example

```
[System]
Script=205
Board=1;Multimedia
Frequency=16000
Coding=PCM;Linear
Channels=2
Delimiter=;
CharacterSet=ANSI
Memo=Microphone T45, echoing room, window is opened
```

3.1.2 Line table

Key	Fields	Description	Type
Script	1	Unique script ID	int[4]
Board	1	Board ID	int[2]
	2	Board description*	char[20]
Frequency	1	Sampling rate in Hz	long[]
Coding	1	Coding type	char[10]
	2	Compression type	char[10]
ByteFormat	1	Byte order	char[4]
Channels*	1	Number of acquisition channels used	int
Delimiter*	1	Delimiter string	char[]
CharacterSet*	1	Character set	char[10]
Memo*	1	Additional text describing record circumstances	char[1024]
Coupling*	1	Coupling string	char[10]
DOS Codepage	1	Local DOS code page	Int
ANSI Codepage	1	Local ANSI (Windows) code page	Int

3.1.3 Description

1. ***Script***:
   The unique SCRIPT ID number

2. ***Board***
   The board ID identifies the speech board used for acquisition. The description following this ID is optional in order to increase the readability of the line.
   Follows the list of possible board ID's:

   1 : Multimedia compatibles (e.g. Creative Labs™ Soundblaster)
   2 : NIDAQ compatibles (e.g. Nat. Instruments ™ AT_2200 or AT_2150 board)
   (**3 : LSI C25 board** obsolete)
   4 : Dialogic™ 41E multinational telephony board
   (**5 : Dialogic™ 41E** Wizard of Oz session obsolete)
   6: CAPI 2.0 compatibles (e.g. AVM™ ISDN controller)
   9: VX Pocket 440 (4 channel PCMCIA card)

3. ***Frequency***
   The sampling rate in Hz

4. ***Coding***
   The coding type is always
   PCM
   The compression type may be
   Linear
   Mu-Law
   A-Law

5. ***ByteFormat***
   A formal representation of the byte order used in the recorded speech files. Possible strings are:
   1 1 byte files (A-Law or Mu-Law)
   01 2 byte files - LSB/MSB (Linear PCM on PC)
   10 2 byte files - MSB/LSB (Linear PCM on SUN)

6. ***Channels***
   Total number of channels for recording. Default is 1. When the SCRIPT stipulates a multiple line recording, the output sample files are multiplexed and the [Record states] byte offsets always refer to multiplexed addresses.

7. ***Delimiter****
   The delimiter string for the SPL file fields. Each time the Speech Logging file is activated, it will first look for the delimiter line. If the delimiter line is not defined, the semi-colon string **";"** will be considered as the delimiter. An example of such delimiter is: **">-<"**

8. ***CharacterSet****
   Specifies which character set is used for the SPL text. The default character set is the local ANSI code page.
   The possibilities are:
   DOS in case of DOS extended ASCII character set
   ANSI in case of local ANSI code page

9. ***Memo****
   Additional text that allows to the collection operator to enter supplementary information concerning the circumstances in which the Record session has been done. The *Desktop Speech Data Recorder* allows the editing of a memo text.

10. ***Coupling****
    Coupling set during recording. The possible values are:
    AC AC coupling
    DC DC coupling

11. ***DOS Codepage***
    The DOS code page number of the acquisition workstation.

12. ***ANSI Codepage***
    The ANSI code page number of the acquisition workstation.

3.1.4 The [System] flags

Follows a set of optional entries that may be used in more specialized projects. The flags are represented by the value 0 (off) or 1 (on).

Key	Description
Male/Female differentiation	Make a different sheet attribution table for male and female speakers
Use Sheet ID as Session ID	Use the ID of the selected utterance sheet as record session ID. In this way, you ensure a one to one relation between sheet and session

3.2 The [Channels] section

Each line of the [Channels] section contains the description of an acquisition channel.
The key entry of each section line is a digit referring to the corresponding channel number. Acquisition channels are sequentially numbered starting with 1.
Since the channel configuration is always described in the SCRIPT file, this section is an exact copy of the [Channels] section of the SCRIPT file used for the creation of the record session.

