NOAH

Storing Audiological Measurements

Audiogram Standard

Version 1.1

HIMSA A/S

Preface

This is the first of a series of documents to be prepared by HIMSA A/S. Its purpose is to present and specify standard data formats for the storage and exchange of audiogram data within the framework of NOAH-compatible measurement and fitting software. The next documents in the series will specify standards for Real Ear, Hearing Instrument Test and Loudness Scaling.

Hearing Instrument Manufacturers' Software Association A/S (HIMSA A/S) was founded at the beginning of 1993 by a group of hearing instrument manufacturers. It has been HIMSA A/S's mission to develop and market the NOAH software, and to make it a de facto standard for integrated hearing care software within the entire hearing industry. Today, more than 50 companies support the NOAH standard, representing 85% of the hearing care industry. Further, NOAH has been sold to more than 5,000 dispensers and clinics thus constituting a market share of 25%.

The NOAH Fitting Framework is a software application that enables fitting and measurement software to share data on a common platform (NOAH). The fitting and measurement applications are provided by manufacturers who have signed a Know-How licence agreement with HIMSA and thereby obtained the right to distribute the NOAH software, and to develop NOAH-compatible software applications, also referred to as modules.

Data format standards are a natural prerequisite for the ability to share data. Therefore, in cooperation with its licencees, HIMSA has prepared data format standards for Audiogram, REM/HIT, Loudness Scaling, Impedance, Otoacoustic Emission and Evoked Response Audiometry measurement types.

The documentation for these standards is available in so-called header files. These files are part of the 'software development kit' which HIMSA automatically distributes to its licencees.

Unfortunately, it is our experience that the header files are too easily misinterpreted. It has thus been decided that HIMSA must prepare a comprehensive standard document for each of the aforementioned measurement types. These documents will provide a detailed presentation of the data structure of the measurement formats as well as describe the application of the various types of, e.g. 'specific audiograms'.

The various data standards are subject to revision twice a year by a committee consisting of manufacturers of audiological measurement equipment (AEMs). Based on input prepared by HIMSA, it will be the responsibility of this committee to approve both new standard documents and updates of existing standards.

The AEM Committee will meet on the Saturday following the end of the UHA Convention in Germany, i.e. in October, and on the Saturday following the end of the AAA Convention in the US, i.e. in April.

HIMSA also invites non-licencees to take part in the process of preparing and maintaining measurement data standards.

Figure 1 presents the principles by which NOAH administrates the measurement formats. Each block of stored data must be equipped with a header. This header uniquely identifies, e.g. the manufacturer who created the measurement, the type of measurement data contained in the data block and the measurement data format's revision number.

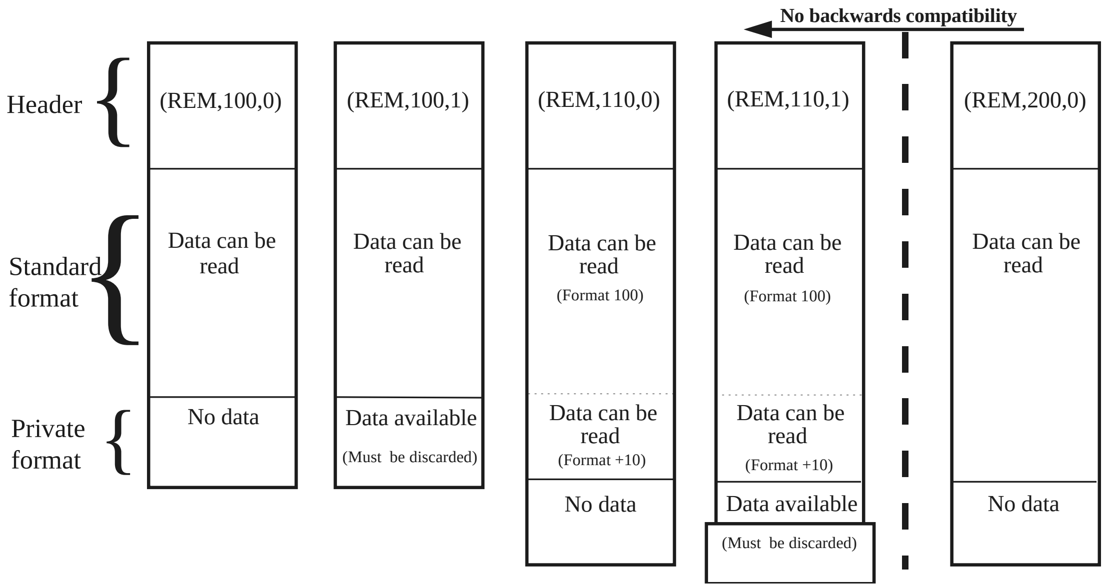


Figure 1: The handling of measurement data by NOAH

The basic revision number for a data format is 100. A data format with the revision number 110 is a direct extension of the basic 100 format. It is therefore possible for a revision 100 module to still read and understand a data block generated by a revision 110 module as it will simply discard the '+10' extension. A data format with the version number 200 would constitute a totally new revision thus making it impossible for revision 1xx modules to read revision 2xx data formats.

It is possible for a manufacturer to add non-standardised measurements to the public data block.

Document History

ver.	0.1	96-08-31	Document Template
ver.	0.5	96-09-30	First draft version
ver.	0.6	96-10-07	Second Draft version. Explanations for 13 audiograms.
ver.	0.65	96-10-09	Text changes. EndCurve explained.
ver.	0.66	96-10-09	Minor changes. Page numbers still missing in index.
ver.	0.70	96-11-13	Text added about reading and writing Measuring Conditions.
ver.	0.71	96-11-18	The Initial Condition (as defined in initialCond) for PresentType1,2 changed to NoPresentType (1). The paragraphs 2.2.3 and 2.2.4 updated accordingly.
ver.	0.80	96-11-28	Text added about the use of the value undefInt (-32767) “nothing stored” and the values Unknown (0) and Not Used (channel, parameter…) (1).
ver.	1.0	96-11-30	Index entries added. English language corrected. Minor text changes.
ver.	1.1	09-05-01	Note on adding extra byte for alignment to Section 3.1.2

1.1 Contents

1.2 References

[Audidat] NOAH v. 1.3 Audiometry data ver.0.4. 1995. Edited by Pallas Informatik A/S.

[Framework] NOAH Framework ver. 0.85. System Architecture Specification. Pallas Informatik A/S.

[HOCA-4] Handbook of Clinical Audiology, edited by Jack Katz.
Williams & Wilkins, 1994, 4. Edition.

2. Introduction

2.1 A few words about programming with AUDDEF.H

The intention of this document is to explain the use of the NOAH version 2.0 standard for storing audiograms (AUDDEF.H). The structure used for storing audiograms is documented in the header file written for the programming language "C" and defines the structure TAudioSession. "T" stands for type definition, and the structure AudioSession contains a total of 76 audiograms and covers 13 different kind of audiograms.

The aim of this document is to explain the correct use of the structure TAudiosession. This is done by reading the header file AUDDEF.H "upside down" , starting with the "outer" definition of TAudioSession, the 76 audiograms with a Measuring Condition structure (MeasCond) attached to each and every audiogram. Then comes the type definition of the 13 different audiograms, their Measuring Conditions and their associated curve points, ending with the definition of all "inner" types, all defined as integers with some associated constants ("#define" NamedValues).

3. The NOAH standard for audiograms

3.1 Data Structure

In order to describe the data structure as it is defined in AUDDEF.H, an extended version of the Abstract Syntax Notation No. 1 (ASN.1) language is used [1]. This is done for the following reasons:

1. Explanation of the data structure in AUDDEF.H starting with "the 13 audiograms" and the structure that keeps them together: TAudioSession ('T' in this context means type definition). From this "outer", all-embracing type, all constituent types are defined as we meet them. (In effect the header file 'upside down'). The definition in ASN.1 ends in the case of this header file by defining all the fundamental types as integers.

2. ASN.1 contains a few useful distinctions which are used in this chapter to explain important places in AUDDEF.H; where the order of variables matters, and where it does not. Note that variables are called 'components' when in an outer structure:

SEQUENCE	Ordered collection of component types.
SEQUENCE OF	Ordered collection of variables of the component type.
SET	Unordered collection of component types, all distinct.
SET OF	Unordered collection of variables of the component type

[1] ASN.1 is defined by ISO and the International Telecommunication Union (ITU) (See ISO 8824) with a set of so-called Basic Encoding Rules, which we shall NOT use here. Instead a "Direct Encoding Rule" can be formulated: Data are encoded exactly as they are shown, down to the definition of the INTEGER as consisting of two byte, low-order transmitted first (placed at lower address).

3.1.1 The Integer type used in AUDDEF.H

minInt	-32768 #8000 hex	Lowest negative value represented in two bytes using standard "2's complement" representation. According to [Framework] , this value is **illegal** for the integer types defined in AUDDEF.H.
undefInt	-32767 #8001 hex	Used to indicate that the value is **undefined,** a value which is assigned to the constant undefInt. Ref. [Framework]
minParmInt	-32766 #8002 hex	Lowest negative **legal** value in parameters defined as integer types in AUDDEF.H according to [Framework].
Unknown	0 #0000 hex	*In Parameters:* The parameter is **defined,** though to an **unknown** value**.** *In curve points:* Use logic here! For the types TdB10, TPct100, TTime100, TTime1000, Twords and TSiSiIncrements the value 0 is of course **defined** and **valid.** However, for the THertz and THertz10 types the value means **undefined**.
NoParam	1 #0001 hex	*In Parameters:* The parameter is **defined** Not Used (channel, parameter…), see AUDDEF.H for different explanations of the different types: noSignal, noAUXParm and so on. *In curve points:* Note the type TPointStatus where noStatus (1) is used for **valid codepoints.**
MaxInt	32767 #7FFF hex	Highest positive value. Ref. [Framework].

