NOAH

Storing Audiological Measurements

Oto Acoustic Emissions Standard

DataFmtCodeStd 200
Version 1.1

HIMSA II K/S

The information in this document is subject to change according to the review policies established by HIMSA II K/S.

HIMSA II K/S MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OR SUITABILITY FOR A PARTICULAR PURPOSE. HIMSA shall not be liable for errors contained herein or for incidental consequential damages in connection with the supply of, performance of, or use of this material.

This document contains proprietary information that is protected by copyright. All rights are reserved. No parts of this document may be photocopied, reproduced or distributed to Non-HIMSA member companies without the prior permission of HIMSA II K/S.

Copyright © 2012 HIMSA II K/S

Preface

This is the fifth 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 measurement related data within the framework of NOAH-compatible measurement and fitting software.

The 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.

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 co-operation with its licensees, 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 licensees.

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-licensees 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.

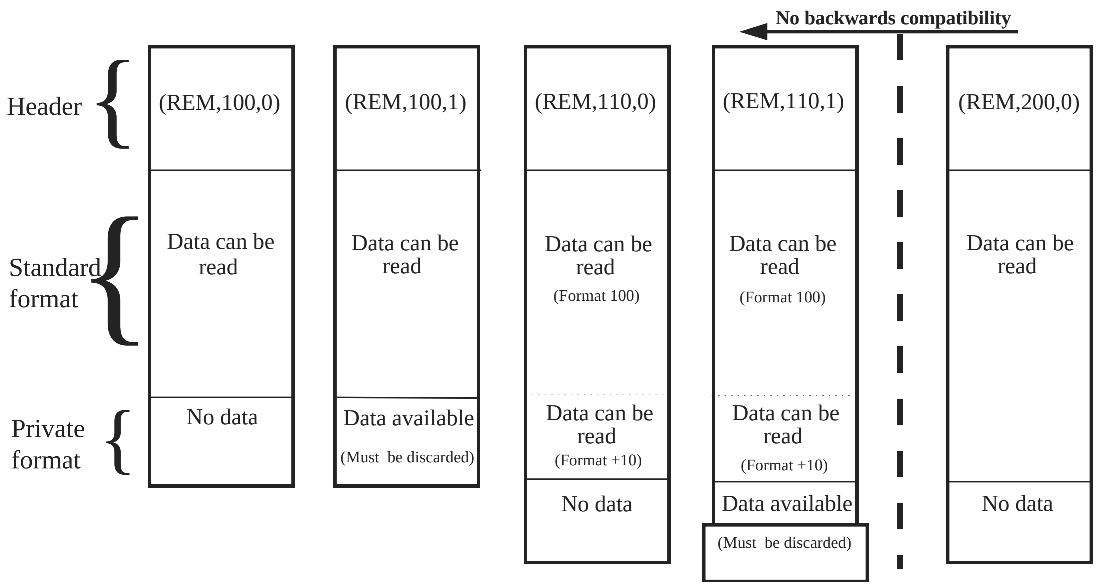


Figure 1: The handling of measurement data by NOAH

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

DataFmtCodeStd 200

ver.	1.00	99-06-30	Version 1.00 as agreed on the Standards Committee meeting in Florida in April 1999.
ver.	1.1	09-05-01	Note on extra byte for alignment – Section 2.1.3.

1 Introduction

1.1 A few words about programming with OAEDEF.H

This document intends to explain the use of the NOAH ver 2.0 standard for storing Oto Acoustic Emission Measurements according to the OAEdef.H header file. This header file written in the programming language "C" defines five different outer structures, in which measurement data can be saved including measuring conditions:

Oto Acoustic Emissions (OAE) Data

Different types of OAE Measurements: Side of tested Person:

Type of OAE Data	*Data Structure*	Left Ear *DataTypeCode*	Right Ear *DataTypeCode*
Spontaneous	T**S**OAEData	dtc_SOAE_L (9)	dtc_SOAE_R (10)
Transient **E**voked	T**TE**OAEData	dtc_TEOAE_L (11)	dtc_TEOAE_R (12)
Distortion **P**roduct Dia**gram**	T**DPGRAM**Data	dtc_DPGRAM_L (13)	dtc_DPGRAM_R (14)
Distortion **P**roduct **I**nput-**O**utput Curve	T**DPIO**Data	dtc_DPIOOAE_L (25)	dtc_DPIOOAE_R (26)
Probe Fitting	T**ProbeFit**Curve	dct_PROBEFITOAE_L (27)	dtc_PROBEFITOAE_R (28)

The aim of this document is to explain the correct use of these five OAEdef data structures. This is done by reading the five parts of the header file OAEdef.H "upside down" starting with the "outer" definition of the five outer data structures, continuing with the necessary supporting inner structure definitions, ending with the definition of all "inner" types, all defined as integers, words or floats.

This document is written as a part of the documentation for software developers of the NOAH Framework Programming Interface:

NOAH: Storing Audiological Measurements Document series		
Document Title	*Header File explained*	*Status*
Audiogram Standard *dataFmtCodeStd 100*	formats\audiogrm\AUDdef.h	Ver. 1.0 available
REM/HIT Standard *dataFmtCodeStd 200*	formats\remhit\REMHIT.h	Ver. 1.0 available.
Loudness Scaling Standard *dataFmtCodeStd 100*	formats\loudness\LSdef.h	Ver. 1.0 available.
Extended Loudness Scaling Std. *dataFmtCodeStd 110*	formats\loudness\Extended LSdef.h	Ver. 1.0 available.
Impedance Measurement Std. *dataFmtCodeStd 100*	formats\impedan\IMPdef.h	Ver. 1.0 available.
Oto Acoustic Emissions Standard *dataFmtCodeStd 200*	formats\oae\OAEdef.h	(This document) Ver 1.00 released in June 1999.
Electric Response Audiometry Std. *dataFmtCodeStd 200*	formats\era\ERAdef.h	Ver 0.90 Draft put on hold in June 1999.

Data can be exchanged across these interfaces among the NOAH modules. In this way data can be shared among different Hearing Instrument- and Audiological Equipment-manufacturers.

This document describes the Oto Acoustic Emissions Measurement format and it can be read independently of other NOAH documentation. It is intended as a starting point for interested, prospective licensees.

1.2 Contents

1.3 References

[AECP] Auditory Electrophysiology in clinical practice, by Claus Elberling and Poul Aabo Osterhammel. Oticon . Copenhagen 1989. (?).

[ANSI-C] The ANSI "C" Programming Language. By Brian W. Kernighan and Dennis M. Ritchie, Prentice Hall Software Series, Englewood Cliffs, New Jersey, Second Edition 1988.

[Framework] NOAH Framework ver. 0.85. System Architecture Specification. By Pallas Informatik A/S. Himsa, Copenhagen 1996.

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

[IEC-60711] IEC 60711 (1981-01): Occluded-ear simulator for the measurement of earphones coupled to the ear by ear inserts.

2 The NOAH standard for Oto Acoustic Emission Measurements

2.1 Data Structure

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

1. Explanation of the data structures in OAEDEF.H starting with the outer five structures defined for saving Probe Fitting, Spontaneous OAE measurements, Evoked OAE, Distortion Product DP-Gram and Distortion Product Input-Output curve DP-I/O-Curve respectively. Starting with these "outer" five structures, all constituent types are defined as we go by. (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 "C" integers, words or floats.

2. ASN.1 contains a few useful distinctions, used in this chapter to explain important places in OAEDEF.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).

2.1.1 The Integer type used in OAEDEF.H

The intention of the following table is to provide information across all the NOAH interface standards about the type Integer.

minInt	-32768 #8000 hex	Lowest negative value represented in two byte using standard "2's complement" representation. According to [Framework] , this value is **illegal** for the integer types defined in OAEdef.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 value **legal** in parameters defined as integer types in OAEdef.H according to [Framework].
Unknown	0 #0000 hex	*In Parameters:* The parameter is **defined,** however to an **unknown** value**.** *In Curve points:* Use logic here! For the types TdB10, TPct100,.. the value 0 is of course **defined** and **valid,** however for the THertz type, the value means **undefined**.
NoParam	1 #0001 hex	*In Parameters:* The parameter is normally **defined** Not Used (channel, parameter …), refer e.g. AUDdef.H, but for OAEdef.H the value 1 is defined and valid, refer the defined values for the different types.
MaxInt	32767 #7FFF hex	Highest positive value. Ref. [Framework].

2.1.2 The Float type used in OAEDEF.H

Since OAEDEF.H introduces the use of the Floating Point storage type in NOAH 2.0, this document has been provided with an appendix describing the use of this type, refer to Appendix C: The Floating Point Type used in NOAH.