3.2.1 Example

```
[Channels]
1=0 ; StarMic P1 ; Close talk
2=1 ; TalkElectronics E34 ; Far talk
```

3.2.2 Field table

NB	Field	Description	Type
1	**Channel offset**	Offset of a channel in multiplexed sample	int
2	**Microphone**	Brief description of the microphone used	char[]
3	**Type**	Brief description of the recording type	char[]

3.2.3 Description

1. ***Channel offset***:
   The offset of the channel sample within the multiplexed sample. For the byte offset within a multiplexed sample, the channel offset must be multiplied with the sample byte size.
   The first offset is always 0.
   For instance, suppose we recorded 16-bit samples at four channels. This means that each multiplexed sample has a size of ***sample size * number of channels***, in this case 2 * 4 = 8 bytes. Normally, the channel offsets are attributed to the channels in numeric order. So, we find the first offset at byte position 0 (0 * 2), the second at position 2 (1 * 2), the third at position 4 (2 * 2), etcetera

2. ***Microphone***:
   Contains a brief description of the microphone used for acquisition.

.

3. ***Type***:
   Contains a brief description of the acquisition type.

3.3 The [Session] section

The [Session] section contains general information related to the Record session. An alphanumeric key introduces each information line.
Remark that the [Session] section can contain entries that are not defined here, but that are inserted as additional information to some acquisition projects.

3.3.1 Example

```
[Session]
Number of recordings=8
RecDate=18/10/95
RecTime=12:06:54
ValDate=25/11/95|03/02/96
Operator=Bill Clinton|Boris Jeltsin
```

3.3.2 Line table

Key	Fields	Description	Type
Number of recordings	1	Total number of recorded utterances	int
RecDate	1	Date of recording	char[10]
RecTime	1	Time of recording	char[10]
RecDur	1	Duration of Record session	char[10]
ValDate	1	Validation date list	char[]
Operator	1	Validation operator list	char[]
Imported sheet file*	1	Imported sheet file path	char[]
Sheet number	1	Sheet file number	Int
Synchronization mark	1		Int
Channel	1	Input channel number	Int

3.3.3 Description

1. ***Number of recordings***
   Total number of recorded utterances during the record session. This number should represent the total number of [Record states] lines.

2. ***RecDate***
   Date when Record session was done. Format: DD/MM/YY

3. ***RecTime***
   Time when Record session was done. Format: HH:MM:SS

4. ***RecDur***
   Duration in minutes and seconds of a Record session, from the display of the information sheet until the recording of the last utterance.

5. ***ValDate***
   The last date each operator of the *operator* field list worked on the Record session. The date strings are separated with a vertical bar ('|').
   Format: DD/MM/YY

6. ***Operator***
   A list of the last three operators who validated the SPL file. The operator names are separated by a vertical bar ('|').

7. ***Imported Sheet File****
   The prompt sheet file pathname if DSDR used a prompt sheet file as supplementary input.

8. ***Sheet number****
   The sheet number indication the sheet used in the DSDR

9. ***Synchronization mark****
   Time elapsed in milliseconds between the last DTMF beep and the record start beep end. Only used for SpeechDat-Car recordings.

10. ***Channel****
    The input channel number used for recording.

3.4 The [Record states] section

Each line of the [Record states] section logs the results of an utterance session. Once logged, a [Record states] line can never be modified anymore. Later modifications or additions about the utterance info will be stored in another SPL information block: the [Validation states] section.
Originally, the [Record states] lines contained six fields that are mandatory. The other fields have been added later.

