Abstract

This document deals with the formats and naming conventions to be used for the data collection in VERBMOBIL II.
First we will describe the conventions for file naming, show the file structure of a VERBMOBIL CD−ROM and what it shall content.
We will define in more detail the format of the NIST header of a signal file, the contents of a marker file and a recording protocol file and how a speaker protocol file looks like.

Table of Contents

1. File Names

a. General

The naming follows the "DOS" convention: not more than 8 letters for the name and 3 for the extension (exception: name of the partitur file has a longer name).

b. Naming

Table 2 and table 3 "All Name Formats of a Dialog" show the names of all the files which are related to a recorded dialog. The file names can be separated in segments with different meanings containing detailed information about the concerned file.
The information is coded as follows, related to the headlines of each column of the table:

Language ::= [gejm]

where

g: language of the dialog is German
e: language of the dialog is English
j: language of the dialog is Japanese
m: the dialog partners use different languages

Within the dialog name, the first letter indicates the language of the conversation.

DialogNumber ::= [0−9][0−9][0−9]

Each dialog has an identification number.

Scenario ::= [abcd]

where

a: main scenario
b: information desk
c: remote maintenance
(d: scenario according to VERBMOBIL1)

In this position the experimental design as one of four different scenarios is coded.

RecordSetup ::= [crt]

where

c: close microphone
r: room microphone

t: telephone

Within RecordSetup the recording tool is indicated.

```
ChannelSpecification ::= [hncmpwdr]
```

If RecordSetup = t   /* telephone*/

```
ChannelSpecification ::= [mpwd]
```

where

m: mobile phone
p: analogue phone
w: wireless phone
d: dect

If RecordSetup = r   /* room microphone*/

```
ChannelSpecification::= [r]
```

where

r: room (there is no specification for the moment)

If RecordSetup = c   /* close microphone */

```
ChannelSpecification::= [chn]
```

where

c: clip microphone
h: headset microphone
n: neckholder microphone

```
ChannelNumber ::= [1−9]
```

The channel number shows to which recording channel the transliterated turn relates.

```
TurnId ::= TurnCount SpeakerId
```

The TurnId gives information about the count of the turn within the dialog and contains the SpeakerId. The TurnId is separated from the ChannelNumber with an underline. Between the TurnCount and the SpeakerId is a further underline.

```
TurnCount ::= "_" [0−9][0−9][0−9]
```

The TurnCount consists in three digits starting with 000 for the first turn.

```
SpeakerId ::= "_" [A−Z][A−Z][A−Z]
                     (also "_" [a−z][a−z][a−z] in case of speaker protocol file
                      name)
```

The SpeakerId consists in three capital letters (no umlauts) and shall be an unambiguous identification of the speaker of a certain native language, also in related databases. For the name of the speaker protocol file, lower case letters shall be used instead of capital letters.

```
CdromId ::= "_" [0−9][0−9][0−9][0−9][0−9][0−9]
```

CdromId is initialized with an underline and identifies the CD−ROM volume (2 characters) and edition (2 characters) which contains the dialog (it must correspond to the CDR declaration of the header of the transliteration file)
0000 (4 characters), e.g. 1200 (CD−ROM No.12)
Additionally the version of the Transliteration (in case of up−dates) is coded in the fifth and sixth digit (this version number corresponds to the TRV code from the header of the transliteration file)
00 (2 characters), e.g. 02 (second up−date).
The values for CdromId shall be set from the CD−ROM provider.

```
Extension: ".rpr" | ".trl" (| ".mix" | ".jpn" | ".rmn") | ".mar" | ".16" | ".al" | ".ul" |
                ".par" | ".spr"
```

The extensions code the contents of the file:

extension	***explanation***
.rpr	recording session protocol
.trl (also .mix .jpn .rmn)	transliteration file (in case of Japanese, the trl−files contains the Roman transcription separated into words (− file contains Kanji−kana and Roman transcription, − file contains Kanji−kana transcription separated into phrases − file contains Roman transcription separated into phrases)
.mar	turnmarker
.16	16 kHz 16 Bit signal
.al	alaw signal
.ul	ulaw signal
.par	symbolic information in "partitur" format
.spr	speaker protocol file

Table 1

```
TurnLanguage ::= [|_ENG|_GER|_JAP]
```

where

"" : no TurnLanguage necessary
_ENG : TurnLanguage is English
_GER : TurnLanguage is German
_JAP : TurnLanguage is Japanese

TurnLanguage is the language that is spoken in a certain Turn of a multilingual dialogue. It is only used in Turnmarkerfiles.