The floating type used in OAEdef.h is the 32 bit type as standardised by IEEE and supported by the floating point co-processor 80x87 from Intel. In analogy with the above table provided for the Integer type, a list of named values is placed in the table below:

minFloat	$-2.0 * 2^{127}$ # BF FF FF FF	Lowest negative value represented as sign bit = 1, Exponent = 127 and Mantissa = $2.0 – 2^{-23} = \sim 2.0$
undefFloat	(yet to be defined)	1. Candidate for undefFloat: 1.0/0.0 = PLUS_INFINITY, #7F 80 00 00.
smallestPosFloat	$+1.0 * 2^{-126}$ #00 80 00 00	The Smallest Positive Float with sign bit = 0, Exponent = - 126 and Mantissa exactly equal to 1.0
+0	+ 0.0 #00 00 00 00	NULL. The value 0.0 has two different representations, +0.0 and – 0.0, dependent on the sign bit. 2. Candidate for undefFloat.
- 0	- 0.0 #80 00 00 00	MINUS_NULL (See explanation for +0.0). 3. Candidate for undefFloat.
maxFloat	$+2.0 * 2^{127}$ # 3F FF FF FF	Highest positive value represented as sign bit = 0, Exponent = 127 and Mantissa = $2.0 – 2^{-23} = \sim 2.0$
notANumber	Exponent field E = 255 or 0, all Mantissas	E = Exponent + 127, where E is the number represented in the Float storage. The two values E = 255 and E = 0 are reserved for signaling error from e.g. division by 0.

2.1.3 Definition of OAE 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.

OAEDEF DEFINITIONS ::=

IMPORTS ALL FROM Noahdef -- noahdef.h

-- DataFmtCodeStd = 200

2.1.3.1 TProbeFitCurve

Probe Microphone Fitting Curve

TProbeFitCurve	The probe fit curve is used to check that the probe is properly inserted in the patient's ear. The probe fitting curve consists of a time curve with 128 points. An FFT and an amplitude characteristic for the system probe - ear can be derived from the time curve.
Procedure for Probe Microphone Fitting	A click stimulus is chosen when measuring the Probe Microphone Fitting (ProbeFit). Reference [EACP] The resulting time curve is measured in coupler 711. For information about this type of coupler, refer [IEC-60711]. The time curve finally saved must be expressed in physical units, i.e. micropascal (µPa). The microphone sensitivity will be expressed indirectly through the time curve. The time curve or sampled click response can also be saved after being corrected with the microphone frequency characteristic. This can be achieved by passive filtering in order to smoothen the microphone frequency response or it can be done by digital filtering. If the time curve is corrected for the microphone frequency characteristic, the boolean "TimeCurvesCorr" will indicate this by the value 'TRUE'. The saved microphone frequency characteristic is defined as showing the microphone sensitivity as relative to 1 kHz, i.e. its value at 1 kHz is defined as being 0 dB. The saved microphone frequency characteristic is this independent of the calibration which is described in physical measures (see above). It is also independent of the chosen measuring method, except that the acoustic coupler being utilised at the measurement must be 711. Reference [IEC-60711]. Identical probefit FFTs are obtainable for different manufacturers in this way.

(TProbeFitCurve continued)

The preferred procedure when exchanging OAE data via the NOAH database and the OAEdef.h interface becomes:

1 Create an FFT from the time curve

2 Recalculate the FFT to an amplitude characteristic in dB.
(or centibel, dB x 10)

3 If "TimeCurvesCorr" is TRUE, the saved time curve then contains the microphone characteristic correction.

4 If “TimeCurvesCorr" is FALSE, the microphone characteristic is subtracted from the FFT resulting from the time curve.

Note: If the probe fit is leaky, the amplitude characteristic will have a low level at low frequencies. The structure includes measurement parameters.

Measuring Format: DataTypeCodes used for Probe Fitting Curve Measurement:
Side of tested person:

Left	*Right*
dtc_PROBEFITOAE_L (27)	dtc_PROBEFITOAE _R (28)

```
TProbeFitCurve ::= SEQUENCE {
    timeCurvesCorr          BOOLEAN,            -- See explanation above
    probeMic                TprobeMicCurve,     -- Probe mic amplitude characteristic
    level                   TdB10,              -- Stimulus level used
    accMeas                 INTEGER,            -- Accepted measurements
    rejMeas                 INTEGER,            -- Rejected measurements
    samplePeriod            FLOAT,              -- Sample period in milliseconds (ms)
    sample SEQUENCE OF fittingNSamples  FLOAT   --Amplitudes in micropascal (µPa)
}
```

2.1.3.2 FittingNSamples

Probe Microphone Fitting Number of Samples

FittingNsamples: The probe fit curve is used to check that the probe is properly inserted in the patient’s ear. The probe fitting curve consists of a time curve with fittingNSamples (128) samples or measuring points. An FFT and an amplitude characteristic for the system probe - ear can be derived from the time curve.

fittingNSamples INTEGER ::= 128

2.1.3.3 TProbeMicCurve

Probe Microphone Curve

TProbeMicCurve	If correcting spectra (TEOAE and probefit) with the microphone amplitude characteristic, it is necessary to save the microphone curve.
minFreq	The Minimum Frequency specifies the frequency of index 0 in the "Sample" array.
maxFreq	The Maximum Frequency specifies the frequency of index validSamples-1 in the array.
validSamples	Up to probeMicNSamples (1024) samples can be saved as curve points. The actual number of valid samples may be lower. Therefore, the samples in the array are equally spaced with a frequency distance of (maxFreq - minFreq) / (validSamples –1).
sample	Note that the samples are an ordered collection of data, each sample represents a Sound Pressure Level in centibel or dB x 10.

```
TProbeMicCurve ::= SEQUENCE {
{
    minFreq                    THertz,                      // Freq corresponds to first sample
    maxFreq                    THertz,                      // Freq corresponds to last sample
    validSamples               INTEGER,                     // Number of valid samples
    sample SEQUENCE OF probeMicNSamples TdB10               // Amplitudes in dB10 SPL
}
```

2.1.3.4 ProbeMicNSamples

Probe Microphone Curve – Number of samples

probeMicNSamples	Up to probeMicNSamples (1024) samples can be saved as curve points. The actual number of valid samples may be lower.

probeMicNSamples INTEGER ::= 1024

2.1.4 Spontaneous OAE Data

2.1.4.1 TSOAEData

Spontaneous OAE Data

Spontaneous Oto Acoustic Emissions Basic Description Ref. [HOCA-4] chapter 29	Spontaneous Oto Acoustic Emissions (SOAE) are more or less continuos narrowband signals emitted by about 50 pct. of human ears even in the absence of external acoustic stimulation. Their existence was first postulated by Gold in 1948, but the first extensive measurements were reported by Kemp (1979) and Zurek (1981). (..) SOAEs are relatively simple to measure: A probe containing a sensitive, low-noise microphone is placed in the external ear canal. The shape of the probe is similar to those used in Immitance testing, and Immitance tips are frequently adapted for use in measuring OAEs (..) The output of the microphone is generally led to a preamplifier and high-pass filter. It is usually necessary to filter out body noise and external noise below 3-400 Hz. The output of the preamplifier and filter is then led to an FFT analyser. There is general agreement that SOAEs in humans are concentrated in the frequency region from 1-3 kHz, but they have been observed between 0.5 and 9.0 kHz. They range in amplitude from about -25 dB SPL up to 20 dB SPL, with the majority falling between -10 and +10 dB SPL. Audible SOAEs up to 50 dB SPL have been reported in cats, dogs and in Humans. In spite of detailed investigation, it has not been possible to prove a relation to the tinnitus phenomenon.
maxMeasNo	Up to 6 measurements can be saved in the unordered collection defined by the following SET:

TSOAEData ::= SET OF maxMeasNo TSOAECurve

2.1.4.2 TSOAECurve

Spontaneous OAE Curve

TSOAECurve	SOAE data consists of an amplitude spectrum and 10 fix point (frequencies) to indicate responses. The record also includes measurement parameters.
maskSignal	Masking signal type applied to the other ear (contra lateral ear) . Refer para.2.1.10.2: TMaskSignal on page 30.
maskFreq	Frequency of the masking signal applied.

maskLevel	Level of the masking signal applied.