3.4.1 Example

```
[Record states]
1= 2 ;  Start  ; Start  ; 1024  ;  29312  ;  u0306001.wav ;  start.voc ; -1 ; -1 ; IWa ; ; ;
2= 2 ;  Stop  ; Stop  ; 1024  ;  28408  ;  u0306002.wav ;  stop.voc ; -1 ; -1 ; IWa ; ; ;
3= 2 ;  Continue  ; Continue  ;  1024  ;  36874  ;  u0306003.wav  ;  cont.voc ; -1 ; -1 ; IWa ; ; ;
4= 2 ;  Österrike ; Österrike ; 1024 ; 487424 ;  u0113009.wav ; ;  -1 ; -1 ; IWn3 ; ; ;
5= 2 ;  6 ; 6 ; 1024 ; 385024 ; u0113116.wav ; ; -1 ; -1 ; ID1 ; ; ;
6= 2 ;  4 ; 4 ; 1024 ; 385024 ; u0113117.wav ; ; -1 ; -1 ; ID2 ; ; ;
```

3.4.2 Field table

NB	Field	Description	Type
1	**Type**	Type ID of the record state	Int
2	**Name**	Utterance name	char[]
3	**Text**	Utterance presentation text	char[]
4	**Begin**	Absolute begin offset of the utterance segment	Long
5	**End**	Absolute end offset of the utterance segment	Long
6	**File**	File where recorded utterance is stored	char[15]
7	**Old file***	Original file name of the utterance	char[15]
8	**Comp Beg**	Complementary begin offset of utterance	Long
9	**Comp End**	Complementary end offset of utterance	Long
10	**Sentence Definition***	Sentence definition string	char[10]
11	**Utterance ID***	Database ID string (e.g. as used for a phonetic transcription tool)	char[65]
12	**Comment***	Comment string	char[]

3.4.3 Description

1. ***Type*** :
   Record state type ID. We distinguish four types:

   - **0: dummy**
     The utterance has not been recorded. Future read operations should ignore this record[1]
   - **1: Spontaneous**
     The utterance is a response to a question. The content is not known beforehand.

[1] The 'dummy' record state has been introduced to guarantee a one to one relationship between the SPL and SCRIPT record states. In this way, each SPL record state keeps its original ranking position and numeric key entries are not needed anymore

- **2: Fixed**
  The speaker read the presentation text as it was printed on screen or paper.
- **3: dynamic utterance**
  The utterance label is created at run-time

2. ***Name***:
   The interpretation of the field depends on the state type:

   - **0: dummy**
     Any of the following fields
   - **1: Spontaneous**
     Describes the spontaneous utterance.
   - **2: static utterance**
     The transcription text adopted from the SCRIPT file.
   - **3: dynamic utterance**
     The label sentence name adopted from the SCRIPT file

3. ***Text***:
   The text as it has been presented to the speaker

4. ***Begin***
   Absolute begin offset in bytes of the utterance segment in the speech file.

5. ***End***
   Absolute end offset in bytes of the utterance segment in the speech file.

6. ***File***
   Name without directory specification of the utterance audio file

7. ***Original file****
   The name of the file where the utterance was originally stored before it was converted into the ADB Speech Data File format. In [Record states] lines created with the SCANSOFT acquisition tools, this field is not used.

8. ***Comp Begin***
9. ***Comp End***
   The absolute begin and end offsets in bytes of the utterance segment for the complementary speech collection mode. Old-fashioned data acquisition tools gathered all the record session utterances in one concatenated speech file. More recently, all utterances are stored in separate speech files. Nevertheless, the acquisition tools keep track of both acquisition modes. The ***Begin*** and ***End*** fields contain the offsets where the speech has been actually stored. The ***Comp Begin*** and ***Comp End*** fields contain the offsets of the opposite speech collection mode. These offsets are calculated according to the speech data specifications enclosed in the [System] section.
   From ADB Tool version 6.4 on, these fields may contain –1, which implies that the concatenated offsets are ignored.