3.1.2 Definition of Audiogram standard

NOTE: When adding a rule name, a single byte needs to be used for alignment. For example, if a field is defined to have 51 characters, where each character is 1 byte, then an extra byte needs to be added for alignment purposes. This is an empty byte, set aside to serve as a placeholder.

```
Audiogram  DEFINITIONS ::=

BEGIN
-- Basic structure of 13 different audiograms. Comments until end of line use two hyphens !
-- The whole structure has to be saved even though only 1 of the 76 audiograms is actually used.
-- The NOAH database will compress data after reception, DO NOT attempt to do this !

TAudioSession ::= SEQUENCE
{
    toneTHRAudiograms      SET OF  6 TToneTHRAudiogram    --  6 x  308 = 1848
    toneMCLAudiograms      SET OF  6 TToneMCLAudiogram    --  6 x  308 = 1848
    toneUCLAudiograms      SET OF  6 TToneUCLAudiogram    --  6 x  308 = 1848
    ablbAudiograms         SET OF  1 TABLBAudiogram       --  1 x 1988 = 1988
    stengerAudiograms      SET OF  1 TStengerAudiogram    --  1 x  308 =  308
    dliAudiograms          SET OF  2 TDLIAudiogram        --  2 x  356 =  712
```

```
    dlfAudiograms          SET OF  2 TDLFAudiogram          --  2 x  356 =  712
    sisiAudiograms         SET OF  2 TSiSiAudiogram         --  2 x  404 =  808
    decayAudiograms        SET OF  2 TDecayAudiogram        --  2 x  668 = 1336
    speechDLAudiograms     SET OF 12 TSpeechDLAudiogram     -- 12 x  260 = 3120
    speechSRTAudiograms    SET OF 12 TSpeechSRTAudiogram    -- 12 x  260 = 3120
    speechMCLAudiograms    SET OF 12 TSpeechMCLAudiogram    -- 12 x   76 =  912
    speechUCLAudiograms    SET OF 12 TSpeechUCLAudiogram    -- 12 x   76 =  912
}                                                                              --
Total 19472 bytes
```

3.1.3 Standard Threshold (THR) Audiogram

Hearing Threshold (THR) Audiogram [68 + 24 x 10 = 308 bytes]. Note that the max. 24 curve points can come in any order as [freq, intensity] pairs. This is the basic audiogram for recording the patient's Hearing Threshold Level (HTL).

Method: Presentation of Pure tone stimulus through headphones. Masking in the opposite ear is frequently used.

Ref. [HOCA-4, Chapter 7: Puretone Air-Conduction Threshold Testing]

```
TToneTHRAudiogram ::= SEQUENCE
{
    measCond               TMeasCond                        -- Measuring Conditions
    curve                  SET OF 24 TTonePoint             -- 24 Threshold points
}
```

3.1.4 Most Comfortable Loudness (MCL) Audiogram

Most Comfortable Loudness (MCL) Audiogram [68 + 24 x 10 = 308 bytes]. Note that the max. 24 curve points can come in any order as freq, intensity pairs. Each point is properly identified by its standard frequency for which the intensity is recorded. The patient indicates that the tones presented are heard at "Most Comfortable Loudness level". Ref. [HOCA-4, Chapter 13: Integrating Audiometric Results]

```
TToneMCLAudiogram ::= SEQUENCE
{
    measCond               TMeasCond                        -- Conditions
    curve                  SET OF 24 TTonePoint             -- 24 MCL points
}
```

3.1.5 Uncomfortable Loudness (UCL) Audiogram

Complete UCL Audiogram: [68 + 24 x 10 = 308 bytes]. Note that the max. 24 curve points can come in any order as [freq, intensity] pairs. Each point is properly identified by its standard frequency for which the intensity is recorded. The patient indicates that the tones presented are heard too loudly at an "Uncomfortable Loudness level". Ref: [HOCA-4, Chapter 13: Integrating audiometric results]

```
TToneUCLAudiogram ::= SEQUENCE
{
    measCond        TMeasCond               -- Conditions
    curve           SET OF 24 TTonePoint    -- 24 UCL points
}
```

3.1.6 Alternate Binaural Loudness Balance (ABLB) Audiogram

The ABLBAudiogram [68 + 192 x 10 = 1988 bytes] is used when a patient has one ear that is hearing impaired. Tones are presented alternately to both ears, and the patient is asked to make equal loudness judgements for each. Recruitment is tested as follows: At the frequencies [500,1000,2000,4000] Hz the loudness balance test is performed 10 dB above the obtained threshold for the frequency being used.

The tones are still alternated, and the patient indicates, when the loudness is equal in both ears. In this way, equal loudness curves for [0,10,30,50,70,90] dB can be obtained. The audiogram can store up to 192 points of equal loudness.

Ref. [HOCA-4, Chapter 11: Tests of cochlear function].

```
TABLBAudiogram  ::= SEQUENCE
{
    measCond            TMeasCond                   -- Conditions
    curve               SET OF 192 TTonePoint       -- 192 ABLB points
}
```

3.1.7 Stenger Audiogram

Stenger Audiogram [68 + 24 x 10 = 308 bytes]. The Stenger test is used when a patient claims that his hearing is impaired in one ear. It involves presentation of a tone of one frequency to both ears. The audiometer should allow separate intensity control for each channel.

The tone is first introduced to the good ear at a level which is 5-10 dB above the known threshold. The tone is then gradually introduced into the “bad” ear until it is at a point some 10 dB louder than the level of the tone in the good ear. If the patient is simulating a hearing loss he would hear the tone louder in the bad ear, but still claim that he hears it with the good ear. The level of the tone in the good ear is then dropped below its threshold point and entirely removed from the good ear.

If the patient has a loss as measured, he will report hearing the tone in the bad ear when it exceeds the actual hearing loss by 5 - 10 dB. Once the intensity of the tone in the bad ear is sufficient to reach the actual threshold, an increase in intensity of 10-15 dB will cause the tone to be heard in the bad ear.

Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TStengerAudiogram ::= SEQUENCE
{
    measCond            TMeasCond               -- Conditions
    curve               SET OF 24 TTonePoint    -- 24 (no. response in good ear)
}
```

3.1.8 Difference Limen Intensity (DLI) Audiogram

Difference Limen Intensity Audiogram [68 + 24 x 12 = 356 bytes], also called Amplitude Modulation (AM) Test or the Lüscher-Zwislocki test. In this audiogram, an Amplitude Modulation is added to a steady tone, and the patient is asked to indicate the smallest variation he can detect.

The result is then recorded in dB. The most significant level of recording is found to be approx. 40 dB above the hearing threshold.

Ref. [HOCA-4, Chapter 5: Psychoacoustic considerations in clinical audiology] and also
Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TDLIAudiogram ::= SEQUENCE
{
    measCond        TMeasCond               -- Conditions
    curve           SET OF 24 TDLIPoint     -- 24 DLI points
}
```

3.1.9 Difference Limen Frequency (DLF) Audiogram

Difference Limen Frequency Audiogram [68 + 24 x 12 = 356 bytes], also called Frequency Modulation (FM) Test. In this audiogram, a Frequency Modulation is added to a steady tone, and the patient is asked to indicate the smallest variation he can detect.

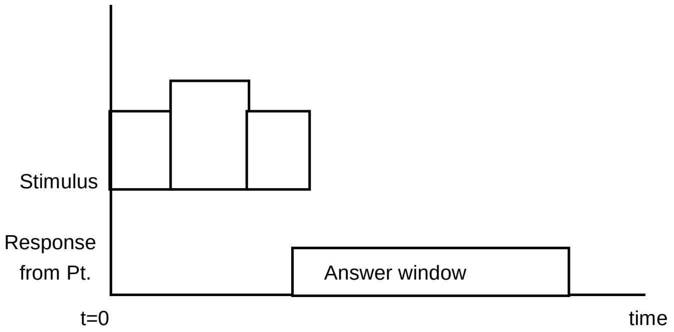
The result is then recorded in dB. The most significant level of recording is found to be approx. 40 dB above the hearing threshold.

Ref. [HOCA-4, Chapter 5: Psychoacoustic considerations in clinical audiology] and also
Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TDLFAudiogram ::= SEQUENCE
{
    measCond        TMeasCond               -- Conditions
    curve           SET OF 24 TDLFPoint     -- 24 DLF points
}
```

3.1.10 Short Increment Sensitivity Index (SISI) Audiogram

Short Increment Sensitivity Index (SISI) [68 + 24 x 14 = 404 bytes]. The SISI test is used to examine recruitment in cases where a severe hearing loss has been detected. The test is a further development of the Difference Limen Intensity Test, and it is based on an automated series of stimuli, where the patient is asked to give a signal when a change in intensity is detected:



Example of SiSi-test: Amplitude Modulation (AM) test with a predefined sequence of stimuli.

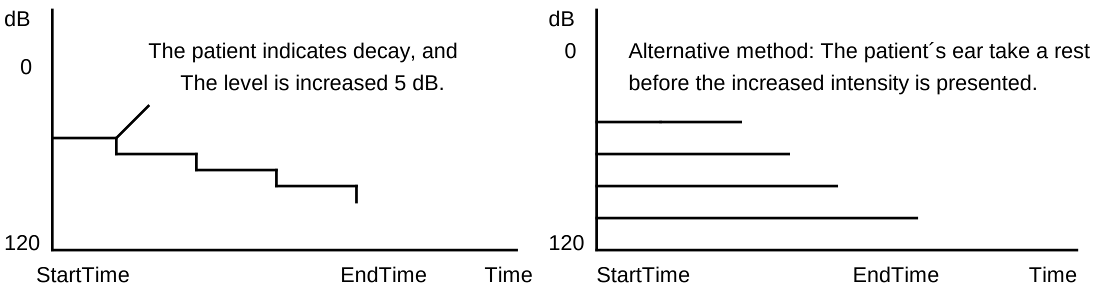
The patient will indicate within the specified time interval whether he can hear the amplitude modulation.

Ref. [HOCA-4, Chapter 11: Tests of cochlear function] and also
Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TSiSiAudiogram ::= SEQUENCE
{
    measCond        TMeasCond                   -- Conditions
    curve           SET OF 24   TSiSiPoint      -- 24 SiSi points
}
```

3.1.11 Decay Audiogram

Decay Audiogram [68 + 50 x 12 = 668 bytes]. A Decay Audiogram shows the ability of a patient (PT) to hear a sustained tone. PT indicates with a reply button when the tone is decayed (not heard) and the tone intensity is raised, e.g. by 5 dB, and the test is repeated:



Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TDecayAudiogram ::= SEQUENCE
{
    measCond        TMeasCond                  -- Conditions
    curve           SET OF 50 TDecayPoint      -- 50 decay sections
}
```

3.1.12 Speech Discrimination Loss (DL) Audiogram

Speech Discrimination Loss Audiogram [68 + 24 x 8 = 260 bytes]. Speech discrimination is defined as the ability to discriminate phonemic lists of higher complexity than those used for determining the Speech Reception Threshold.(SRT) (understanding 50 % of a series of easily understandable words).

The speech Discrimination Loss is the percentage of phonemess not repeated correctly. Even if the intensity is increased to, e.g. 40 dB above the SRT and the DL is measured at this level, the hearing impaired will often have a considerable Discrimination Loss (DL). The Discrimination Loss (DL) Curve shows the score when reproducing words (monosyllables or disyllabless) presented at different levels:

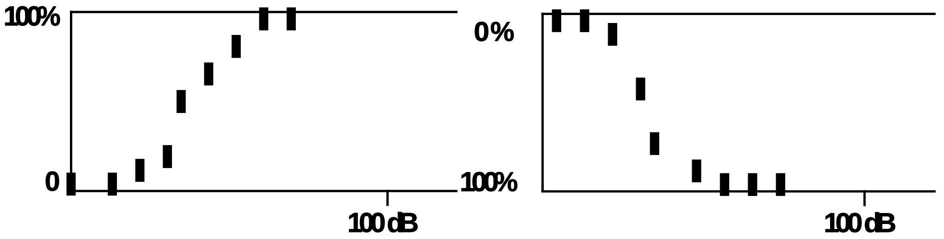


Fig: Discrimination Loss (DL) Fig: Discrimination Score.

The Discrimination Score shows the same graph as the DL but with the y-axis pointing downwards.

Ref. [HOCA-4, Chapter 10: Speech threshold and recognition / Discrimination testing].

```
TSpeechDLAudiogram ::= SEQUENCE
{
    measCond        TMeasCond                  -- Conditions
    curve           SET OF 24 TSpeechPoint     -- 24 DL points
}
```

3.1.13 Speech Reception Threshold (SRT) Audiogram

Speech Reception Threshold (SRT) Audiogram [68 + 24 x 8 = 260 bytes]. The SRT is also defined as the Threshold of Intelligibility (TI). It is defined as the loudness increase (in dB HL) necessary for the patient to reach a score of 50 % when presented for a series of easily understandable phonemess. The SRT can be compared with the tone audiometry thresholds at the frequencies [500, 100, 2000] Hz.

The SRT measurement is carried out as a control of the tone THR audiogram. Monosyllables are used. The intensity is gradually increased starting at the patient's threshold for hearing tones - the best of the above mentioned values are used.

Ref. [HOCA-4, Chapter 10: Speech threshold and recognition / Discrimination testing].
Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TSpeechSRTAudiogram ::= SEQUENCE
{
    measCond        TMeasCond                  -- Conditions
```

```
    curve              SET OF 24 TSpeechPoint        -- 24 SRT points
}
```

3.1.14 Speech Most Comfortable Loudness (MCL) Audiogram

Speech Most Comfortable Loudness (MCL) Audiogram [68 + 1 x 8 = 76 bytes]. Measurement of Discrimination Loss is done at Most Comfortable Loudness for the patient, normally 30-40 dB above the SRT (or TI).

Some patients might find this level too high (recruitment present). This may make it difficult to find the correct level for recording the MCL speech audiogram. In such cases a complete S-curve should be recorded.

Ref. [HOCA-4, Chapter 10: Speech threshold and recognition / Discrimination testing] and also
Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TSpeechMCLAudiogram ::= SEQUENCE
{
    measCond           TMeasCond                     -- Conditions
    curve              SET OF 1 TSpeechPoint         -- 1 MCL point
}
```

3.1.15 Speech Uncomfortable Loudness of Hearing (UCL) Audiogram

Speech UnComfortable Loudness (UCL) Audiogram [68 + 1 x 8 = 76 bytes]. The SRT, MCL and UCL for speech audiograms are saved as a single intensity value.

A complete S-curve would show the whole range from Speech Reception Threshold (SRT), to Most Comfortable Loudness (MCL) to Uncomfortable Loudness.(UCL)

Ref. [HOCA-4, Chapter 10: Speech threshold and recognition / Discrimination testing].
Ref. [HOCA-4, Chapter 13: Integrating audiometric results].

```
TSpeechUCLAudiogram ::= SEQUENCE
{
    measCond           TMeasCond                     -- Conditions
    curve              SET OF 1 TSpeechPoint         -- 1 UCL point
}

END -- of Audiogram definitions
```

3.1.16 Curve points

```
CurvePoints DEFINITIONS ::=


BEGIN

-- Data point used to store thresholds of tone audiograms:
-- [5 x 2 = 10 bytes]

TTonePoint ::= SEQUENCE
{
    freq1           THertz            -- Hz for stim channel
    intensity1      TdB10             -- dB for stim channel
    freq2           THertz            -- Hz for mask channel
    intensity2      TdB10             -- dB for mask channel
    status          TPointStatus      -- Status
}



-- Data point for DLI thresholds; see Difference Limen Intensity (DLI) Audiogram
-- Note that status will normally be equal to noStatus for saved codepoints.
-- The difference limen (DL) will be saved for the frequency in question in the variable modSize:
-- [6 x 2 = 12 bytes]

TDLIPoint ::= SEQUENCE
{
    freq1           THertz          -- Hz for stim channel
    intensity1      TdB10           -- dB for stim channel
    freq2           THertz          -- Hz for mask channel
    intensity2      TdB10           -- dB for mask channel
    modSize         TdB10           -- Difference Limen Threshold dB
    status          TPointStatus    -- Status
}



-- Data point for DLF thresholds; see Difference Limen Frequency (DLF) Audiogram
-- [6 x 2 = 12 bytes]

TDLFPoint ::= SEQUENCE
{
    freq1           THertz          -- Hz for stim channel
    intensity1      TdB10           -- dB for stim channel
    freq2           THertz          -- Hz for mask channel
    intensity2      TdB10           -- dB for mask channel
    modSize         TPct100         -- Difference Limen Threshold
    status          TPointStatus    -- Status
}
```

```
-- Data point for Short Increment Sensitivity Index (SISI) Audiogram
-- [7 x 2 = 14 bytes]


TSiSiPoint ::= SEQUENCE
{
    freq1              THertz              -- Hz for stim channel
    intensity1         TdB10               -- dB for stim channel
    freq2              THertz              -- Hz for mask channel
    intensity2         TdB10               -- dB for mask channel
    modSize            TdB10               -- SiSi Increment Size in dB
    nofHits            TSiSiIncrements     -- Number of Answers
    nofIncrements      TSiSiIncrements     -- Total No. of SiSi Increments [20]
}



-- Data point for Decay sections, see the paragraph defining  Decay Audiogram above
-- [6 x 2 = 12 bytes]

TDecayPoint ::= SEQUENCE
{
    freq1              THertz              -- Hz for stim channel
    intensity1         TdB10               -- dB for stim channel
    freq2              THertz              -- Hz for mask channel
    intensity2         TdB10               -- dB for mask channel
    startTimeSec       TTime100            -- Start Time
    endTimeSec         TTime100            -- End Time for this section
}



-- Data point for Speech Scores, used by all speech audiograms
-- [4 x 2 =  8 bytes]

TSpeechPoint ::= SEQUENCE
{
    intensity1         TdB10               -- dB for stim channel
    intensity2         TdB10               -- dB for mask channel
    scorePct           TPct100             -- Speech score Hit Rate in %.
    words              TWords              -- Score based upon no of words
}

END – of CurvePoints
```

3.1.17 Reading and writing curve points

The reading of curve points in a tone audiogram from NOAH ver 2.0 is per definition done in the following way:

The freq1 is read first. The curve points might be ordered but since they are defined as a set, they might also be *unordered* with respect to frequency. Read the curve points while checking freq1.

Curve points are read until the namedValue endCurve appears:

```
-- Do not overlook this end of curve marker !!!

endCurve TTonePoint ::=
{
    undefInt,                -- freq1 = undefInt defines the endCurve
    undefInt,                -- or any other value
    undefInt,                -- or any other value
    undefInt,                -- or any other value
    undefInt                 -- or any other value
}
```

After endCurve, curve points with freq1 = 0 or freq1 = undefInt (-32 767) are discarded. Curve points with such unreasonable frequency should be discarded any time during the reading.

When writing curve points, place them sorted with freq1 in ascending order ending with endCurve and fill the rest of the array with endCurve markers (undefInts). The latter is not mandatory but will be appreciated by fellow programmers.

3.1.18 MeasCond

-- Information about each recorded curve: [17 x 2 x 2 = 68 bytes]

```
MeasuringConditions DEFINITIONS ::=

BEGIN

TMeasCond ::= SEQUENCE
{
    signalType1        TSignal,            -- Essential for determining tone or speech
    signalType2        TSignal,            -- Used if channel 2 is active

    warbleModFreq1     THertz,
    warbleModFreq2     THertz,
    warbleModSize1     TPct100,
    warbleModSize2     TPct100,
    auxParm1           TAUXParm,           -- Normally used in speech audiograms
    auxParm2           TAUXParm,           -- If an AUX channel 2 is in use
                                           --              in a speech audiogram
    signalOutput1      TSignalOutput,      -- Essential to determine Left or Right ear
    signalOutput2      TSignalOutput,      -- Used if channel 2 is active
    presentType1       TPresentType,
    presentType2       TPresentType,

    pulseModFreq1      THertz10,
    pulseModFreq2      THertz10,
    pulseDutyCycle1    TPct100,
    pulseDutyCycle2    TPct100,

    amModSize1         TdB10,
    amModSize2         TdB10,
    fmModSize1         TPct100,
    fmModSize2         TPct100,
    onTime1            TTime1000,
    onTime2            TTime1000,
    offTime1           TTime1000,
    offTime2           TTime1000,

    siSiParm1          TdB10,
    siSiParm2          TdB10,

    transType1         TTransType,
    transType2         TTransType,
    transCalStand1     TTransCalStand,
    transCalStand2     TTransCalStand,

    dBWeighting1       TdBWeighting,              -- Normally HTL
    dBWeighting2       TdBWeighting,              -- Normally HTL if used
    condition1         TCondition,                -- Indicates whether a hearing aid is used
    condition2         TCondition                 -- do., for channel 2
}
```

END – of MeasuringConditions

3.1.19 *Defined values*