If Language != m /*that means e, g or j*/

```
TurnLanguage ::= ""
```

where

"" : no entry

In this case TurnLanguage is not necessary, because the language of all turns is either english, german or japanese.

If Language = m /*multilingual*/

```
TurnLanguage ::= "_" [ENG|GER|JAP]
```

where

ENG : TurnLanguage is English
GER : TurnLanguage is German
JAP : TurnLanguage is Japanese

In this case TurnLanguage codes the language that is spoken in a certain Turn of a multilingual dialogue.

c. All Name Formats of a Dialog

Level	***FileName of:***	***Lan−guage***	***DialogNumber***	***Scenario***	***Record Setup***	***Channel Specification***	***Channel Number***	***TurnId***		***CDromId***	***Ex−tension***
								TurnCount	***SpeakerId***		
		[gejm]	[0−9][0−9][0−9]	[abcd]	[c] [r] [t]	[chn] [r] [mpwd]	[1−9]	_[0−9][0−9][0−9]	_[A−Z][A−Z][A−Z]	_[0−9][0−9][0−9][0−9][0−9][0−9]:	
Directory Level	Directory	**g**	**101**	**a**							
	Recording Session Protocol	**g**	**101**	**a**							**.rpr**
Dialog Level	*Transli−teration*	**g**	**101**	**a**	**c**						**.trl**
	Turn Marker	**g**	**101**	**a**	**c**						**.mar**
Channel Level	16kHz 16Bit Signal	**g**	**101**	**a**	**c**	**n**	**1**				**.16**
	Alaw Signal	**g**	**101**	**a**	**c**	**n**	**1**				**.al**
	Ulaw Signal	**g**	**101**	**a**	**c**	**n**	**1**				**.ul**
Turn Level	*Partitur Format*	**g**	**101**	**a**	**c**	**n**	**1**	**_003**	**_AAA**		**.par**
	Turn Name	**g**	**101**	**a**	**c**	**n**	**1**	**_003**	**_AAA**	**_150000:**	

Table 2 shows all units necessary for coding the different file names of one dialog in the shaded fields. The examples in bold letters at the shaded fields show a (**g**)erman dialog with the number **101** in scenario **a** recorded with a (**c**)lose microphone of the type (**n**)eckholder. The channel number of speaker **AAA** is **1**, it is the fourth turn (**003**) of the dialog (note: turn numbers start at 000). The dialog is published on CD−ROM volume **15** and it is the first version of the transliteration (_1500 **00**). All the files with the name in italics (*Transliteration, Partitur format* and *Turn Name*) will be published on the VERBMOBIL server at the DFKI − Saarbrücken, the other files are stored on the CD−ROM and can be found in the directory named **data**.

Speaker Protocol(spr)

Directory Name	**spr**										
Speaker Protocol File		**g**							**_aaa**		**.spr**

Table 3: The speaker protocol files are also stored on the CD−ROM in the directory named **spr**. The data of a single speaker are collected in a file named with one letter for the **language** separated with an **underline** of the **three letters** for the **speaker identification**. Note, that in this case we use **lower case letters** for the speaker identification. The extension of a speaker protocol file is ». **spr**«. Therefore, the data of the German speaker AAA of the example are stored on the CD−ROM in **spr/g_aaa.spr**.

2. Data Distribution

a. General

One part of the recorded data will be distributed on the VERBMOBIL server at the DFKI Saarbrücken (transliteration files) and the other part on CD−ROM (signal files and files closely related to the signal files).

b. Logical Structure of a CD−ROM

1) File System

The file system of the CD is iso9660.

2) Directory Names

The CD−ROM contains two main directories:
data
spr
and the information text in **readme.**

3) CD−ROM Structure

A set of files for one German dialog between two speakers using close and room microphones as well as telephones simultaneously would consist of the following files:

	Level 1	***Level 2***	***Level 3***			
(infotext)	**readme**					
(speaker protocols)	**spr/**	**g_aaa.spr** **g_aab.spr** **...**				
(dialog directories)	**data/**	**g001a/**		*(signal)*	*(turnmarker)*	*(recording protocol)*
			(close)	**g001acn1.16** **g001acn2.16**	**g001ac.mar**	**g001a.rpr**
			(room)	**g001arr1.16** **g001arr2.16**	**g001ar.mar**	
			(tele)	**g001atm1.al** **g001atp2.al**	**g001at.mar**	
		g002a/				
			(close)	g002acn1.16 g002acn2.16	g002ac.mar	g002a.rpr
		g0...				