10. ***Sentence definition***
    A code string of up to 10 characters (typically 1 to 3 char.) that is used as an alias of the utterance label. This is especially used in case the label content can vary, namely in acquisition projects where sentence sheets where used. We define a *sentence definition* as a combination of an **utterance type** and a **ranking number**. The latter is only used the make a sentence definition unique within a SCRIPT file. For the utterance type, we have defined a limited number of code strings that are use in most of the recent databases. The table below gives an overview of the sentence definitions and code strings that may be used. The sentence definition allows us to link the utterance with variable text lines residing in the utterance sheet file.
    This string is defined in the original SCRIPT and is unique within the [Record states] section.

11. ***Utterance ID***
    A string of up to 64 characters that contains a unique utterance identifier.

12. ***Comment****
    Additional comment string.

Sentence definitions and code strings that may be used:

Type	Description	Example(s)
CA	Continuous Alphabet letters	Y M C A
CD	Continuous Digits	9 1 1
CSa	Continuous Sentences application based	Select the first line. Make it bold and move it to the bottom of the text…
CSp	Continuous Sentences phonetically based	One upon a time, in a land far from here, lived a little princess. She was the most beautiful girl…
FN	Free format Number	09/239 8000
Form_ID	Form identification number <UUUU><SSSSSS><CC>	000154789566
IA	Isolated Alphabet	B
ID	Isolated Digit	5
IN	Isolated Number	1999
ISa	Isolated Sentence application based	Insert this name in my list.
ISp	Isolated Sentence phonetically based	A cold supper was ordered and a bottle of port.
IWa	Isolated Word application based (short command)	Stop Other examples: Left, Print etc.
IWn	Isolated Word: name(s)	Microsoft, New York, Jonathan
IWp	Isolated Word phonetically based	Apple tree Other examples: Lobster, Jogging etc.
Sheet_ID	Sheet ID number	012589225712
SS	Spontaneous Speech: continuous	Spontaneous speech typically resulting from a task given to the speaker.
SY	Syllable(s)	chi (as in: Hi-ta-chi)
T	Time/date expression	25th June 1964
CW	Continuous Words	Apple tree, Lobster, Jogging, Ready, Perfect, Additional
BN	Background Noise	Speaker does not speak; can be used to perform measurements using artificial signals

3.5 The [Info states] section

The [Info states] section lines contain the information items the speaker entered in the *Info Sheet*.

3.5.1 Example

[Info states]
1= Speaker ; 3120
2= Name ; James Joyce
3= Phone ; 025/874569
4= Age ; 85
5= Region of Birth ; Ireland
6= Region of Youth ; Ireland
7= Sex ; Male
8= Native ; TRUE

3.5.2 Field table

NB	Field	Description	Type
1	**Item name**	Name of the information item	char[20]
2	**Text**	Information string	char[]

3.5.3 Description

1. ***Item name***
   Brief description of the information item adopted from the SCRIPT file. Some Item strings are

protected. For a complete list of these strings, see Appendix B of this document.

2. ***Text***
   Informative text entered in the information sheet

3.6 The [Validation states] section

The [Validation states] section covers all the utterance information added after the speech acquisition. The goal of this information block is double:
1. Enlarge the basic utterance information provided by the [Record states] section
2. Prevent that the original data after acquisition is overwritten. In this way, it is always possible to overrule the validation work and to restart again with the original data.

Because the validation state line is a complement to a record state line, the [Validation states] section must contain as many lines as the [Record states] section.

The validation state line contained originally 12 fields. The other fields have been added later.