```
DefinedValues DEFINITIONS ::=

BEGIN

-- The value 0: Unknown. This goes for all '#defines'
-- Values >= LastX should be set to UnKnown to ensure that extensions to the
-- '#define' structure can be dealt with


TPointStatus ::= INTEGER          -- Status of threshold point
{
   undefInt              -32767,            -- Nothing stored  (NOAH definition)
   unknown               0,                 -- Unknown
   noStatus              1,                 -- No Special Status
   alwaysResponse        2,                 -- Patient might hear better than this
   noResponse            3,                 -- Patient cannot hear this point
   lastPointStatus       4                  -- End of define List
}


TSignal ::= INTEGER              -- Description of signal
{
   undefInt              -32767         -- Nothing stored  (NOAH definition)
   unknown               0,             -- Unknown
   noSignal              1,             -- Channel without any signal
   tone                  2,             -- Pure Tone
   warble                3,             -- Warble Tone
   nBN                   4,             -- Narrow Band Noise
   sN                    5,             -- Speech Noise
   wN                    6,             -- White Noise
   pN                    7,             -- Pink Noise
   aUX                   8,             -- Auxiliary Signal (CD / Tape)
   mIC                   9,             -- Live Voice from microphone
   lastSignal            10             -- End of define List
}

TAUXParm    ::= INTEGER    -- External signal description
{
   undefInt              -32767             -- Nothing stored  (NOAH definition)
   unknown               0,                 -- Unknown
   noAUXParm             1,                 -- No parameters
   monoSyllabicWords     2,                 --
   multiSyllabicWords    3,                 --
   dichoticWords         4,                 --
   freiburger            5,                 --
   reim                  6,                 --
   numerals              7,                 --
   lastAUXParm           8                  -- End of define List
}
```

```
TSignalOutput           ::= INTEGER -- Output transducer choice
{
   undefInt             -32767,        -- Nothing stored  (NOAH definition)
   unknown              0,             -- Unknown
   noSignalOutput       1,             -- No Output connected
   acL                  2,             -- Air Conductor    Left
   acR                  3,             --                  Right
   acBin                4,             --                  Binaural
   bcL                  5,             -- Bone Cond.       Left
   bcR                  6,             --                  Right
   bcBin                7,             --                  "Binaural"
   ffL                  8,             -- Free Field Left
   ffR                  9,             --                  Right
   ffBin                10,            --                  Binaural
   ipL                  11,            -- Insert Phone     Left
   ipR                  12,            --                  Right
   ipBin                13,            --                  Binaural
   lastSignalOutput     14             -- End of define List
}


TPresentType ::= INTEGER               -- How the signal is presented
{
   undefInt             -32767,        -- Nothing stored  (NOAH definition)
   unknown              0,             -- Unknown
   noPresentType        1,             -- Invalid presentation
   continuos            2,             -- Normally steady and unmodulated
   pulse                3,             -- Pulsed presentation
   abLB                 4,             -- Alternating between ch.1 & ch.2
   am                   5,             -- Amplitude Modulated
   fm                   6,             -- Stepwise Frequency modulated
   impulse              7,             -- Random pulses
   siSi                 8,             -- SiSi signal (on 200 mS)
   lastPresentType      9              -- End of define List
}

TTransType      ::= INTEGER        -- Transducer type description
{
   undefInt             -32767,        -- Nothing stored  (NOAH definition)
   unknown              0,             -- Unknown
   noTransType          1,             -- No transducer connected
   tDH39                2,             --
   hDA200               3,             --
   eartone3A            4,             --
   dt48                 5,             --
   tdh49                6,             --
   b71                  7,             --
   b72                  8,             --
   beoton               9,             --
   holmberg             10,            --
   lastTransType        11             -- End of define List
}
```

```
TTransCalStand  ::= INTEGER              -- Transducer calibrated according to:
{
   undefInt                -32767,           -- Nothing stored  (NOAH definition)
   unknown                 0,                -- Unknown
   noTransCalStand         1,                -- No Standard available
   iso389                  2,                --  According to ISO standard
   iso389_FFEQ             3,                --  Free Field Equalized
   iso7566                 4,                -- According to ISO standard
   iso7566_FFEQ            5,                -- Free Field Equalized
   iso8798                 6,                -- According to ISO standard
   iso8798_FFEQ            7,                -- Free Field Equalized
   iso226                  8,                -- According to ISO standard
   iso226_FFEQ             9,                -- Free Field Equalized
   ansiS36                 10,               -- According to ISO standard
   ansiS36_FFEQ            11,               -- Free Field Equalized
   lastTransCalStand       12                -- End of define List
}


TdBWeighting ::= INTEGER                 -- Interpretation of dB values
{
   undefInt                -32767,           -- Nothing stored  (NOAH definition)
   unknown                 0,                -- Unknown
   nodBWeighting           1,                -- Weighting unavailable
   htl                     2,                -- values in HTL according to Standard
   spl                     3,                -- values in SPL (dB/µV)
   abs                     4,                -- values in ABS (ex. dB/µN)
   csl                     5,                -- values in CSL according to Standard
   lastdBWeighting         6                 -- End of define List
}


TCondition ::= INTEGER                   -- Condition for the test
{
   undefInt                -32767,           -- Nothing stored  (NOAH definition)
   noCondition             1,                -- Condition unavailable
   unAided                 2,                -- Measured without Hearing Aid
   aided                   3,                -- Measured with Hearing Aid
   lastCondition           4                 -- End of define List
}

-- All the previous definitions are based on 2-byte integers:

THertz              ::= INTEGER       -- Frequency in Hz (NOAH definition)
THertz10            ::= INTEGER       -- Frequency in Hz x 10
TdB10               ::= INTEGER       -- Decibel   x 10  (NOAH definition)
TPct100             ::= INTEGER       -- Percent   x 100  (Note: Missing "0" in AUDDEF.H)
TTime100            ::= INTEGER       -- Time in seconds x 100 [0-320] sec.
TTime1000           ::= INTEGER       -- Time in seconds x 1000 [0-32] sec.
TWords              ::= INTEGER       -- Number of words in speech tests
TSiSiIncrements     ::= INTEGER       -- Number of SiSi Increments
```

END – Of defined values

3.2 Reading and writing Audiogram Measurements

In the previous chapter, the TAudioSession structure was explained. This chapter will provide some hints for the actual reading and writing of a TAudioSession as the structure is defined in the NOAH standard version 2.0.

The TAudioSession structure was expanded but also simplified in NOAH version 2.0: The structure now contains 76 audiograms of 13 different kinds and the representation is kept simple and structured in a uniform way (as previously described).

Unfortunately, this means that only a small fraction of the 19,472 bytes of a TAudioSession is filled with usable data. The NOAH database caters for this by: compressing data before adding them to its database / expanding them before supplying the data to an external software module. The drawback is in other words is a slowed down communication, while the advantage is the uniform structure of audiogram data.

All audiogram data that were converted from NOAH version 1.2 contain lots of empty audiograms, and for the audiograms containing useful data, their Measuring Conditions use the Minimum Settings for tone or speech audiograms, described later in this chapter.

3.2.1 Reading the Audiogram Measurements

The NOAH ver. 2.0 specification attaches a comprehensive measurement condition structure called Measuring Conditions (Type definition TMeasCond) to each recorded curve.

In order to find the audiograms containing useful data when reading a TAudioSession, your program should read the Measuring Conditions attached to each Audiogram.

In this chapter a namedValue[2] called initialCond is introduced. Most of the measurement conditions will be equal to this namedValue: initialCond. Subsequent chapters describe the minimum changes in initialCond that make MeasCond valid for one of the two audiogram types, “Tone” or “Speech”.

Note 1: If the Measuring Conditions for an audiogram are completely identical to initialCond, this means that the associated audiogram is empty.

Note 2: The definitions for integer values written at the beginning of this chapter apply. However, the value zero can be found in empty audiograms where the correct value would be undefInt.

3.2.2 Writing the Audiogram Measurements

When writing an Audiogram Measurement, use the following method:

1) Initialise the 76 audiograms in the structure by setting all Measuring Conditions to the initial conditions: initialCond (see below). The codepoints should be initialised with endCurve. Refer to the paragraph 3.1.17, Reading and writing curve points.

2) Insert the appropriate values in the actual Measuring Conditions for the audiogram(s) that you want to save. Start with the minimum settings shown in the two subsequent chapters and modify according to the measuring conditions that were actually applied when recording the audiogram.

The curve points are then inserted in accordance with the directions mentioned in paragraph 3.1.17, Reading and writing curve points.

[2] ASN.1 defines namedValues as structures of an indicated type with a defined content.

Initial Measurement Conditions, the namedValue initialCond

Data Type	*Fields*	*Value*
TSignal	SignalType1, SignalType2	NoSignal (1)
THertz10	WarbleModFreq1, WarbleModFreq2	undefInt (-32767)
TPct100	WarbleModsize1, WarbleModSize2	undefInt (-32767)
TAUXParm	AUXParm1, AUXParm2	NoAUXParam (1)
TSignal Output	SignalOutput1, SignalOutput2	NoSignalOutput (1)
TPresentType	PresentType1, PresentType2	NoPresentType (1)
THertz10	PulseModFreq1, PulseModFreq2	undefInt (-32767)
TPct100	PulseDutyCycle1, PulseDutyCycle2	undefInt (-32767)
TdB10	AMModSize1, AMModSize2	undefInt (-32767)
TPct100	FMModSize1, FMModSize2	undefInt (-32767)
TTime1000	OnTime1, OnTime2	undefInt (-32767)
TTime1000	OffTime1, OffTime2	undefInt (-32767)
TdB10	SisiParm1, SisiParm2	undefInt (-32767)
TTransType	TransType1, TransType2	NoTransType (1)
TTransCalStand	TransCalStand1, TransCalStand2	NoTransCalStand (1)
TdBWeighting	dBWeighting1, dBWeighting2	NoDBWeighting (1)
TCondition	Condition1, Condition2	NoCondition (1)

3.2.3 Minimum settings for tone audiograms

Tone Audiograms		
Stimulus channel ; Channel 1 per definition :		
Field in MeasCond	*Value*	*Explanation*
SignalType1	Tone (2)	E.g. pure tone or warble tone
SignalOutput1	ACL (2) / ACR (3) BCL (5) / BCR (6)	Air Conduction left or right, depending on side. Bone Conduction, likewise.
PresentType1	Continuous (2)	The sound is presented in a time window, e.g. 200 - 1000 ms.
dBWeighting1	HTL (2)	Hearing Threshold Level weighting.
Masking channel / Additional Stimulus channel for Binaural Measurement ; Channel 2:		
Field in MeasCond	*Value*	*Explanation*
SignalType2	<masking signal>	Could be noSignal, a tone or a noise type. Legal values are [1..9]; see the data type TSignal.
SignalOutput2	ACR (3) / ACL (2)	Air Conduction opposite that chosen for channel 1.
PresentType2	Continuous (2)	The sound is presented in a time window, e.g. 100 - 1000 ms.
DBWeighting2	HTL (2)	Hearing Threshold Level weighting.

3.2.4 Minimum settings for speech audiograms

Speech Audiograms		
Stimulus channel ; Channel 1 per definition :		
Field in MeasCond	*Value*	*Explanation*
SignalType1	AUX (8) / MIC (9)	Stimulus from Compact Disc or Audio Tape.
AUXParam1	< list of phonemess>	E.g. monosyllabic words, numerals, etc. Legal values are [1..7] , see the data typeTAUXParm.
SignalOutput1	ACL (3) / ACR (2)	Air Conduction left or right, depending on side
dBWeighting1	HTL (2)	Hearing Threshold Level weighting.
Masking channel / Additional stimulus channel for Binaural Measurement ; Channel 2:		
Field in MeasCond	*Value*	*Explanation*
SignalType2	SN (5)	Speech Noise.
SignalOutput2	ACR (3) / ACL (2)	Air Conduction opposite that chosen for channel 1.
DBWeighting2	HTL (2)	Hearing Threshold Level weighting.

Appendix A: Vocabulary and Abbreviations

A

ABLBAudiogram	Alternate Binaural Loudness Balance (ABLB) test, also known as the Fowler Test. The patient is required to make equal-loudness judgements for tones alternately presented to both ears. The test is used when one of the patient’s ears is essentially normal while the other is hearing impaired.
ABLB · See TPresentType	the presentation method is “Alternating between channel 1 and channel 2”.
ABLB points · ABLB Curve ·	The result of an ABLB test is stored in up to 192 ABLB Points with equal or different frequencies. The points are stored in a structure called the ABLB curve.
ABS · See TdBWeighting	Absolute Value (ABS) of a measurement without any corrections.
ACBin · See TSignalOutput	Air Conduction Binaural (ACBin): The sound is presented in a headset with a tone stimulus in both ears.
ACL · See TSignalOutput	Air Conduction Left (ACL): The sound is presented in a headset with a tone stimulus for the Left ear. The Right ear may receive a masking noise.
ACR · See TSignalOutput	Air Conduction Right ACR) : The sound is presented in a headset with a tone stimulus for the Right ear. The Left ear may receive a masking noise.
Aided · See TCondition	Measured while the patient uses a hearing instrument. Most often a free field measurement; the stimulus is presented via a loudspeaker.
AlwaysResponse · see TPointStatus	The patient might hear better than indicated at this point, since all presentation levels are reported as heard.
AM · See TPresentType	Amplitude Modulated (AM)
AMModSize1 · AMModSize2 ·	The AM Modulation in centiBel (or dB x 10). Variables in the structure TMeasCond of type TdB10.
AM - Test	In this test, the intensity of the stimulus is varied continuously and the patient must indicate the smallest variation that he can detect. The test is also called Difference Limen Intensity (DLI) test or also the Lüscher-Zwislocki test.
ANSIS36 · See TTransCalStand	Transducer Calibration according to ANSI standard S3.6 for audiometers. 1989. See Draft # 5 1995 for a proposed revision of the standard. Measurement according to this standard.

ANSIS36_FFEQ ·	Free Field Equalized (FFEQ) . The Transducer stimulus is equalized with the transfer function of the free field measurement (loudspeaker + Room frequency response).
Audiometric session ·	A Complete Audiometric session excl. patient data etc. can be saved in the TAudioSession structure.
AUX · see Tsignal	Auxiliary Signal (AUX) : Stimulus input from Compact Disc or Audio Tape.
AUXParm1 · AUXParm2 ·	Description of external signals in channel 1 and 2 respectively, as described in TAUXParm. AUXParm1 and 2 are variables in the structure TMeasCond of type TAUXParm.

B

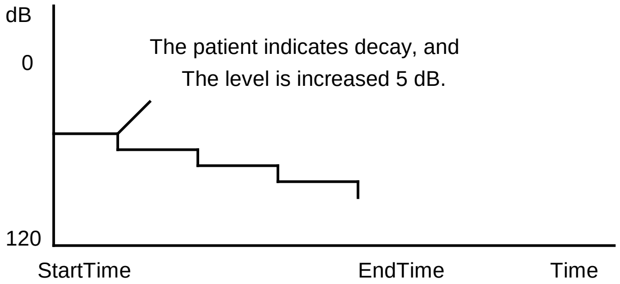
B71, B72 · See TTransType	B71 and B72 are Bone Conduction Vibration Devices.
BCBin · See TSignalOutput	Bone Conductor "Binaural" (BCBin) : Two Bone Conductors are used as stimuli, one on each side of the patient's jaws.
BCL · See TSignalOutput	Bone Conductor Left (BCL) : A bone conductor is used as stimulus. The conductor is placed at the patient's left jaw, or, in medical terms: "bone conductors are usually handheld against the mastoid area, the projecting part of the temporal bone behind the ear". Ref. [HOCA-4] Chapter 9, Bone-Conduction Threshold Testing.
BCR · See TSignalOutput	Bone Conduction Right (BCR). A bone conductor is used as stimulus. The conductor is placed at the patient's right jaw.
Beoton · See TTransType	The Beoton is a Bone Conduction Vibration Device.

C

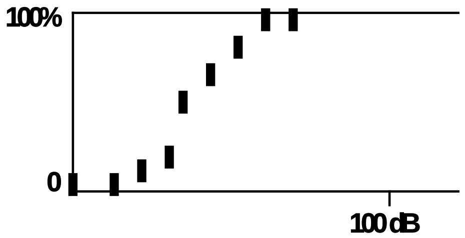
Complete DLI Curve ·	A Complete Difference Limen Intensity (DLI) Curve is saved using the structure TDLIAudiogram.
Complete MCL Curve ·	A Complete Most Comfortable Loudness (MCL) Curve is saved in the structure TToneMCLAudiogram.
Continuous · See TpresentType	Continuous presentation of the stimulus.
CSL · See TdBWeighting	Comfortable Speech Level (CSL). According to ISO standard.
Curve ·	A variable of type TTonePoint used to store curve points in all the audiograms, i.e. THR, MCL, UCL, ABAB, Stenger, DLI, DLF, SiSi, Decay, SpeechDL, SpeechSRT, SpeechMCL and SpeechUCL Audiograms.

D

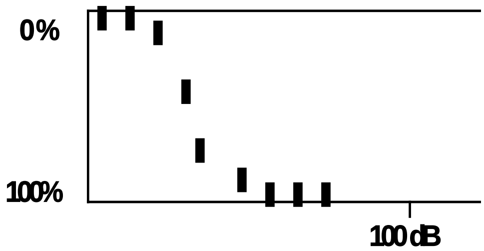
DataFmtCodeStd ·	Constant, defined in the beginning of AUDDEF.H (= 100). This Constant makes a version check possible.
DataTypeCode ·	Constant, defined in the beginning of AUDDEF.H (= dtc_Audiogram)
dBWeighting1 · dBWeighting2 ·	Variable of type TdBWeighting in structure TMeasCond.
Decay Audiogram · Decay Curve ·	A Decay Audiogram shows the ability of a patient (PT) to hear a sustained tone. PT indicates with an answer contact when the tone is decayed (not heard) and the tone intensity is raised e.g. 5 dB, and the test is repeated.



Decay sections ·	The Decay curve is intersected by 5 dB steps as a result of the measuring method described above.
DichoticWords · see TAUXParm	Dichotic: Affecting or relating to the two ears differently in regard to one aspect of the stimulus. DichoticWords: External signal description.
Difference Limen Threshold dB ·	The Difference Limen Intensity (DLI) Threshold is saved in the variable ModSize, Type TdB10 in the structure TDLIPoint. (dB x 10)
Difference Limen Threshold in pct. ·	The Difference Limen Frequency (DLF) Threshold is saved in the variable ModSize, Type TPct100 in the structure TDLFPoint. (% x 100).
DL Curve ·	The Discrimination Loss (DL) Curve shows the score when reproducing words (single or two …) presented at different levels:



The Discrimination Score shows the same graph as the DL, but with the y-axis is pointing downwards:



DLFAudiogram DLF Curve ·	The result of a Difference Limen Frequency (DLF) Test. This test is similar to the AM-test, the only difference being that in the DLF test, the frequency is varied instead of the amplitude. The resulting DLF Curve shows the patient’s threshold for hearing small frequency variations.
DLF threshold · DLI threshold ·	The patient’s threshold for hearing a Frequency / Intensity modulation.
DLI Test	This test is also called Amplitude Modulation (AM) Test or the Lüscher-Zwislocki test. Essentially, the intensity is varied continuously and the patient must indicate the smallest variation that he can detect.
DT48 · See TTransType	The DT48 is a special headset used in audiometry.
dtc_Audiogram ·	The Constant DataTypeCode is set to this defined value (1).

E

EARTONE3A · See TTransType	Special insert-phone used in audiometric measurements.
EndTimeSec ·	End time for a section of a Decay curve.

F

FFBin · See TSignalOutput	Free Field Binaural (FFBin). Sound presented from loudspeakers to both the patient’s ears.

FFL · See TSignalOutput	Free Field Left (FFL). Sound presented from a loudspeaker to the patient's left ear.
FFR · See TSignalOutput	Free Field Right (FFR) Sound presented from a loudspeaker to the patient's right ear.
FM · See TPresentType	Frequency Modulated (FM) The Frequency Modulation is done stepwise, e.g. 1, 2, 3, 4, 5 % of the frequency of the steady tone used.
FMModSize1 · FMModSize2 ·	Frequency Modulation: Modulation Size (FMModSize). Variables of the type TPct100 in the TMeasCond structure.
Freiburger · see TAUXParm	A special test sequence of words used for Speech audiometry. Used in Germany. The sequence of words is presented from CD player or tape.