Table 4

Table 4 shows the file allocation on the CD−ROM. The names in bold letters are examples for a possible distribution of one dialog. Speaker information files are situated in the **spr** directory, each recorded dialog has its own directory in the **data** directory. In these dialog directories, all the files related to one dialog recording are stored (signal files, turnmarker files and the recording protocol file). Depending on the recording site, there may exist parallel recordings done by means of different recording setups (via telephone, room microphone and close microphone). This explains the three different sets of signal and marker files of the example above. The minimum version of files of a recorded dialog would consist in only one set of close or room or tele files.

3. Signal File Formats

a. Physical Signal Characteristics

Signal files containing room or close microphone data are coded in the following format:
16 bit, 16 kHz, mono, linearly coded (pcm), little endian (intel byte order) .
Files containing telephone data are coded:
8 bit alaw, 8 kHz, mono.

b. Logical Signal Characteristics

Each signal file contains one complete recording of a dialog session of one speaker by means of one of the recording setups (room−, close−microphone, telephone).

c. NIST Header Field Definitions

The signal files begin with a header following the NIST conventions. It has a defined size of 1024 bytes and consists of ASCII characters.
The format is as follows:

Key	***type***	***description***	***(possible) value(s)***
database_id	string	database	VERBMOBIL2
database_version	string	version	1.0
scenario_language	string	recorded language	[german|english|japanese|multi_english|multi_german|multi_japanese|multi_english_german|multi_english_japanese|multi_german_japanese]
scenario_id	string	scenario	[main|information_desk|remote_maintenance|vm1]
dialog_id	string	# of dialog	[000−999]
speaker_id	string	speaker	[AAA−ZZZ]
recording_site	string	site	[CMU|LMU|ATR|UBN|UHH]
recording_medium	string	rec. medium	[telephone|room|close]

Key	***type***	***description***	***(possible) value(s)***
recmed_spec	string	spec. of mic/tel−type	[mobile|wireless|analog|neckband|dect|headset|clip|room]
sample_coding	string	coding	[alaw|ulaw|pcm]
sample_n_bytes	int	bytes/samle	1,2,...
channel_count	int	# of channels	1,2,...
sample_count	int	# of samples	(total number of samples)
sample_byte_format	string	little/big endian one byte	[01|10|1]
sample_rate	int	samp. freq	
scenario_date	string	logical date of recording	(YYMMDD) e.g.: 980101

Table 5

The **remaining** bytes are filled with **spaces**.

Example header: g010acn1.16

```
NIST_1A
   1024
database_id −s10 VERBMOBIL2
database_version −s3 1.0
scenario_language −s6 german
scenario_id −s4 main
dialog_id −s3 010
speaker_id −s3 ABA
recording_site −s3 LMU
recording_medium −s5 close
recmed_spec −s8 neckband
sample_coding −s3 pcm
sample_n_bytes −i 2
channel_count −i 1
sample_count −i 124798
sample_byte_format −s2 01
sample_rate −i 16000
scenario_date −s6 980101
end_head
```

If necessary, software for extracting information from the header, editing header information etc. can be obtained from the NIST ftp−server under the address:

ftp://jaguar.ncsl.nist.gov/pub/ .

The source package (for unix) is called "**sphere_2.6a.tar.Z**".

4. Turnmarker File Format

a. Single Language Dialogues

The marker file contains the onset and offset of a turn in **samples** and a string coding the speaker, the signal file and the turn number. These three fields are separated by a single **space**. The format is plain ASCII.

Field 1	***Field 2***	***Field 3***
onset in samples	offset in samples	STRING

Table 6

The format of STRING corresponds to the name of a partitur format file (see table 2: »All Names of a dialog«):

Language DialogNumber Scenario RecordSetup ChannelSpecification ChannelNumberTurnId

For each turn of the dialog there has to be one line in the file.

Example: g024ac.mar

```
46560 125760 g024acn1_000_ABA
110080 151200 g024acn2_001_ABC
154400 185920 g024acn1_002_ABA
193120 245760 g024acn2_003_ABC
...
...
979520 1002080 g024acn1_020_ABA
```

On− and offset are relative to the beginning of the signal in the file. To get the physical position of a turn within the file, 1024 bytes for the header have to be added.

b. Multilingual Dialogues

The marker file in multilingual dialogues is similar to the file described in the previous section, but STRING in *Field 3* has one more entry coding the Language which is spoken in the Turn.