3.6.1 Example

[Validation states]
1= james joyce ;4028 ; 29060 ; QUA:B ; NOI:0 ; SND:0 ; SPC:0 ; UTT:0 ; DST:0 ; u0205001.wav ; 4028 ; 29060; ; ; ;
2= zero two five eight seven four five six nini ; 41512 ; 103666; QUA:B ; NOI:0 ; SND:0 ; SPC:0 ; UTT:0 ; DST:0 ; u0205002.wav ; 9512 ; 71666 ; ; ; ;
3= yes#/#jEs ; 135842 ; 148300 ; QUA:D ; NOI:0 ; SND:0 ; SPC:0 ; UTT:1 ; DST:0 ; u0205003.wav ; 7842 ; 20300; Not enough trailing silence; ; ;
4= no#/#no: ; 162600 ; 175026 QUA:B ; NOI:0 ; SND:0 ; SPC:0 ; UTT:0 ; DST:0 ; u0205004.wav ; 12600 ; 25026; ; ; ;
5= hello#/#hElo ; 189116 ; 201112; QUA:C ; NOI:0 ; SND:1 ; SPC:0 ; UTT:0 ; DST:0 ; u0205005.wav ; 7116 ; 19112; door was opened; ; ;
6= to be or not to be that is the question; 217478 ; 265128 ; QUA:B ; NOI:0 ; SND:0 ; SPC:0 ; UTT:0 ; DST:0 ; u0205006.wav ; 3478 ; 51128; ; ; ;
7= one seven one eight#/#wan sEven wan Ejt ; 290800; 302408 ; QUA:B ; NOI:0 ; SND:0 ; SPC:0 ; UTT:0 ; DST:0 ; u0205007.wav ; 2532 ; 14140; ; ; ;
8= office eight; 336150 ; 371054; QUA:B ; NOI:0 ; SND:0 ; SPC:0 ; UTT:0 ; DST:0 ; u0205008.wav ; 10150 ; 45054; ; ; ;

3.6.2 Field table

NB	Field	Description	Type
1	**Transcription**	Orthographic and/or phonetic transcription	char[]
2	**Begin**	Validation begin offset	long
3	**End**	Validation end offset	long
4	**Quality label**	Quality label	char[5]
5-9	**Quality switches**	Quality switches (5)	char[5]
10	**Split file name**	File name in case of one utterance per file (=split file)	char[15]
11	**Split begin**	Begin offset in case of a split file	long
12	**Split end**	End offset in case of a split file	long
13	**Comment**	Additional comment line	char[]
14	**Events**	Event table	char[]
15	**Score**	Automatic Recognition result	float

3.6.3 Description

1. ***Transcription*** ***<orthographic string>["#/#"<phonetic string>]***
   Contains the orthographic and/or phonetic transcription of the utterance. By default, it is the transcription implemented in the *text* field of the complementary [Record states] line, but the validation operator can edit this text.
   The *transcription* differs substantially from the *presentation text,* because it represents the text the speaker really pronounced in the utterance. In consequence, all corrections must be put in the transcription field while the [Record states] text field remains untouched. Another difference with the *presentation text* is that transcriptions contain only fully spelled text. Abbreviations, numbers, dates, digit strings, etc. are edited as text. Finally, the transcription may in certain cases also contain event annotations. *Events* are small sequences within an utterance, which are not speech.

Events may occur in background – e.g. a passing car – or can be caused by the speaker himself, like coughs and fillers. Events are inserted as small text blocks embraced by curly brackets ({ }). For more information about events, see the specific database documentation if applicable.
The orthographic and phonetic strings are separated by a dedicated string: **"#/#"**.

2. ***Begin***
The validation or fine-tuned concatenated begin offset. By default, it is the same as the [Record states] begin offset for concatenated mode, but the validation operator can move it forwards in order to fine-tune the segment and logically remove redundant leading speech. See also (11). From ADB Version 6.4 on, this field may contain –1 that means that the concatenated offset is unknown and should be ignored.

3. ***End***
The validation or fine-tuned concatenated end offset. By default, it is the same as the [Record states] end offset for concatenated mode, but the validation operator can move it backwards in order to fine-tune the segment and logically remove redundant trailing silence.[2] See also (12) From ADB Version 6.4 on, this field may contain –1 that means that the concatenated offset is unknown and should be ignored.

4. ***Quality label "QUA:"<Quality label character>***
The general quality label attributed to the utterance. When the utterance is not yet validated, the quality label is **'X'**.