Freq1 · Freq2 ·	The variables Freq (Frequency) of the type THertz are found in the structures TTonePoint, TDLIPoint, TDLFPoint and TSiSiPoint.

H

HDA200 · See TTransType	The HDA200 is a Sennheiser Headset used in audiometry.
Holmberg · See TTransType	The Holmberg Headset is used in audiometry.
HTL · See TTransCalStand	Hearing Threshold Level. This Transducer Calibration Standard is the default for all the 13 audiogram types that can be saved in NOAH.

I

Impulse · See TPresentType	Random Pulses, e.g. 200 ms; Variable intensity.
Increment Size ·	See SiSi Increment Size.
Intensity1 · Intensity2 ·	The intensity variables of type TdB10 (or centiBel) are found in the structures TTonePoint,TDLIPoint, TDLFPoint and TSiSiPoint.
IPBin · See TSignalOutput	Insert Phone Binaural (IPBin) Transducers placed in both the patient's ears.
IPL · See TSignalOutput	Insert Phone Left (IPL) - A transducer placed in the left ear.
IPR · See TSignalOutput	Insert Phone Right (IPR) - A transducer placed in the right ear.
ISO226	ISO 226: "Acoustics - Normal Equal-Loudness Level Contours First

· See TTransCalStand	Edition ". 1987. Supersedes ISO 454.
ISO226_FFEQ · See TTransCalStand	Free Field Equalised (FFEQ) . The Transducer stimulus is equalised with the transfer function of the free field measurement (loudspeaker frequency response). See ISO 226 above.
ISO389 · See TTransCalStand	ISO 389: "Acoustics - Standards Reference Zero for the Calibration of Pure-Tone Air Conduction Audiometers". Third edition, 1991. ISO 389-2: "Acoustics - Reference Zero for the Calibration of Audiometric Equipment" Part 2: "Reference Equivalent Threshold Sound Pressure Levels for Pure Tones and Insert Earphones". First Edition, 1994. ISO 389-3: "Acoustics - Reference Zero for the Calibration of Audiometric Equipment". Part 3: " Reference Equivalent Threshold Force Levels for Pure Tones and Bone Vibrators". First Edition, 1994. ISO 389-4: "Acoustics - Reference Zero for the Calibration of Audiometric Equipment". Part 4: "Reference Levels for Narrow-Band Masking Noise". First Edition. Supersedes ISO 8798.
ISO389_FFEQ · See TTransCalStand	Free Field Equalised (FFEQ). The Transducer stimulus is equalised with the transfer function of the free field measurement (loudspeaker frequency response). See ISO 389 above.
ISO7566 · See TTransCalStand	ISO 7566: Acoustics - Standards Reference Zero for the Calibration of Pure - Tone Bone Conduction Audiometers". First Edition 1987. (Withdrawn !)
ISO7566_FFEQ · See TTransCalStand	Free Field Equalised (FFEQ). The Transducer stimulus is equalised with the transfer function of the free field measurement (loudspeaker frequency response). See ISO 7566 above.
ISO8798 · See TTransCalStand	ISO 8798: "Acoustics - Reference Levels for Narrow - Band Masking Noise". First Edition.(Replaced by ISO 389-4 !)

L

LastAUXParm · see TAUXParm	End of AUX parameters define list.
LastCondition · See TCondition	End of Conditions define list.
LastdBWeighting · See TTransCalStand	End of dBWeighting define list.
LastPointStatus · see TPointStatus	End of Point Status define list.
LastPresentType · See TPresentType	End of PresentType define list.

LastSignalOutput · See TSignalOutput	End of SignalOutput define list.
LastSignal . see Tsignal	End of Signal define list.
LastTransCalStand · See TTransCalStand	End of Transducer Calibration Standards define list.
LastTransType · See TTransType	End of Transducer Type define list.

M

mask channel ·	The mask channel is defined as channel 2 in the structures TTonePoint,TDLIPoint, TDLFPoint and TSiSiPoint.
MCL point ·	Most Comfortable Loudness (MCL). The MCL Points are saved in the variable Curve[ ] in the structure TToneMCLAudiogram.
MCLAudiogram ·	An Audiogram showing the Most Comfortable Loudness of hearing level. The MCLAudiogram can be based on Tone or Speech.
MeasCond ·	The structure MeasCond (Measuring Conditions) is included in all the 13 different audiograms that form the AudioSession structure. There are a total of 78 Measuring conditions stored in AudioSession. MeasCond essentially contains all the information about the recorded audiogram.
MIC · see Tsignal	Live voice from microphone.
ModSize ·	ModSize Recorded as the Difference Limen Threshold in %. The Modulation level is measured as a percentage of the level of the steady tone.
MonoSyllabicWords · see TAUXParm	As an Auxiliary sound source is used a CD or tape with one-syllable words is used.
MultiSyllabicWords · see TAUXParm	A CD or tape containing a recording of words with more than 2 syllables is used as an auxiliary sound source.

N

NBN · see Tsignal	Narrow Band Noise (NBN). Can be used as the masking signal of channel 2.
NoAUXParm · see TAUXParm	No parameter available for the External Signal Description.
NoCondition	Test conditions not available.

· See TCondition

NodBWeighting	Weighting unavailable (interpretation of dB values)
NOFIncrements	Total number of SiSi increments.
NoPresentType · See TPresentType	Invalid presentation of the signal. (Error condition)
NoResponse · see TPointStatus	Patient cannot hear this point.
NoSignal · see Tsignal	Channel without any signal.
NoSignalOutput · See TSignalOutput	No output transducer connected.
NoStatus · see TPointStatus	No special status ! Status for valid code points !
NoTransCalStand · See TTransCalStand	No Transducer Calibration Standard available.
NoTransType · See TTransType	No Transducer Type Description available.
Numerals · see TAUXParm	A CD or tape containing a recording of numerals is used as an auxiliary sound source.

O

OffTime1 · OffTime2 ·	The duty cycle of the presented tones can be expressed in terms of Ontime and Offtime. Channel 1: Stimulus. Channel 2: Masking signal.
OnTime1 · OnTime2 ·	(see Offtime above)

P

patient data ·	The Patient data are added to the TAudioSession structure. The definition lies outside AUDDEF.H.
PN	Pink Noise (PN) used as masking signal. As defined by ISO.
PointStatus	Status of threshold point. See AUDDEF.H for the range of possible defined values. Note that NoStatus is the value defined for stored, valid curve points.
PresentType1 · PresentType2 ·	Presentation of channel 1: signal and channel 2: Masking or additional signal. The variables PresentType of the type TPresentType are parts of the TMeasCond structure.

Pulse
· See TPresentType

Pulse is an example of Signal Presentation where a tone is presented with on and off intervals.

PulseDutyCycle1 ·
PulseDutyCycle2 ·

The PulseDutyCycle values of the type TPct100 are found in the TMeasCond structure. They express the Ontime in pct. / Offtime x 100 relation.

PulseModFreq1 ·

The Pulse Modulation Frequency (PulseModFreq) variables of type THerz10 are found in the TMeasCond structure. The frequency is 1 / (Offtime + Ontime).

R

Reim · see TAUXParm

A special sequence of words recorded on CD or tape. Used as a signal when recording a speech audiogram. Originates from Germany.

S

ScorePct ·

Score based upon number of words. This is the normal way of recording the result of a speech audiogram. It expresses the percentage of words reproduced correctly. The type is TPct100 or pct x 100.

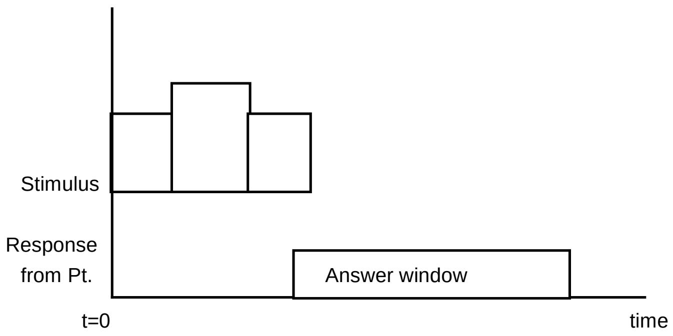
SignalOutput1 ·
SignalOutput2 ·
· see TSignalOutput

The output Transducer choice for stimulus (channel 1) and masking or additional stimulus(channel 2) is saved in these variables.

SignalType1 ·
SignalType2 ·
· see TSignal

The two variables: SignalType of type TSignal are found in the MeasCond structure. SignalType1 contains the description of the stimulus. SignalType2 can contain the description of an additional stimulus or a masking signal.

SiSiAudiogram ·



Example of SiSi-test: Amplitude Modulation (AM) test

with a predefined sequence of stimuli.

The patient answers within the specified time interval whether he can hear the amplitude modulation.

SiSi Increments ·

The value of an Amplitude Modulation measured in dB x 10 (centiBel) in a SiSi test.

SiSi points ·	A SiSiAudiogram consists of up to 24 SiSi points.
SiSiParm1 · SiSiParm2 ·	The 2 SiSi parameters of type TdB10, dB x 10 or centiBel are found in MeasCond.
SiSiScores ·	The SiSiAudiogram consists of up to 24 points, all containing SiSi scores.
SiSiTable ·	The word SiSitable is in AUDDEF.H a synonym for SiSiAudiogram.
SN · see Tsignal	Speech Noise. As defined by ISO.
SpeechSRTAudiogram ·	The basic Speech Reception Threshold (SRT) Audiogram. The word "speech" is added in order to separate tone and speech audiograms; see AUDDEF.H. The SRT is defined as the 50 pct score curve (half of the presented phonemess are reproduced correctly).
SPL · See TTransCalStand	Sound Pressure Level. Standard Physical Calibration with 0 dB SPL equal to 20 µPa.
SRT Curve · SRT points ·	Speech Reception Threshold (SRT). See SRT Audiogram above. A curve[] structure holds a number of saved and valid curve points.
StartTimeSec ·	Start time for a decay section. See DecayAudiogram.
Status ·	A variable of type TPointStatus found in the structures TTonePoint,TDLIPoint and TDLFPoint.
StengerAudiogram	The Stenger test is used when a patient claims that his hearing is impaired in one ear. It involves presentation of a tone of one frequency to both ears. The audiometer should allow separate intensity control for each channel. The tone is presented to the good ear first at a level 5-10 dB above the known threshold. The tone is then gradually presented to the "bad" ear until it is at a point some 10 dB louder than the level of the tone in the good ear. If the patient is simulating a hearing loss, he will hear the tone louder in the bad ear but still claim that he hears it with the good ear. The level of the tone in the good ear is then dropped below its threshold point and entirely removed from the good ear. If the patient has a loss as measured, he will report hearing the tone in the bad ear when it exceeds the actual hearing loss by 5 - 10 dB. Once the intensity of the tone in the bad ear is sufficient to reach the actual threshold, an increase in intensity of 10-15 dB will cause the tone to be heard in the bad ear.
Stenger Curve ·	The result of a Stenger Test. (Speech Audiogram).

stim channel ·	The Stimulation Channel is defined as channel 1 in AUDDEF.H. Please note that channel 2 can either be unused, or it can contain an additional stimulus (Binaural tests) or masking signal.

T

TABLBAudiogram	Example of Type definition in AUDDEF.H. See explanation under ABLBAudiogram, which is a variable of the type preceded by a "T".	
Type definitions in AUDDEF.H:	TAudioSession	TSiSi Increments
	TAUXParm	TSiSiAudiogram
	TCondition	TSiSiIncrements
	TdB10	TSiSiPoint
	TdBWeighting	TSpeechDLAudiogram
	TDecayAudiogram	TSpeechMCLAudiogram
	TDecayPoint	TSpeechPoint
	TDLFAudiogram	TSpeechSRTAudiogram
	TDLFPoint	TSpeechUCLAudiogram
	TDLIAudiogram	TStengerAudiogram
	TDLIPoint	TTime100
	THertz	TTime1000
	THertz10	TTone
	TMeasCond	TTonePoint
	TPct100	TToneTHRAudiogram
	TPresentType	TTransCalStand
	TPointStatus	TTransType
	TSignal	TWords
	TSignalOutput	
TDH39 , TDH49 · See TTransType	Two different types of Teletronics headsets used for audiometric measurements.	
Threshold Curve ·	Collection of curve points: Curve[ ] with information about the patient's Hearing Threshold Level (HTL).	
Tone · see Tsignal	Pure Tone. For all tone audiograms, the variable SignalType1 of type TSignal must be set to the value Tone (2).	
ToneTHRAudiogram ·	This audiogram records the patient's Hearing Threshold Level (HTL).	
TransCalStand1 · TransCalStand2 · · see TTransCalStand	Transducer Calibration Standard (TransCalStand) : 2 variables of the type TTransCalStand in the MeasCond structure.	
TransType1 · TransType2 · · see TTransType	Transducer Type (TransType) : 2 variables of the type TTransType in the MeasCond structure.	

U

UCLAudiogram ·	Uncomfortable Loudness (UCL) points can be recorded as both Tone

UCL point ·
Uncomfortable Loudness Audiogram ·

and Speech Audiograms. They indicate combinations of stimulus level and frequency where the patient indicates that the presented sound was uncomfortably loud !

UnAided
· See TCondition

UnAided means Measured without a Hearing Aid.

UndefInt ·

The “Undefined Integer” (UndefInt) is in AUDDEF.H defined for all variables in MeasCond. It is defined as decimal -32767, hex 80 01.

UnKnown ·

The value “Unknown” is legal for all variables in MeasCond. It is defined as 0.

AUDDEF.H contains the following explanation:
“Note that values >= <Last parameter in list of defined values> should be set to UnKnown to ensure that extensions to the definedvalues can be dealt with.”

However, extensions have to be defined and supported by the NOAH database structure in order to avoid misinterpretations when exchanging audiograms, i.e. values still have to be defined by AUDDEF.H.

W

Warble
· see Tsignal

A Steady tone on which a tone of different amplitude and frequency is superimposed results in a “warble” tone.

WarbleModFreq1 ·
WarbleModFreq2 ·

Warbletone Modulation Frequency: 2 variables of the type THerz found in the MeasCond structure. A warble tone of the indicated frequency is superimposed on a steady tone. If 0, no warble tone is present.

WarbleModSize1 ·
WarbleModSize2 ·

Warbletone Modulation Size: 2 variables of the type TPct100 found in the MeasCond structure. The amplitude of the warble tone in pct. x 100 of that of the steady tone.

WN
· see Tsignal

White Noise (WN). Noise with equal distribution in the complete frequency band 20 Hz - 20 kHz.

Words ·

A variable of the type TWord found in the structure TSpeechPoint which is included in all speech audiograms. This integer expresses the score based upon the number of words reproduced correctly during a speech audiogram test.

Appendix B: The header file AUDDEF.H