Language DialogNumber Scenario RecordSetup ChannelSpecification ChannelNumberTurnId TurnLanguage

5. Speaker Protocol Format

For every recorded speaker, a speaker protocol file has to be filled out. The link between a speaker protocol and a recorded dialog is the speaker identification code and the native language of the speaker. Therefore, attention shall be payed on using unambiguous speaker codes in any language and on using absolutely the same code for this speaker in the transliteration.
At least the rows 1−3 in the following table 7 have to be filled out, the other rows are optional. Lines are skiped if there is no entry in the left field.
Tag and value are separated **by a tab**!
The filename is coded by the native language of the speaker, it’s speakerId and the extension ».spr«.

LanguageSpeakerId

	row	*Tag*	*Value*
▨	1	id	[AAA−ZZZ] use you own unambiguous speaker code, upper case
▨	2	sex	[m|f]
▨	3	date_of_birth	(YMMDD), e.g. 15th Febr. 1972 = 720215
	4	own_native_language	
	5	native_language_father	
	6	native_language_mother	
	7	primary_school	county/city of primary school years
	8	dialect	region in which the speaker lived most of the time; of which the accent/dialect is characteristic
	9	education	highest educational degree
	10	profession	
	11	height	Number[cm|foot|..] (no space between number and measuring unit, e.g. 169cm)
	12	weight	Number[kg|stone|pound...] (no space between number and measuring unit, e.g. 60kg)
	13	smoker	[y|n|former]
	14	right_left_handed	[r|l|ambi]
	15	comments	when present: at the end of the document, line feeds are allowed

Table 7

Example: g_fgr.spr

```
id              FGR
sex             f
date_of_birth   541215
own_native_language       g
native_language_father    g
native_language_mother  g
primary_school            K"oln
dialect         Rhein
education       Universit"at
profession      Lehrer
height          168cm
weight          58kg
smoker          n
right_left_handed         r
```

6. Recording Protocol Format

For every recording session, a recording protocol has to be filled out. This protocol relates to the recorded dialog, therefore, if there are parallel recordings by means of different recording setups only one recording protocol per dialog has to be filled out.

Row 1, 5 and 21 are optional.
Row 12 to 14 are only in use in case of multilingual dialogs.

Tag and value are separated **by one tab**!

	row	*Tag*	*Value*
	1	session_no	[0−9]+, digits (can be filled out with 0, optional)
▨	2	dialogue_name	[gejm][0−9][0−9][0−9][abcd] (dialog directory name)
▨	3	recording_date	(YYMMDD) e.g.: 971512
▨	4	scenario_date	(YYMMDD) e.g.: 971512 logical_date (acted date)
	5	recording_by	name of person who carried out the recording (optional)
▨	6	recording_site	[LMU|CMU|ATR|UHH|UBN]
▨	7	scenario_id	[a|b|c|d]
▨	8	no_speakers	[2−9]

	row	***Tag***	***Value***
	9	speaker1_id	[AAA−ZZZ]
	10	speaker2_id	[AAA−ZZZ]
	11	speaker3_id	[AAA−ZZZ]
	12	speaker1_language	[language][language_command],[language][language_command] [language] := g(erman), e(nglish), j(apanese); [language_command] := 0 (native), 1−3 (non−native, command level); native language on last position; example: g1,e0 English native speaker, speakes very good German; speakes German and English in this dialogue.
	13	speaker2_language	...
	14	speaker3_language	...
	15	speaker1_recmed_spec	All the recording setups used for speaker1 in this recording session: (close) (room) (tele) (at least one setup!) close: h(headset)|n(eckband mic)|c(lip mic), room: r(oom), tele: m(obile telephone)|p(hone, analog)|w(ireless)|d(ect) example: hrp = speaker1 has been recorded by means of headset, room−micro and analog phone
	16	speaker2_recmed_spec	...
	17	speaker3_recmed_spec	...
	18	speaker1_micbrand	comma separated list of types of used microphones, use underline within names, e.g. beyer_dynamics_115
	19	speaker2_micbrand	...
	20	speaker3_micbrand	...
	21	comments	when present: at the end, line feeds allowed

Table 8

Example: g012a.rpr

```
session_no        0003
dialogue_name     g012a
recording_date    970707
scenario_date     970601
recording_by      DO
recording_site    LMU
scenario_id       a
no_speakers       2
speaker1_id       ABA
speaker2_id       ABD
speaker1_recmed_spec    rnp
speaker2_recmed_spec    rnw
speaker1_micbrand       beyer_dynamic_mce_10, beyer_dynamic_nem_191
speaker2_micbrand       beyer_dynamic_mce_10, beyer_dynamic_nem_191
```