In case a human evaluates the utterances, we distinguish five labels represented by the first five letters of the alphabet:

'A'	Exceptional or excellent quality
'B'	Normal good quality
'C'	In between good and bad quality
'D'	Bad quality but still intelligible
'E'	Unacceptable quality because unintelligible or wrong

In case the validation state has been processed with a Automatic Recognition Engine, the label can adopt the following letters:

'R'	Recognition with a confidence level higher or equal than the threshold
'P'	Recognition with a confidence level lower than the threshold
'N'	No recognition

Normally, the *confidence level* and the *confidence level threshold* are kept in field 15 of the validation state. The *confidence level* is a value between 0 and 100 that represents the confidentiality of the recognized sample. The *threshold* is the minimum value the confidence level should have to consider the utterance as 'fully' recognized.

5. ***Noise switch "NOI:" 1|0***
The noise quality switch indicates whether or not there is a continuous background noise in the signal.

6. ***Sound switch*** **"SND:" *1|0***
The sound switch indicates if there is an unusual sound detectable in the background.

7. ***The speaker switch*** **"SPC:" *1|0***
Indicates if there is another person than the target speaker speaking in the background.

8. ***The utterance switch*** **"UTT:" *1|0***
This switch concerns the content of the utterance segment. It indicates that the utterance is not understandable or incomplete or that the content does not match the label.

9. ***The distortion switch*** **"DST:" *1|0***
The signal is distorted: e.g. the recorded utterance contains detectable signal clipping

[2] In our approach, leading and trailing silences are never physically removed from the file because we want to keep an utterance in its original state after acquisition

10. ***Split file name***
    This field contains the name of the sample file. In case of a *split* acquisition, the sample file name is the name of the utterance file. In older databases recorded in *concatenated* mode, the file name field contains the utterance file name as if the database was recorded in split mode.

11. ***Split begin***
    The validation or fine-tuned split begin offset. By default, it is the same as the [Record states] begin offset for split mode, but the validation operator can move it forwards in order to fine-tune the segment and logically remove redundant leading speech. See also (2).

12. ***Split end***
    The validation or fine-tuned split end offset. By default, it is the same as the [Record states] end offset for split mode, but the validation operator can move it backwards in order to fine-tune the segment and logically remove redundant trailing silence. See also (3).[3]

13. ***Comment****
    A comment string which adds supplementary information to the utterance report. The validation operator edits this comment field.

14. ***Event segment list****
    This character string contains the extra event segment descriptions. Each event contains two parts: the *event symbol*, possibly accompanied with a phonetic transcription, and the *time alignment* of the segment. Both parts are separated by a colon (**':'**). Several event segments can be concatenated with a vertical bar (**'|'**). The event symbol also appears in the transcription orthographic text embraced with curly brackets ({}).
    Further description of the extra speech events is found with the databases in case the extra event annotation is used.

15. ***Score****
    The score of the recognition.

3.6.4 Use of multiple [Validation states] tables

It is possible to insert different [Validation states] tables within a Speech logging file. This is particularly useful but rarely used in case you wish to validate multiple channel samples and you want to keep track of the validation results of each separate channel.

The first part of an additional Validation section is identical to the base section (**"Validation states"**) but extended with a ranking number starting with '2':

[Validation states < ID>] where

<ID> represents the validation section ranking number

According to multiple [Validation states] tables, three important remarks must be made:
1. The [Validation states] section (without number extension) always refers to the 'base' validation (this means: first channel, first validation session) and corresponds with the non-existing section [Validation states 1].
2. Each Validation section has a 'source' table. This is the table that has been originally used to create the Validation states section. By default, the 'source' table of a Validation states section is the [Record states] section. This means that a validation starts with the acquisition results. But also each other Validation section within the Speech Logging file can serve as 'source' of a new Validation table. The source of the Validation table is stored in the [Validation specification <ID>] section.
3. You can specify the channel upon which the validation is based in the [Validation states <ID>] section.