TSOAECurve Continued

accMeas	Number of accepted measurements.
rejMeas	Number of rejected measurements.
nrLevel	Noise rejection level measured in centibel or dB x 10.
minFreq	Frequency corresponding to the first sample in the SEQUENCE OF 1024 TdB10, i.e. the sample[0].
maxFreq	Frequency corresponding to the sample in the SEQUENCE OF 1024 TdB10 which is numbered (validSamples – 1), i.e. sample[validSamples – 1].
validSamples	Number of valid samples placed in the SEQUENCE OF sOAENSamples TdB10. sOAENSamples is defined 1024. Up to sOAENSamples (1024) samples can be saved as curve points. The actual number of valid samples may be lower. Therefore, the samples in the array are equally spaced with a frequency distance of (maxFreq - minFreq) / (validSamples – 1).
sample	The samples in the array
markIdx	The 10 Marked frequencies Index (fix points) are used to indicate local maxima. They contain the index in the sample array of a local maximum in an emission.
FORMAT:	DataTypeCode =dtc_SOAE_L (9) or dtc_SOAE_R (10) DataFmtCodeStd=200

```
TSOAECurve ::= SEQUENCE {
    maskSignal        TMaskSignal,                      -- Masking signal type applied
    maskFreq          THertz,                           -- Masking signal frequency
    maskLevel         TdB10,                            -- Masking Signal Level
    accMeas           INTEGER,                          -- Accepted measurements
    rejMeas           INTEGER,                          -- Rejected measurements
    nrLevel           TdB10,                            -- Noise rejection level
    minFreq           THertz,                           -- Freq corresponding to first sample
    maxFreq           THertz,                           -- Freq corresp to sample validSamples-1
    validSamples      INTEGER,                          -- Number of valid samples
    sample SEQUENCE OF sOAENSamples TdB10,              -- Amplitude in dB SPL
    markIdx SEQUENCE OF 10 INTEGER                      -- Marked frequencies Index
}
```

2.1.4.3 SOAENSamples

Spontaneous OAE – Number of samples

SOAENSamples	The amplitude spectrum typically comes from an FFT analysis, so 1024 is a good number.

sOAENSamples INTEGER ::= 1024

2.1.5 Transient Evoked OAE Definitions

2.1.5.1 TTEOAEData

Transient Evoked Oto Acoustical Emissions

TTEOAEData	The outer structure for transient evoked OAE called TEOAEData consists of 6 time response curves of type TTEOAECurve.
Transient Evoked Otoacoustic Emissions Ref. [HOCA-4]	Transient Evoked Otoacoustic Emissions (TEOAEs) also referred to as click evoked OAEs are frequency dispersive responses following a brief acoustic stimulus, such as a click or tone burst. Because this was the first emission type reported in the literature by D. T. Kemp in 1978, the term *evoked otoacoustic emissions* is often applied specifically to transient evoked emissions. They are also known as *Kemp echoes*, and *delayed evoked otoacoustic emissions.* TOAEs (i.e. , TEOAEs) are obtained by using synchronous time-domain averaging techniques similar to those used to measure auditory evoked potentials. (..) A sealed probe in the patient's ear containing sound ducts for a microphone and a stimulus transducer is inserted in the patient's ear canal. Responses to several stimuli (e.g. 500-2000) are averaged to improve the signal-to-noise ratio. The ear canal sound pressure is amplified by a factor 100- 10 000, and high-pass filtered at 3-400 Hz. It is then sampled at a sampling rate of 40-50kHz. The first few milliseconds of the response are normally eliminated in order to remove the stimulus. One of the most important characteristics of the response is that it is frequency dispersive - high frequencies emerge sooner (i.e. have shorter latency) than low frequencies. This frequency dispersion is consistent with frequency coding along the basilar membrane, i.e. high frequencies are coded basally, whereas low frequencies are coded apically. The latencies of emission components are roughly twice that of forward travel time for any given frequency. This supports the hypothesis that an emission of a particular frequency originates from the cochlear location tuned to that frequency. It is important to note that if we had used a different time window / filtering / stimulus, emission components would be present at higher and lower frequencies, depending on the parameters chosen. The measured response is determined by the evoking stimulus and recording parameters as well as the status of the peripheral auditory system.

```
TTEOAEData ::= SEQUENCE {
    timeCurvesCorr          BOOLEAN,             -- Ref. [Framework chapter 3]
    probeMic                TProbeMicCurve,      -- Ref. 2.1.3.3 TProbeMicCurve
    data SET OF maxMeasNo   TTEOAECurve          -- Ref. 2.1.5.2 TTEOAECurve
}
```

2.1.5.2 TTEOAECurve

Transient evoked Oto Acoustic Emission (TEOAE)

TTEOAECurve	Transient Evoked Oto Acoustic Emissions (TEOAE) curve. Acoustic emissions are measured and discrete samples are saved spaced by a fixed sample time T_s
Response Curve	Each time response curve consists of the actual samples, data qualifiers and some parameters describing the measurement. The curve represents 512 discrete points in time measured at a given sample rate.
A-B method	SampleA and SampleB are measured alternately. The sum of curves A+B is interpreted as the resulting curve and the difference A-B is interpreted as the noise.
	Explanation to the components of the TTEOAECurve:
maskSignal	Masking signal type applied to the other ear (contra lateral ear) . Refer para.2.1.10.2: TMaskSignal on page 30.
maskFreq	Frequency of the masking signal applied.
maskLevel	Level of the masking signal applied.
stimPar	The stimulus parameter is imported from ERAdef.h. It is reprinted in this document in para.2.1.5.4: TTEOAEStimPar on page 20.
stimLevel	SPL stimulus level. The Peak Equivalent SPL level must be applied for click stimuli. Refer TTEOAEStimPar and the definition given in [AECP]. Sound Pressure Level of the stimulus measured in centibel or dB x 10.
stimAdj	Stimulus adjustment defined INTEGER. The actual level will change from the desired level if e.g. the ear volume is not the same as when calibrating the probe in a coupler (normally 2cc). 1 The stimulus level is adjusted using Coupler 711, Reference [IEC-60711].i.e. it is not compensated for the actual acoustical conditions. 2 The stimulus level is Cavity Corrected i.e. it is adjusted to compensate for the different volume actually used at the measurement. 3 The stimulus level is In Situ Corrected, i.e. it is adjusted by using the probe microphone placed in the test persons ear so the actual level can be measured.
stimSuppress	Number of milliseconds to suppress after the stimulus onset.
linAquisMode	The Linear Acquisition Mode: TRUE Linear FALSE Non Linear

(TTEOAECurve continued)

accMeas — Number of accepted measurements.

rejMeas — Number of rejected measurements (due to noise induced by muscle activity)

nrLevel — Noise rejection Level measured in centibel or dB x 10.

samplePeriod — Sample period measured in milliseconds (ms). Saved in a 32-bit Float. (Refer appendix C).

sampleA
sampleB — These two sequences consist of TEOAENSamples (512) samples of unit micropascal (µPa). The Type is the “C” built-in type float, here represented as FLOAT.

qualifier — The four qualifiers of the "C" builtin type float are used for validation of data and could be correlation coefficients.

Qualifiers are for manufacturer-internal purposes until a proper definition has been agreed. The definition below is tentative (i.e. not fully worked out or developed, ref. Webster’s Dictionary !)

The qualifiers are used for validation of data and they could also be used to save the correlation coefficient in the time interval 5-20 ms.

Qualifier[0]	Qualifier[1]	Qualifier[2]	Qualifier[3]
Correlation	S/N Ratio	(not defined)	(not defined)

FLOAT — The floating point built-in type is defined as a "C" 32 bit Float with range [-3.4E38 .. –1.18E-38 OR 0.0 OR 1.18E-38 .. 3.4E38] and 7-digit precision.

Reference: Appendix C: The Floating Point Type used in NOAH.