```
//----------------------------------------------------------------------------------------------
//  Description:                         (C) Rexton Danplex A/S
//----------------------------------------------------------------------------------------------
//
//  Data declaration file of Audiometric measurements for NOAH 1.3
//
//  Filename:     AUDDEF.H
//
//  Compiler:     BCW 4.02
//
//  Created by:   Leif Nielsen
//
//  Date:         22/01-1995
//
//  Modified by:            Leif Nielsen
//
//  Date:         23/05-1995
//  ---------------------------------------------------------------------------------------------
//  FORMAT:   DataTypeCode           = dtc_Audiogram (1)
//            DataFmtCodeStd         = 100
//  ---------------------------------------------------------------------------------------------
//  Status        Almost complete. Maybe the number of SpeechSRTAudiogram
//                curves will be incremented by up to 4 more curves.
//                The specification will be fixed before July 1.st
//                The 'new' and final specification will also have
//                DataFmtCodeStd=100.
//----------------------------------------------------------------------------------------------

//----------------------------------------------------------------------------------------------
//  Prevent duplicated definitions
#ifndef __AUDDEF_H
#define __AUDDEF_H

//----------------------------------------------------------------------------------------------

#include "NOAHDEF.H"
//----------------------------------------------------------------------------------------------

typedef int   THertz;              // Frequency in Hz (NOAH definition)
typedef int   THertz10;            // Frequency in Hz x 10
typedef int   TdB10;               // Decibel   x 10  (NOAH definition)
typedef int   TPct100;             // Percent   x 10  (Please Read Percent x 100)
typedef int   TTime100;                      // Time in seconds x 100 [0-320] sec.
typedef int   TTime1000;           // Time in seconds x 1000 [0-32] sec.
typedef int   TWords;              // Number of words in speech tests
typedef int   TSiSiIncrements;     // Number of SiSi Increments

//----------------------------------------------------------------------------------------------

#define UndefInt          (-32767)        // Nothing stored  (NOAH definition)
#define UnKnown             0             // Unknown... Goes for all '#defines'
                                          // Values >= LastX should be set UnKnown
                                          // to ensure that extensions to the
                                          // '#define' structure can be dealt with
//----------------------------------------------------------------------------------------------
```

```
typedef int    TPointStatus;          // Status of threshold point
//-----------------------------------------------------------------------------------------------
#define NoStatus                  1         // No Special Status
#define AlwaysResponse            2         // Patient might hear better than this
#define NoResponse                3         // Patient cannot hear this point
#define LastPointStatus                     4         // End of define List
//-----------------------------------------------------------------------

typedef int    TSignal;               // Description of signal
//-----------------------------------------------------------------------
#define NoSignal                            1         // Channel without any signal
#define Tone                      2         // Pure Tone
#define Warble                    3         // Warble Tone
#define NBN                       4         // Narrow Band Noise
#define SN                        5         // Speech Noise
#define WN                        6         // White Noise
#define PN                        7         // Pink Noise
#define AUX                       8         // Auxiliary Signal (CD / Tape)
#define MIC                       9         // Live Voice from microphone
#define LastSignal                10        // End of define List
//-----------------------------------------------------------------------------------------------

typedef int    TAUXParm               // External signal description
//-----------------------------------------------------------------------------------------------
#define NoAUXParm                 1         // No parameters
#define MonoSyllabicWords         2         //
#define MultiSyllabicWords        3         //
#define DichoticWords             4         //
#define Freiburger                5         //
#define Reim                      6         //
#define Numerals                  7         //
#define LastAUXParm               8         // End of define List
//-----------------------------------------------------------------------------------------------

typedef int    TSignalOutput;                 // Output transducer choice
//-----------------------------------------------------------------------------------------------
#define NoSignalOutput            1         // No Output connected
#define ACL                       2         // Air Conductor Left
#define ACR                       3         //                  Right
#define ACBin                     4         //                  Binaural
#define BCL                       5         // Bone Conduction Left
#define BCR                       6         //                  Right
#define BCBin                     7         //                  "Binaural"
#define FFL                       8         // Free Field       Left
#define FFR                       9         //                  Right
#define FFBin                     10        //                  Binaural
#define IPL                       11        // Insert Phone     Left
#define IPR                       12        //                  Right
#define IPBin                     13        //                  Binaural
#define LastSignalOutput          14        // End of define List
//-----------------------------------------------------------------------------------------------
```

```
typedef int    TPresentType;              // How the signal is presented
//-------------------------------------------------------------------------------------------------
#define NoPresentType           1         // Invalid presentation
#define Continuos               2         // Normal steady and unmodulated
#define Pulse                   3         // Pulsed presentation
#define ABLB                    4         // Alternating between ch.1 & ch.2
#define AM                      5         // Amplitude Modulated
#define FM                      6         // Stepwise Frequency modulated
#define Impulse                 7         // Random pulses
#define SiSi                    8         // SiSi signal (on 200 ms)
#define LastPresentType         9         // End of define List
//-------------------------------------------------------------------------------------------------

typedef int    TTransType;                // Transducer type description
//-------------------------------------------------------------------------------------------------
#define NoTransType             1         // No transducer connected
#define TDH39                   2         //
#define HDA200                  3         //
#define EARTONE3A               4         //
#define DT48                    5         //
#define TDH49                   6         //
#define B71                     7         //
#define B72                     8         //
#define Beoton                  9         //
#define Holmberg                          10        //
#define LastTransType           11        // End of define List
//-------------------------------------------------------------------------------------------------

typedef int    TTransCalStand;            // Transducer calibrated according to:
//-------------------------------------------------------------------------------------------------
#define NoTransCalStand         1         // No Standard available
#define ISO389                  2         //
#define ISO389_FFEQ             3         //
#define ISO7566                 4         //
#define ISO7566_FFEQ                      5         //
#define ISO8798                 6         //
#define ISO8798_FFEQ                      7         //
#define ISO226                  8         //
#define ISO226_FFEQ             9         //
#define ANSIS36                 10        //
#define ANSIS36_FFEQ            11        //
#define LastTransCalStand       12        // End of define List
//-------------------------------------------------------------------------------------------------

typedef int    TdBWeighting;        // Interpretation of dB values
//-------------------------------------------------------------------------------------------------
#define NodBWeighting           1         // Weighting unavailable
#define HTL                     2         // values in HTL according to Standard
#define SPL                     3         // values in SPL (dB/µV)
#define ABS                     4         // values in ABS (ex. dB/µN)
#define CSL                     5         // values in CSL according to Standard
#define LastdBWeighting         6         // End of define List
//-------------------------------------------------------------------------------------------------
```