[3] The presence of two offset pairs may seem bizarre but dates from the time databases used concatenated speech files. In practice, both pairs are updated simultaneously when the validation operator decides to move the segment offset border, regardless of the fact he is working with split or concatenated speech files. For recent databases, field 11 and 12 are more important than their concatenated counterparts in fields 2 and 3.

3.7 The [Validation specifications <ID>] section

The [Validation specifications <ID>] section contains information about the requirements the operator has to respect during the utterance validation. <ID> is the ranking number of the Validation section is referring to. If the Validation specifications section refers to the base Validation section, <ID> is not mentioned.

3.7.1 Example

```
[Validation specifications]
LeadingSil=200|100
TrailingSil=200|100
Orthographic Events=TRUE
Phonetic Events=FALSE
PhonAlphabet=L&H+
Engine=ASR1600v1
Models=Fre_Fv1.1
Threshold=1.3
Source table=0
Channel=1
```

3.7.2 Line table

Key	Fields	Description	Type
LeadingSil	1	Leading silence and tolerance	intlint
TrailingSil	1	Trailing silence and tolerance	intlint
Orthographic Events	1	Orthographic event annotation flag	char[5]
Phonetic Events	1	Phonetic event annotation flag	char[5]
PhonAlphabet	1	Phonetic alphabet set	char[20]
Engine	1	ASR Engine used	char[25]
Models	1	Training models	char[25]
Threshold	1	Acceptance threshold	float
Source table	1	Source section	Int
Channel	1	Channel	Int

3.7.3 Description

1. ***LeadingSil*** ***<leading silence>|<tolerance>***
   The leading silence in milliseconds required before the beginning of the utterance.
   The second part of the field contains the fault tolerance in milliseconds.

2. ***TrailingSil*** ***<trailing silence>|<tolerance>***
   The trailing silence in milliseconds required before the end of the utterance.
   The second part of the field contains the fault tolerance in milliseconds.

3. ***Orthographic Events*** ***TRUE or FALSE***
   If TRUE, the operator annotates the signal and the orthographic utterance transcription with extra speech event segments (See Appendix A).

4. ***Phonetic_Events*** ***TRUE or FALSE***
   If TRUE, the operator annotates the signal and the phonetic utterance transcription with extra speech event segments (See Appendix A).

5. ***PhoneAlphabet***
   Phonetic alphabet used for the phonetic transcriptions in the [Validation states] section.

6. ***Engine***
   ASR engine used for automatic validation

7. ***Models***
   Training models used for automatic validation

8. ***Threshold***
   Acceptance threshold used for automatic validation

9. ***Source table***
   The source table is the validation table used to create the table <Validation section> is referring to.
   The value can be:
   0 if based upon the [Record states] table
   1 if based upon the [Validation states] 'base' table
   <other value> referring to the Validation section ID

   The default source table is 0 ([Record states] table)

10. ***Channel***
    The channel on which the validation is based. The default channel is 1.

3.8 The [Operators] section

The [Operators] section provides information about the operators working on the different validation tables.

3.8.1 Example

```
[Operators]
1=JV; 19 oct 1997;29 oct 1997
2=EG; 4 dec 1997;12 dec 1997
3=JV|2_1; 10 mar 1998;25 mar 1998
```

3.8.2 Field table

NB	Field	Description	Type
1	**a. Operator**	Operator name or ID	char[]
	b. Table ID	Validation state section ID	int
2	**Begin date**	First validation date	long
3	**End date**	Last validation data	long

3.8.3 Description

1. ***Operator/Table ID*** ***<Operator ID>[|<Val. Table ID>]***
   The operator ID or name. Generally, this is entered in an input box in the application.

   If the operator is working on a Validation table other than the base table, the table ID is added to the field.

2. ***Begin date***
   Date where the operator began the first validation.

3. ***End date***
   Date where the operator did the last validation