Bit No.	*Field Length*	*Usage*
[31]	(1 bit)	Sign
[23..30]	(8 bit)	Biased exponent (incl. exponent sign)
[0..22]	(23 bit)	Significand

Measuring Format — DataTypeCode =dtc_TEOAE_L or dtc_TEOAE_R (11/12)

DataFmtCodeStd=200

```
TTEOAECurve ::= SEQUENCE {
    maskSignal        TMaskSignal,                          -- Masking signal type applied
    maskFreq          THertz,                               -- Masking signal frequency
    maskLevel         TdB10,                                -- Masking Signal Level
    stimPar           TTEOAEStimPar,                        -- Refer para. 2.1.5.4 on page 20
    stimLevel         TdB10,                                -- SPL Stimulus level  (Peak Eq for click)
    stimAdj           INTEGER,                              -- (See explanation in the table above)
    stimSuppress      FLOAT,                                -- milliseconds to suppress after stimulus
    linAquisMode      BOOLEAN,                              -- TRUE = Linear
    accMeas           INTEGER,                              -- No. of accepted measurements
    rejMeas           INTEGER,                              -- No. of rejected measurements
    nrLevel           TdB10,                                -- Noise rejection level in centibel
    samplePeriod      FLOAT,                                -- Sample period in milliseconds (ms)
    sampleA SEQUENCE OF tEOAENSamples FLOAT[2],             -- Unit: micropascal (μPa)
    sampleB SEQUENCE OF tEOAENSamples FLOAT,                -- Unit: micropascal (μPa)
    qualifier SEQUENCE OF 4 FLOAT                           -- (see explanation above)
}
```

2.1.5.3 TEOAENSamples

Transient Evoked OAE – Number of samples	
TEOAENSamples	A TEOAE response curve consists of TEOAENSamples (512) curve points recorded at a given sample rate.

tEOAENSamples INTEGER ::= 512

[2] FLOAT is not really an ASN.1 builtin type. Please assume the "C" 32-bit float builtin type in its 4-byte format. See appendix C.

2.1.5.4 TTEOAEStimPar

Transient Evoked Oto Acoustic Emission (OAE) Measurement Type

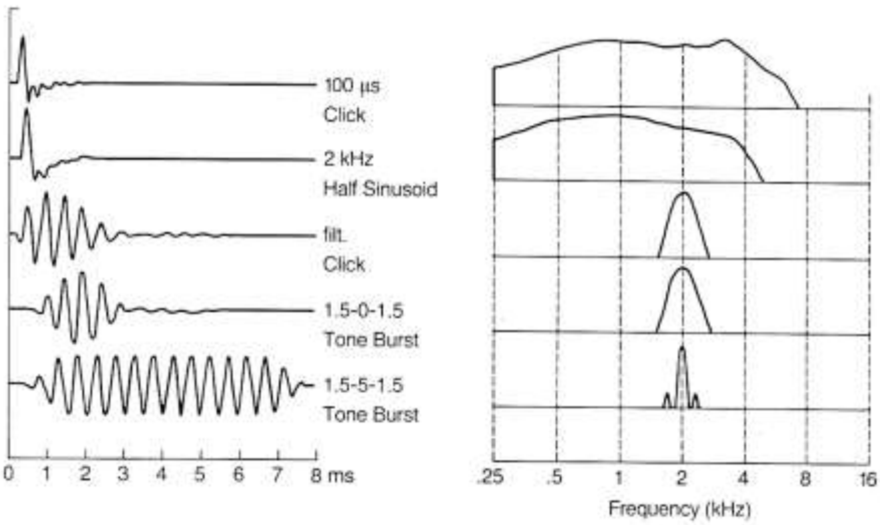
TTEOAEStimPar — The Stimulus parameter for the recording of Transient Evoked OAE. Notice the different sets of parameters for click stimulus and for tone burst stimulus[1].

This definition of stimulus type was originally defined for use within Electric Response Audiometry (ERA) but in the present document it is adopted for use when measuring Transient Evoked Oto Acoustical Emissions (TEOAE).

Click

polarity	1	Condensation	The polarity of the stimulus leads to a state of maximum pressure in the resulting sound wave (Compare with “rarefaction” below)
	2	RareFaction	“A state or region of minimum pressure in a medium transversed by compression waves (as sound waves)” (Websters Dictionary, 1980)
clkType	1	Half Wave click	See “Half sinusoid” in the fig. below from [AECP].
	2	Full Wave click	See “100 us Click” in the fig. below from [AECP]
	3	Filtered click	See “filtered click” in the figure below from [AECP]

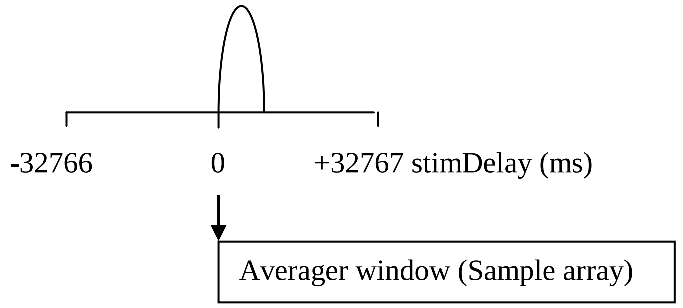
Figure from [AECP] showing the different stimuli used including their spectra. Note the trade-off between a narrow signal in time and a well-defined signal in the frequency domain.



[1] In "C" this is defined by a "union" construction: Either of two different interpretations of the 8 byte format can be used. This is expressed in ASN.1 by the CHOICE construct shown in the standard text on the following page.

(TTEOAEStimPar Continued)

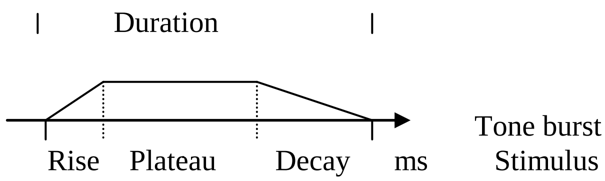
duration (click stimuli)	Click stimulus duration in microsec (µsec) . A click is generated from a sine or square wave electrical signal. The duration is measured as T, i.e. a full period of the electric stimulus signal applied to the transducer. (T can be calculated as 1 / f, where f is measured in Hertz) .
stimDelay	Start of averager window (Sample array) offset to stimulus in millisec. A negative value indicates that the stimulus ontime is later than the start of the averager window



ToneBurst

riseTime	Measured in microsec (µsec) In the figure above from [AECP] (see the previous page), the rise time is 1.5 ms.
decayTime	Measured in microsec (µsec). In the figure above from [AECP], the decay time is 1.5 ms.
duration (Tone burst stimuli)	Tone Burst stimulus duration in microsec (µsec) . A tone burst is generated from a sine or square wave electrical signal. The duration is measured as n * T, i.e. a full number of periods of the electric stimulus signal applied to the transducer. (T can be calculated as 1 / f, where f is measured in Hertz) .

Tone bursts are defined by 4 parameters: RiseTime, DecayTime, Duration and StimDelay (averaging window offset to stimulus in ms):



The averager window offset is measured from the start of the Rise time. The Duration is defined as

Duration = RiseTime + PlateauTime + DecayTime

stimDelay	Samples offset to stimulus in millisec. Same explanation as for Click stimuli.

```
TTEOAEStimPar ::= SEQUENCE{
    stimType    TTEOAEStimType,                     -- Refer para. 2.1.5.5 on page 22
    CHOICE
        Click  {
            polarity      INTEGER,                  -- see explanation above
            clkType       INTEGER,                  -- see explanation above
            duration      INTEGER,                  -- Stimulus duration in us (microsec)
            stimDelay     INTEGER,                  -- Samples offset to stimulus in ms
        },
        ToneBurst  {
            riseTime      INTEGER,                  -- measured in us (microsec)
            decayTime     INTEGER,                  -- measured in us (microsec)
            duration      INTEGER,                  -- Stimulus duration in us (microsec)
            stimDelay     INTEGER,                  -- Samples offset to stimulus in ms
        }
}
```

2.1.5.5 TTEOAEStimType

Transient Evoked Oto Acustical Emission (OAE) Stimulus Type

TTEOAEStimType

oaest_Click	Click stimulus
oaest_ToneBurst	Tone Burst stimulus

```
TTEOAEStimType ::= INTEGER {
    oaest_Click                1,
    oaest_ToneBurst            2
}
```

2.1.6 Distortion product (DP) OAE data

Distortion product (DP) OAE data

DP Diagram and DP Input / Output curve

Distortion Product OAE data are presented as two different measurement types:

1) A so-called DP Diagram with amplitude spectra and
2) Input/output curves (IO-Curve).

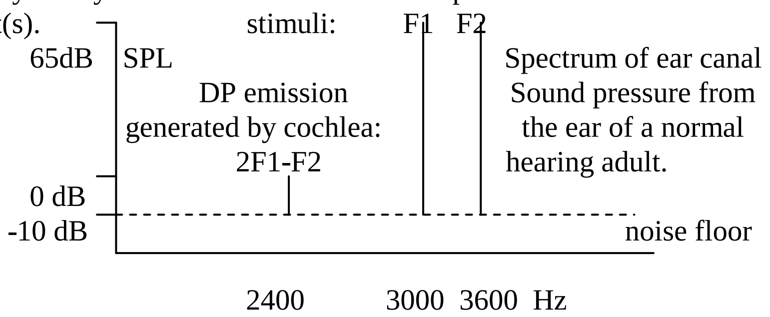
Each IO-Curve is measured at a specific frequency and saved in the TDPIOCurve structure.