```
typedef int    TCondition;          // Condition for the test
//---------------------------------------------------------------------------------------------
#define NoCondition      1          // Condition unavailable
#define UnAided          2          // Measured without Hearing Aid
#define Aided            3          // Measured with Hearing Aid
#define LastCondition    4          // End of define List
//---------------------------------------------------------------------------------------------

//---------------------------------------------------------------------------------------------
//  Information about each recorded curve.  [17 x 2 x 2 = 68 bytes]
//---------------------------------------------------------------------------------------------
typedef struct
{
          TSignal                   SignalType1, SignalType2;

          THertz                    WarbleModFreq1, WarbleModFreq2;
          TPct100                   WarbleModSize1, WarbleModSize2;
          TAUXParm                  AUXParm1, AUXParm2;

          TSignalOutput             SignalOutput1, SignalOutput2;
          TPresentType              PresentType1, PresentType2;

          THertz10                  PulseModFreq1 PulseModFreq2
          TPct100                   PulseDutyCycle1, PulseDutyCycle2;

          TdB10                     AMModSize1, AMModSize2;
          TPct100                   FMModSize1, FMModSize2;
          TTime1000                 OnTime1, OnTime2;
          TTime1000                 OffTime1, OffTime2; ,

          TdB10                     SiSiParm1, SiSiParm2;

          TTransType                TransType1, TransType2;
          TTransCalStand            TransCalStand1, TransCalStand2;

          TdBWeighting              dBWeighting1, dBWeighting2;
          TCondition                Condition1, Condition2;
}
TMeasCond;


//---------------------------------------------------------------------------------------------
//  Data point used to store thresholds [5 x 2 = 10 bytes]
//---------------------------------------------------------------------------------------------
typedef struct
{
          THertz          Freq1;          // Hz for stim. channel
          TdB10           Intensity1;     // dB for stim channel
          THertz          Freq2;          // Hz for mask channel
          TdB10           Intensity2;     // dB for mask channel
          TPointStatus    Status;         // Status
}
TTonePoint;
```

```
//---------------------------------------------------------------------------------------------
//  Complete Threshold Curve [68 + 24 x 10 = 308 bytes]
//---------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond     MeasCond;       // Conditions
        TTonePoint    Curve [24];     // 24 Threshold points
}
TToneTHRAudiogram;

//---------------------------------------------------------------------------------------------
//  Complete MCL Curve              [68 + 24 x 10 = 308 bytes]
//---------------------------------------------------------------------------------------------

typedef struct
{
        TMeasCond     MeasCond;       // Conditions
        TTonePoint    Curve [24];     // 24 MCL points
}
TToneMCLAudiogram;

//---------------------------------------------------------------------------------------------
//  Complete UCL Curve              [68 + 24 x 10 = 308 bytes]
//---------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond     MeasCond;       // Conditions
        TTonePoint    Curve [24];     // 24 UCL points
}
TToneUCLAudiogram;

//---------------------------------------------------------------------------------------------
//  Complete ABLBCurve [68 + 192 x 10 = 1988 bytes]
//---------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond     MeasCond;       // Conditions
        TTonePoint    Curve [192];    // 192 ABLB points with equal or different
                                      // frequencies
}
TABLBAudiogram ;


//---------------------------------------------------------------------------------------------
//  Complete Stenger Curve          [68 + 24 x 10 = 308 bytes]
//---------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond     MeasCond;       // Conditions
        TTonePoint    Curve [24];     // 24 (no. response in good ear)
}
TStengerAudiogram;
```

```
//-----------------------------------------------------------------------------------------------
//  Data point for DLI thresholds      [6 x 2 = 12 bytes]
//-----------------------------------------------------------------------------------------------
typedef struct
{
        THertz          Freq1;          // Hz for stim channel
        TdB10           Intensity1;     // dB for stim channel
        THertz          Freq2;          // Hz for mask channel
        TdB10           Intensity2;     // dB for mask channel
        TdB10           ModSize;        // Difference Limen Threshold dB
        TPointStatus    Status;         // Status
}
TDLIPoint;

//-----------------------------------------------------------------------------------------------
//  Complete DLI Curve [68 + 24 x 12 = 356 bytes]
//-----------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond       MeasCond;       // Conditions
        TDLIPoint       Curve [24];     // 24 DLI points
}
TDLIAudiogram;



//-----------------------------------------------------------------------------------------------
//  Data point for DLF thresholds      [6 x 2 = 12 bytes]
//-----------------------------------------------------------------------------------------------
typedef struct
{
        THertz          Freq1;          // Hz for stim channel
        TdB10           Intensity1;     // dB for stim channel
        THertz          Freq2;          // Hz for mask channel
        TdB10           Intensity2;     // dB for mask channel
        TPct100         ModSize;        // Difference Limen Threshold in pct.
        TPointStatus    Status;         // Status
}
TDLFPoint;


//-----------------------------------------------------------------------------------------------
//  Complete DLF Curve [68 + 24 x 12 = 356 bytes]
//-----------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond       MeasCond;       // Conditions
        TDLFPoint       Curve [24];     // 24 DLF points
}
TDLFAudiogram;
```

```
//----------------------------------------------------------------------------------------------
//  Data point for SiSiScores        [7 x 2 = 14 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
        THertz              Freq1;          // Hz for stim channel
        TdB10               Intensity1;     // dB for stim channel
        THertz              Freq2;          // Hz for mask channel
        TdB10               Intensity2;     // dB for mask channel
        TdB10               ModSize;        // SiSiIncrement Size in dB
        TSiSiIncrements     NOFHits;                // Number of Answers
        TSiSiIncrements     NOFIncrements;          // Total Number of SiSi Increments [20]
}
TSiSiPoint;

//----------------------------------------------------------------------------------------------
//  Complete SiSiTable              [68 + 24 x 14 = 404 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond           MeasCond;       // Conditions
        TSiSiPoint          Curve [24];     // 24 SiSi points
}
TSiSiAudiogram;


//----------------------------------------------------------------------------------------------
//  Data point for Decay sections      [6 x 2 = 12 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
        THertz          Freq1;          // Hz for stim channel
        TdB10           Intensity1;     // dB for stim channel
        THertz          Freq2;          // Hz for mask channel
        TdB10           Intensity2;     // dB for mask channel
        TTime100         StartTimeSec;  // Start Time
        TTime100         EndTimeSec;    // End Time for this section
}
TDecayPoint;


//----------------------------------------------------------------------------------------------
//  Complete Decay Curve [68 + 50 x 12 = 668 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
        TMeasCond       MeasCond;       // Conditions
        TDecayPoint      Curve [50];    // 50 decay sections equal or different
                                        // frequencies
}
TDecayAudiogram;
```

```
//----------------------------------------------------------------------------------------------
//  Data point for Speech Scores       [4 x 2 =  8 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
          TdB10           Intensity1;      // dB for stim channel
          TdB10           Intensity2;      // dB for mask channel
          TPct100         ScorePct;        // Speech score Hit Rate in pct.
          TWords           Words;          // Score based upon no of words
}
TSpeechPoint;

//----------------------------------------------------------------------------------------------
//  Complete DL Curve [68 + 24 x 8 = 260 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
          TMeasCond       MeasCond;        // Conditions
          TSpeechPoint    Curve [24];      // 24 DL points
}
TSpeechDLAudiogram;
//----------------------------------------------------------------------------------------------
//  Complete SRT Curve [68 + 24 x 8 = 260 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
          TMeasCond       MeasCond;        // Conditions
          TSpeechPoint    Curve [24];      // 24 SRT points
}
TSpeechSRTAudiogram;


//----------------------------------------------------------------------------------------------
//  Complete MCL Point              [68 + 1 x 8 = 76 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
          TMeasCond       MeasCond;        // Conditions
          TSpeechPoint    Curve [1];       // 1 MCL point
}
TSpeechMCLAudiogram;

//----------------------------------------------------------------------------------------------
//  Complete UCL Point              [68 + 1 x 8 = 76 bytes]
//----------------------------------------------------------------------------------------------
typedef struct
{
          TMeasCond       MeasCond;        // Conditions
          TSpeechPoint    Curve [1];       // 1 UCL point
}
TSpeechUCLAudiogram;
```

```
//----------------------------------------------------------------------------------------------
//  Complete Audiometric session excl patient data etc.
//----------------------------------------------------------------------------------------------
typedef struct
{
        TToneTHRAudiogram       ToneTHRAudiogram [6];       //  6 x  308 = 1848
        TToneMCLAudiogram       ToneMCLAudiogram [6];       //  6 x  308 = 1848
        TToneUCLAudiogram       ToneUCLAudiogram [6];       //  6 x  308 = 1848
        TABLBAudiogram          ABLBAudiogram [1];          //  1 x 1988 = 1988
        TStengerAudiogram       StengerAudiogram [1];       //  1 x  308 =  308
        TDLIAudiogram           DLIAudiogram  [2];          //  2 x  356 =  712
        TDLFAudiogram           DLFAudiogram [2];           //  2 x  356 =  712
        TSiSiAudiogram          SiSiAudiogram  [2];         //  2 x  404 =  808
        TDecayAudiogram         DecayAudiogram  2];         //  2 x  668 = 1336
        TSpeechDLAudiogram      SpeechDLAudiogram  [12];    // 12 x  260 = 3120
        TSpeechSRTAudiogram             SpeechSRTAudiogram [12];     // 12 x  260 = 3120
        TSpeechMCLAudiogram     SpeechMCLAudiogram [12];    // 12 x   76 =  912
        TSpeechUCLAudiogram     SpeechUCLAudiogram [12];    // 12 x   76 =  912
}                                                           // 19472 bytes
TAudioSession;

#endif
```

Index

2

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T

U

W