Distortion Product definition [HOCA-4]

Acoustic distortion products (Acoustic DPs) result from the interaction of two simultaneously presented pure tones (the primaries). In humans, the most prominent distortion product is the cubic difference tone. Specifically, if two tones of frequencies F1 and F2 (F2 > F1) are presented externally, a third tone of frequency (2F1-F2) will be produced internally.

Acoustic DPs are technologically the easiest types of emissions to measure, being relatively artefact free and requiring no post processing. Two separate channels of signal generation, attenuation and transduction are required for the primary tones. The eliciting tones are presented to the ear through a probe microphone assembly similar to those used in measuring other types of emission except that there are two stimulus delivery ports. (..)

The ear canal sound pressure is averaged to reduce the noise floor and spectrally analysed for the levels of the primaries and the distortion product(s).



Measuring Format:

DataTypeCodes used for Distortion Product (DP) Measurements:
Side of tested person:

Left	*Right*
dtc_DPGRAMOAE_L (13)	dtc_DPGRAMOAE_R (14)
dtc_DPIOOAE_L (25)	dtc_DPIOOAE_L (26)

The following common DP values are superseded by the above two:

dtc_DPOAE_L (13)	dtc_DPOAE_R (14)

2.1.7 Distortion Product Diagram

2.1.7.1 TDPGramData

Distortion product (DP-Gram) OAE data

TDPGramData	Distortion product DP-Gram OAE data. DP-Gram data consists of up to MaxMeasNo (6) DP-grams. Each DP-Gram consists of DPGramNPoint (9) points with amplitude spectrums.
Measuring Format:	DataTypeCode =dtc_DPGRAMOAE_L (13) or dtc_DPGRAMOAE_R (14) These #defines supersedes dtc_DPOAE_L (13) and dtc_DPOAE_R (14) !

TDPGramData ::= SET OF maxMeasNo TDPGram

2.1.7.2 TDPGram

Distortion product DP-Gram

TDPGram	Distortion product DP-Gram OAE data. DP-Gram data consists of up to MaxMeasNo (6) DP-grams. Each DP-Gram consists of DPGramNPoint (9) points with amplitude spectrums.
maskSignal	Masking signal type applied to the other ear (contra lateral ear) . Refer para.2.1.10.2: TMaskSignal on page 30.
maskFreq	Frequency of the masking signal applied.
maskLevel	Level of the masking signal applied.
norm	Up to 32 characters are reserved in the structure to save the name of the norm applied to the DP-Gram curve recording.
point	The saved DP-Gram curve consists of up to dPGramNPoint (9) curvepoints. Refer para.2.1.9.1: TDPPoint on page 27 and para. 2.1.7.3: dPGramNPoint on page 25.

```
TDPGram ::= SEQUENCE {
    maskSignal            TMaskSignal,                  -- Masking signal type applied
    maskFreq              THertz,                       -- Masking signal frequency
    maskLevel             TdB10,                        -- Masking Signal Level
    norm                  TDPNormName,                  -- Norm Name in ASCII characters
    point SEQUENCE OF dPGramNPoint TDPPoint             -- The saved DPGram in 9 points
}
```

2.1.7.3 dPGramNPoint

Distortion product –Diagram: Number of measuring points	
dPGramNPoint	Number of measuring points in a Distortion Product diagram.

dPGramNPoint INTEGER ::= 9

2.1.7.4 TDPNormName

Distortion product Norm Name	
TDPNormName	Name of the norm used when recording and saving the DPGram.

TDPNormName ::= SEQUENCE OF 32 CHARACTER STRING[1]

2.1.8 Distortion Product Input-Output Curve

2.1.8.1 TDPIOData

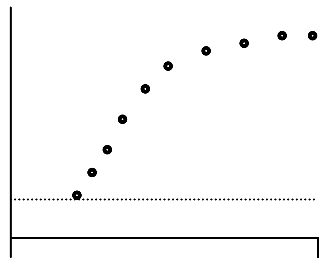
Distortion product (DP) OAE data

TDPIOData — Distortion Product Input Output (DP-I/O) data consists of up to maxMeasNo (6) input/output curves of dPIONPoint (10) points each. Each IO-Curve is measured at the specific frequencies f1 and f2. The frequencies and other relevant measuring conditions are saved in the TDPIOCurve structure.

Definition of a point in the IO-curve. (Emission level against stimulus level)

Example of an Input output curve for a F2 / F1 ratio of 1.1

10 dB SPL



-5 dB

Noise Floor

0 70 dB SPL Stimulus Level

The curve consists of 10 curve points. Refer dPIONPoint para. 2.1.8.3 on page 27.

Measuring Format: DataTypeCode =dtc_DPIOOAE_L (25) or dtc_DPIOOAE_R (26)

TDPIOData ::= SET OF maxMeasNo TDPIOCurve

[1] The TDPNormName is in OAEdef.h defined as a character array of length 32, refer appendix B.

2.1.8.2 TDPIOCurve

Distortion Product OAE Input / Output Curve

TDPIOCurve	The Distortion Product Input Output Curve is defined as a reference frequency with 10 IO-Curve points added. For description and examples of Acoustic Distortion Product Input Output curves, refer [HOCA-4] Chapter 29: Otoacoustic Emissions: An emerging clinical tool.
maskSignal	Masking signal type applied to the other ear (contra lateral ear) Ref. para. 2.1.10.2: TMaskSignal on page 30.
maskFreq	Frequency of the masking signal applied, measured in Hertz.
maskLevel	Level of the masking signal applied, measured in centibel.
norm	Up to 32 characters are reserved in the structure to save the name of the norm applied at the DP-IO curve recording.
freq	The Reference Frequency is typically defined as SQRT(F1*F2) or F1 or F2. Look up Reference Frequency in Appendix A: Dictionary and Vocabulary.
nPoint	Number of points in the saved IO-curve. The structure restricts the number to max. 10.
f1StartLevel f2StartLevel	The structure makes it mandatory to start at one end of the IO-curve, e.g. from high stimulus levels. The examples in [HOCA-4] suggest the same levels for the stimulus frequencies f1 and f2, but different levels are legal.
f1Inc f2Inc	Increment of f1, f2. Since the Distortion Product Input-Output Curve often is measured starting from higher levels with a gradual decrease of the tones f1 and f2, the sign if the increment is often negative. This means in effect a *decrement* of the level. If the interval [70 .. 25] dB SPL is to be covered, suggested common levels for f1 and f2 could be as shown below, f1Inc = f2Inc = - 50 centibel (- 5 dB):

0	1	2	3	4	5	6	7	8	9
70	65	60	55	50	45	40	35	30	25

point	The IO-curve consists of up to dPIONPoint (10) curve points.

```
TDPIOCurve ::= SEQUENCE {
    maskSignal                TMaskSignal,          -- Masking signal type applied
    maskFreq                  THertz,               -- Masking signal frequency
    maskLevel                 TdB10,                -- Masking Level
    norm                      TDPNormName,          -- Norm name
    freq                      THertz,               -- Ref. freq, typical SQRT(F1*F2) or F2
    nPoint                    INTEGER,              -- Number of points in IO-curve
    f1StartLevel              TdB10,                -- Start level of F1
    f2StartLevel              TdB10,                -- Start level of F2
    f1Inc                     TdB10,                -- Increment of F1 (often negative)
    f2Inc                     TdB10,                -- Increment of F2 (often negative)
    point SEQUENCE OF dPIONPoint TDPPoint           -- DP IO-Curve
}
```

2.1.8.3 dPIONPoint

Distortion Product OAE IO-Curve - Number of curve points

dPIONPoint	The maximum number of curvepoints in a Distortion Product OAE IO-curve.

dPIONPoint INTEGER ::= 10

2.1.9 Common Distortion Product definitions

2.1.9.1 TDPPoint

A curve point in a Distortion Product Diagram ("DP-Gram")
- or Input Output curve ("IO-Curve")

TDPPoint — The definition of the Distortion Product curve point is now common for the two measurement types.

stimAdj

Coupler	1
Volume Corrected	2
In Situ	3

timeWindow

Rectangle	1	Bartlett	8	Parzen	15
Triangular	2	Welch	9		
Gaussian	3	Riemann	10	User1	21
Hanning	4	Cauchy	11	User2	22
Hamming	5	Chebyshev	12	User3	23
Blackman	6	Cos10Percent	13	User4	24
Kaiser	7	FlatTop	14	User5	25

f1,f2 — Stimulus frequencies saved in Hertz

selectDP — Selected Distortion Product. Higher order Distortion Products, though less dominant, have been added to the list:

selectDP	Distortion Product
0	Unknown DP
1	(2*F1 - F2)
2	(2*F2 - F1)
3	(3*F1 - F2)
4	(3*F2 - F1)
5	(3*F1 - 2*F2)
6	(3*F2 - 2*F1)

f1Level, f2Level — The measured SPL level for the stimuli in centibel or dB x 10.

dp1Level, dp2Level — Cochlea generated Distortion Product SPL level in centibel or dB x 10.

(TDPPoint continued)

dp1Phase, dp2Phase	Cochlea generated Distortion Product SPL phase in "decidegrees" or degrees x 10. (Full circle is 3600).
dp1Noise, dp2Noise	SPL Noise floors for dp1 and dp2 respectively. The more accepted measurements, the lower the noise floors.
accMeas	Number of accepted measurements, i.e. measurements where the cross correlation coefficient towards the averaged sum of previous samples was adequately high. Another mechanism is rejection if artefact noise is detected. This can be done by monitoring the Sound Pressure Level of the OAE.
rejMeas	Number of rejected measurements. The dominant reason for reject is noise induced by the person under test. (Muscle activity by movement etc.)
nrLevel	Noise rejection level measured in centibel or dB x 10.
minFreq	The minimum frequency specifies the frequency of index 0 in the "sample" array representing a frequency curve.
maxFreq	The maximum frequency specifies the frequency of index (validSamples - 1) in the array.
validSamples	Number of valid samples placed in the SEQUENCE OF dNSamples TdB10. Up to dPNSamples (512) samples can be saved as curve points. The actual number of valid samples may be lower. Therefore, the samples in the array are equally spaced with a frequency distance of (maxFreq - minFreq) / (validSamples – 1).
sample	The measured samples are SEQUENCE OF dPNSamples (512) TdB10 measuring points each representing a Sound Pressure Level measured in centibel (dB x 10).

```
TDPPoint ::= SEQUENCE{
    stimAdj             INTEGER,              -- See parameter use above
    timeWindow          TTimeWindow,          -- Refer Para. 2.1.10.3 page 31
    f1                  THertz,               -- Input freq 1
    f2                  THertz,               -- Input freq 2
    f1Level             TdB10,                -- SPL Level for F1
    f2Level             TdB10,                -- SPL Level for F2
    selectDP            INTEGER,              -- See parameter use above
    dp1Level            TdB10,                -- Output DP1 level
    dp1Phase            TDg3600,              -- Output DP1 phase
    dp1Noise            TdB10,                -- SPL noise floor for DP1
    dp2Level            TdB10,                -- Output DP2 level
    dp2Phase            TDg3600,              -- Output DP2 phase
    dp2Noise            TdB10,                -- SPL noise floor for DP2
    accMeas             INTEGER,              -- Accepted measurements
    rejMeas             INTEGER,              -- Rejected measurements
    nrLevel             TdB10,                -- Noise rejection level
    minFreq             THertz,               -- Freq corresponding to first sample
    maxFreq             THertz,               -- Freq corresponding to last sample
    validSamples        INTEGER,              -- Number of valid samples
    sample SEQUENCE  OF dPNSamples TdB10      -- Amplitudes in dB SPL
}
```

2.1.9.2 dPNSamples

Distortion Product OAE IO-Curve - Number of curve points

dPNSamples	The maximum number of curve points in a Distortion Product OAE Measurement curve – The number is common for DP-grams and DP IO-curves.

dPNSamples INTEGER ::= 512

2.1.10 Oto Acoustic Emissions: Common definitions

2.1.10.1 maxMeasNo

The maximum number of measurements of each type

maxMeasNo	The maximum number of measurements (6). The number goes for Spontaneous OAE Measurement, Transient Evoked Measurements, Distortion Product Diagrams and Distortion Product IO-curves.

maxMeasNo INTEGER ::= 6

2.1.10.2 TMaskSignal

Masking Signal

Spontaneous OAE Measurement, Transient Evoked Measurements, Distortion Product DP-grams and Distortion Product IO-curves can all be measured while applying a masking signal in the patient’s opposite (contra lateral) ear.

0	unknown	Information not available about masking
1	noSignal	Masking signal not applied
2	tone	Pure tone applied as masking signal
3	nBN	Narrow band noise applied as masking signal
4	wN	White noise
5	pN	Pink noise.

```
TMaskSignal ::= INTEGER {
    unknown        0,        -- Information not available about masking
    noSignal       1,        -- Masking signal not applied
    tone           2,        -- Pure tone applied as masking signal
    nBN            3,        -- Narrow band noise applied as masking signal
    wN             4,        -- White noise
    pN             5         -- Pink noise
}
```

2.1.10.3 TtimeWindow

Time windows for Amplitude spectrums

TTimeWindow

By "looking through a window", i.e. using a weighting function of varying length and form, a final set of data can be extracted from a signal, that in principle is stationary and indefinite. If the goal is to make an estimate of the spectrum of the stationary signal, the optimum short-term spectrum estimator is searched.

The short-term spectrum is obtained as a complex folding between the indefinite spectrum and the spectrum of the windowing function. The optimum short-term spectrum estimate is obtained when the window spectrum approximates an impulse function. Ref. "Elektronik Ståbi" 7.edition (Teknisk Forlag, 1995).

timeWindow	Rectangle	1	Bartlett	8	Parzen	15
	Triangular	2	Welch	9		
	Gaussian	3	Riemann	10	User1	21
	Hanning	4	Cauchy	11	User2	22
	Hamming	5	Chebyshev	12	User3	23
	Blackman	6	Cos10Percent	13	User4	24
	Kaiser	7	FlatTop	14	User5	25

```
TTimeWindow ::= INTEGER {
    tw_Rectangle       1,
    tw_Triangular      2,        -- Also called Bartlett
    tw_Gaussian        3,
    tw_Hanning         4 ,       -- Also called Cosine Bel (cos*cos)
    tw_Hamming         5,
    tw_Blackman        6,        -- Also called Blackman-Tuckey
    tw_Kaiser          7 ,       -- Also called Kaiser-Bessel, a=2.5
    tw_Bartlett        8,
    tw_Welch           9,
    tw_Riemann         10,
    tw_Cauchy          11,
    tw_Chebyshev       12,
    tw_Cos10Percent    13,
    tw_FlatTop         14,
    tw_Parzen          15,
    tw_User1           21,       -- Depends on manufacturer codes
    tw_User2           22,       -- do.
    tw_User3           23,       -- do.
    tw_User4           24,       -- do.
    tw_User5           25        -- do.
}
```

2.1.10.4 TDg3600

Degrees	Degrees x 10 or tenths of a degree

TDg3600 ::= INTEGER

END -- of OAEdef definitions

2.1.11 Imported definitions from NOAHdef.h

-- Definition of Measuring Point
-- Import from NOAHdef.h

BEGIN

2.1.11.1 THertz

Frequency	Frequencies saved in Hertz.

THertz ::= INTEGER

2.1.11.2 TdB10

Sound Pressure Level	Sound pressure saved in Decibel x 10, i.e. saved in "centibel".

TdB10 ::= INTEGER

END – of definitions imported from NOAHdef.h

2.1.12 Reading and writing curve points

OAEDEF.H defines the following curves:

OAE curves

Curve Identifier	*Curve Type*	*Curve Point Type*	*Curve X value*	*Curve Y value*
sample	TProbeFitCurve Ref. 2.1.3.1 p. 11	float	Time calculated as sample period (ms) x n	Sound Pressure Amplitude in µPa (10^{-6} pascal)
sample	TProbeMicCurve Ref. 2.1.3.3 p.13	TdB10	Frequency calculated as $\frac{(maxFreq - minFreq)\ Hz}{(validSamples - 1)}$ x n	Sound Pressure Level (centibel re 20 µPa)
sample	TSOAECurve Ref. 2.1.4.2 p. 14	TdB10	Frequency calculated as $\frac{(maxFreq - minFreq)\ Hz}{(validSamples - 1)}$ x n	Sound Pressure Level (centibel re 20 µPa)
sampleA, sampleB	TTEOAECurve Ref. 2.1.5.2 p. 17	float	Time calculated as sample rate (ms) x n	Sound Pressure in (µPa) (10^{-6} pascal)
point	TDPGram Ref. 2.1.7.2 p. 24	TDPPoint	Frequency calculated as $\frac{(maxFreq - minFreq)\ Hz}{(validSamples - 1)}$ x n	Sound Pressure Level Emission and artifact stimulus tones (centibel re 20 µPa)
point	TDPIOCurve Ref. 2.1.8.2 p. 26	TDPPoint	Stimulus Level i.e. Sound Pressure Level (centibel re 20 µPa)	Emission Level i.e. Sound Pressure Level (centibel re 20 µPa)
sample	TDPPoint Ref. 2.1.9.1 p. 27	TdB10	Frequency calculated as $\frac{(maxFreq - minFreq)\ Hz}{(validSamples - 1)}$ x n	Sound Pressure Level (centibel re 20 µPa)

The reading of curve points in an OAE measurement from NOAH ver 2.0 is per definition done in a different way than other NOAH modules:

The "y-value", whether Sound Pressure Amplitude or Sound Pressure Level is read from the relevant structure. The curve points are defined as SEQUENCE OF which means that they are ordered with respect to x-value. Read the curve points while checking that the calculated, corresponding x-value belongs to the correct range.

For "x-values" defined as frequencies:
Check that the curvepoints [0..(validSamples-1)] indeed contain valid "y-value" samples by comparing them to a definition interval chosen for your application, refer the paragraphs 2.2.3.1: Probe Fitting Measuring Conditions on page 41 to paragraph 2.2.3.6: DP Gram Measuring Conditions on page 45.

Continue reading curve points until the namedValue endCurve occurs. This should happen in curvePoint[validSamples]:

2.1.12.1 EndCurve

-- When writing the end of the curves TProbeMicCurve, TSOAECurve and TDPPoint
endCurve dB10 ::= undefInt – See para. 2.1.1 on page 10
-- or
endCurve FLOAT ::= undefFloat -- 0.0 : Sign = 0, Exponent = 0 and Mantissa = 0

When writing curve points, you must place the curvepoints starting in curvepoint [0] in the array according to the lowest “x-value” and then place “y-values” according to ascending “x-values”.

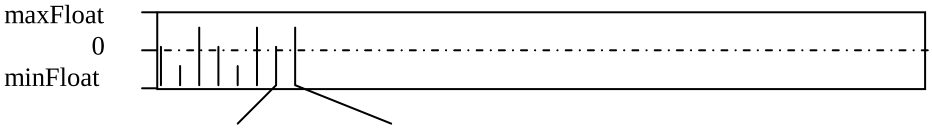
You will end with an endCurve marker and fill the rest of the array with endCurve markers (undefInts or undefFloats) if you have not used all the places in the array. This filling is not mandatory but is considerate to fellow programmers.

Valid code points should placed together. "Holes" in curves are not allowed.

2.1.12.2 OAE Curves with measurement in time

TProbeFitCurve and TTEOAECurve

- Defined SEQUENCE i.e. an ordered collection of data separated by the sample rate in ms.



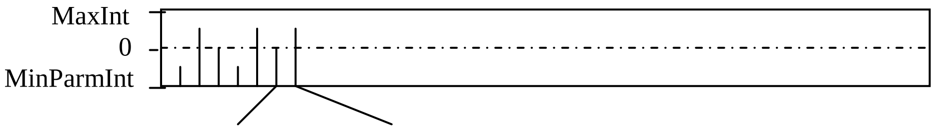
The sample period = The time between samples is saved in milliseconds

Curve Id	*Curve Type*	*Type of curve points*	*X value*	*Y value*
sample	TProbeFitCurve Ref. 2.1.3.1 p. 11	SEQUENCE OF fittingNSamples (128) FLOAT	Time calculated as n * samplePeriod	Sound Pressure (µPa)
sample	TTEOAECurve Ref. 2.1.5.2 p. 17	SEQUENCE OF tEOAENSamples (512) FLOAT	Time calculated as n * samplePeriod	Sound Pressure (µPa)

2.1.12.3 OAE curves with measurement in frequency or Stimulus Level

TProbeMicCurve, TSOAECurve, TDPGram and TDPIOCurve

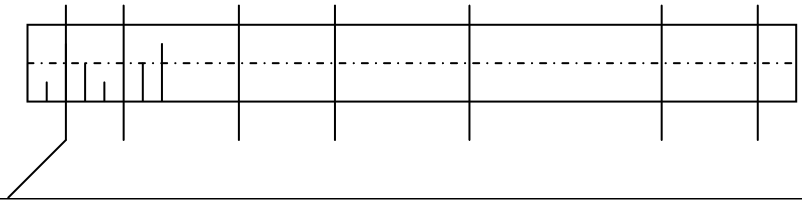
Defined SEQUENCE i.e. an ordered collection of data separated by the sample rate in ms.



The Frequency distance is (maxFreq – minFreq) / (validSamples-1)

Curve Id	*Curve Type*	*Type of curve points*	*X value*	*Y value*
sample	TProbeMic-Curve	SEQUENCE OF probeMicNSamples (1024) TdB10	frequency	Sound Pressure Level (centibel)
sample	TSOAECurve	SEQUENCE OF sOAENSamples (1024) TdB10	frequency	Sound Pressure Level (centibel)
point	TDPGram	SEQUENCE OF dPNSamples (512) TDPPoint	Frequency calculated as $\frac{(maxFreq - minFreq)\ Hz}{(validSamples - 1)} \times n$	Sound Pressure Level: Emission and artifact (centibel re 20 µPa)
point	TDPIOCurve	SEQUENCE OF dPIONPoint (10) TDPPoint	Stimulus Level	Emission Level

Inside the TSOAECurve and the TTEOAECurve structures some special values can be saved:



TSOAECurve marked indexes and TTEOAECurve qualifiers

- Defined SEQUENCE i.e. an ordered collection of data although the use is not yet specified

TSOAECurve: 10 frequencies can be saved normally marking local extremes (minima or maxima).

TTEOAECurve: The 4 qualifiers still need to be defined for use across modules from different manufacturers. One practical use of the first qualifier could be the cross correlation between the A and B curve. The second qualifier could be S/N ratio, calculated between (A+B)/2 (response) and (A-B)/2 (noise) .

Identifier	*Component of Type*	*Type*	*X value*	*Y value*
markIdx	TSOAECurve Ref. 2.1.4.2 p.14	SEQUENCE OF 10 INTEGER	frequency	(no standard use)
qualifier	TTEOAECurve Ref. 2.1.5.2 p. 17	SEQUENCE OF 4 FLOAT	floating value, manufacturer dependent	(no standard use)

The reading of curve points in a standard from NOAH ver 2.0 is normally done in the way described above, but in the OAEdef.h case it is the reading of the shown SETs that ought be treated with some extra care:

Structures defined SET OF might be ordered, but since they are defined as a set, they also might be *unordered* with respect to x-value. Read the SET OF points while checking that the x-value belongs to the correct range chosen for your application.

Structures defined SET OF are read until the namedValue endCurve occurs (see this value above).

2.2 Reading and writing OAE Measurements

In the previous chapter, the OAEdef structures were explained. This chapter will give some hints to the actual reading and writing of the four different types of OAE structures plus the Microphone Probe Fitting curve. These five structures are defined in the NOAH standard version 2.0, and they are named Probe Fitting Curve (TProbeFitCurve), Spontaneous OAE Data (TSOAEData), Transient Evoked (TTEOAEData), Distortion Product Diagram (TDPOAEData) and Distortion Product Input/Output Curve (TDPIOData) respectively.

The basic principle in other NOAH standards is that a whole structure has to be saved although perhaps only one measurement has actually been performed, but with OAE the case is different. Different Data Type Codes have been assigned for the five different kinds of measurements, refer para. 1.1: A few words about programming with OAEDEF.H on page 5.

2.2.1 Reading the OAE Measurements

The NOAH ver. 2.0 specification attaches a comprehensive measurement condition structure to each recorded curve called Measuring Conditions. In OAEDEF this has not been implemented yet, but the following structures contain the parameters that would normally be considered as measurement conditions:

Type of OAE data:

Probe Fitting	*Spontaneous*	*Transient Evoked*	*Distortion Product*
TProbeFitCurve Ref. 2.1.3.1 on page 11.	TSOAEData Ref. 2.1.4.1 on page 14.	TTEOAEData Ref. 2.1.5.1 on page 16.	TDPGramData Ref. 2.1.7.1 on page 24.
TProbeMicCurve Ref. 2.1.3.3 on page 13.		TTEOAEStimPar Ref. 2.1.5.4 on page 20.	TDPIOData Ref. 2.1.8.1 on page 25.
			TDPPoint Ref. 2.1.9.1 on page 27.

Initial Conditions:

probeFitInitialCond Ref. 2.2.2.1 on page 37.	sOAECurveInitialCond Ref. 2.2.2.3 on page 38.	tEOAECurveInitialCond Ref. 2.2.2.4 on page 38.	dPGramDataInitialCond Ref. 2.2.2.6 on page 39.
probeMicInitialCond Ref. 2.2.2.2 on page 37.		tEOAEStimParInitialCond Ref. 2.2.2.5 on page 39.	dPIODataInitialCond Ref. 2.2.2.7 on page 39.
			dPPointInitialCond Ref. 2.2.2.8 on page 40.

In this chapter some Initial Conditions are introduced in the form of namedValues[3], see the table above. If a measurement structure has not been used, it will be equal to one of these namedValues. Subsequent chapters describe the minimum changes in these measuring conditions that make them valid for each of the measurements that constitute a single OAE Measurement.

Note : The definitions for the Integer values used in OAEdef.h are written in para. 2.1.1: The Integer type used in OAEDEF.H. However, the value zero can be found in empty measurements where the correct value should have been undefInt

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

2.2.2 Writing the OAE Measurements

When writing an OAE Measurement, use the following method:

1) Initialise all the measurements in the structure by setting all Measuring Conditions to the initial conditions (see para. 2.2.3.6 : DP Gram Measuring Conditions or TEOAE Measuring Conditions:). The codepoints should be initialised with endCurve Refer to the paragraph 2.1.12 : Reading and writing curve points.

2) Insert the appropriate values in the actual Measuring Conditions for the measurements 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 measurement.

The curve points are then inserted. Their insertion follows the directions mentioned in the paragraph 2.1.12 : Reading and writing curve points.

2.2.2.1 probeFitInitialCond

Initial Measurement Conditions: The namedValue probeFitInitialCond ***Ref. 2.1.3.1: TProbeFitCurve on page 11.***			
Field	*Type*	*Value*	*Initial Hex value*
timeCurvesCorr	BOOLEAN	FALSE	#0000
probeMic	TprobeMicCurve	(see below)	(se initial conditions below)
level	TdB10::= INTEGER	undefInt	#8001
accMeas	INTEGER	undefInt	#8001
rejMeas	INTEGER	undefInt	#8001
samplePeriod	FLOAT	0.0	#0000 0000
sample	SEQUENCE OF 128 FLOAT	128 x 0.0	128 x #00 00 00 00

2.2.2.2 probeMicInitialCond

Initial Measurement Conditions: The namedValue probeMicInitialCond ***Ref. 2.1.3.3: TProbeMicCurve on page 13.***			
Field	*Type*	*Value*	*Initial Hex value*
minFreq	THertz::= INTEGER	undefInt	#8001
maxFreq	THertz::= INTEGER	undefInt	#8001
validSamples	TdB10 ::= INTEGER	undefInt	#8001
sample	SEQUENCE OF 1024 TdB10	1024 x undefInt	1024 x #8001

2.2.2.3 soaeCurveInitialCond

Initial Measurement Conditions: The namedValue soaeCurveInitialCond *Ref. 2.1.4.2: TSOAECurve on page 14.*			
Field	*Type*	*Value*	*Initial Hex value*
maskSignal	TMaskSignal ::= INTEGER	undefInt	#8001
maskFreq	THertz::=INTEGER	undefInt	#8001
maskLevel	TdB10::=INTEGER	undefInt	#8001
accMeas	INTEGER	undefInt	#8001
rejMeas	INTEGER	undefInt	#8001
nrLevel	TdB10 ::= INTEGER	undefInt	#8001
minFreq	THertz::=INTEGER	undefInt	#8001
maxFreq	THertz::=INTEGER	undefInt	#8001
validSamples	INTEGER	undefInt	#8001
sample	SEQUENCE OF 1024 TdB10	1024 x 2 undefInt	1024 x #8001
markIdx	SEQUENCE OF 10 INTEGER	10 x undefInt	10 x #8001

2.2.2.4 teoaeCurveInitialCond

Initial Measurement Conditions: The namedValue teoaeCurveInitialCond *Ref. Para. 2.1.5.2: TTEOAECurve on page 17*			
Field	*Type*	*Value*	*Initial Hex Value*
maskSignal	TMaskSignal::= INTEGER	undefInt	#8001
maskFreq	THertz::=INTEGER	undefInt	#8001
maskLevel	TdB10::=INTEGER	undefInt	#8001
stimPar	TTEOAEStimPar: See initialCond above	(see above)	(see above)
stimLevel	TdB10 ::= INTEGER	undefInt	#8001
stimAdj	INTEGER	undefInt	#8001
stimSuppress	FLOAT	0.0	#00 00 00 00
linAquisMode	BOOLEAN	FALSE	#0000
accMeas	INTEGER	undefInt	#8001
rejMeas	INTEGER	undefInt	#8001
nrLevel	TdB10 ::= INTEGER	undefInt	#8001
samplePeriod	WORD	NULL	#0000
sampleA	SEQUENCE OF 512 FLOAT	512 x 0.0	512 x #00 00 00 00
sampleB	SEQUENCE OF 512 FLOAT	512 x 0.0	512 x #00 00 00 00
qualifier	SEQUENCE OF 4 FLOAT	4 x 0.0	4 x #00 00 00 00

2.2.2.5 tEOAEStimParInitialCond

Initial Measurement Conditions: teoaeStimParInitialCond
Ref. Para. 2.1.5.4: TTEOAEStimPar on page 20.

Field	*Type*	*Value*	*Initial Hex value*
Click:			
polarity	INTEGER	undefInt	#8001
clktype	INTEGER	undefInt	#8001
duration	INTEGER	undefInt	#8001
stimdelay	INTEGER	undefInt	#8001
Tone Burst:			
riseTime	INTEGER	undefInt	#8001
decayTime	INTEGER	undefInt	#8001
duration	INTEGER	undefInt	#8001
stimDelay	INTEGER	undefInt	#8001

2.2.2.6 dPGramInitialCond

Initial Measurement Conditions: The namedValue dPGramInitialCond
Ref. Para. 2.1.7.2: TDPGram on page 24.

Field	*Type*	*Value*	*Explanation*
maskSignal	TMaskSignal::= INTEGER	undefInt	#8001
maskFreq	THertz::=INTEGER	undefInt	#8001
maskLevel	TdB10::=INTEGER	undefInt	#8001
norm	TDPNormName ::= SEQUENCE OF 32 CHARACTER STRING	31 x ASCII SPACE + NULL	31 x #20 + #00 (zero terminated string of ASCII blanks)
point	SEQUENCE OF 9 TDPPoint	(See above)	(See above)

2.2.2.7 dPIOCurveInitialCond

Initial Measurement Conditions: The namedValue dPIOCurveInitialCond
Ref. Para. 2.1.8.2: TDPIOCurve on page 26.

Field	*Type*	*Value*	*Explanation*
maskSignal	TMaskSignal::= INTEGER	undefInt	#8001
maskFreq	THertz::=INTEGER	undefInt	#8001
maskLevel	TdB10::=INTEGER	undefInt	#8001
norm	TDPNormName ::= SEQUENCE OF 32 CHARACTER STRING	31 x ASCII SPACE + NULL	31 x #20 + #00 (zero terminated string of ASCII blanks)
freq	THertz::= INTEGER	undefInt	#8001
nPoint	INTEGER	undefInt	#8001
f1StartLevel	TdB10 ::= INTEGER	undefInt	#8001
f2StartLevel	TdB10 ::= INTEGER	undefInt	#8001
f1Inc	TdB10 ::= INTEGER	undefInt	#8001
f2Inc	TdB10 ::= INTEGER	undefInt	#8001
point	SEQUENCE OF 10 TDPPoint	(see above)	(see above)

2.2.2.8 dPPointInitialCond

Initial Measurement Conditions: The namedValue dpPointInitialCond

Ref. Para. 2.1.9.1: TDPPoint, on page 27.

Field	*Type*	*Value*	*Initial Hex value*
stimAdj	INTEGER	undefInt	#8001
timeWindow	TTimeWindow ::= INTEGER	undefInt	#8001
f1	THertz ::= INTEGER	undefInt	#8001
f2	THertz ::= INTEGER	undefInt	#8001
f1Level	TdB10 ::= INTEGER	undefInt	#8001
f2Level	TdB10 ::= INTEGER	undefInt	#8001
selectDP	INTEGER	undefInt	#8001
dp1Level	TdB10 ::= INTEGER	undefInt	#8001
dp1Phase	Tdg3600::= INTEGER	undefInt	#8001
dp1Noise	TdB10 ::= INTEGER	undefInt	#8001
dp2Level	TdB10 ::= INTEGER	undefInt	#8001
dp2Phase	Tdg3600::= INTEGER	undefInt	#8001
dp2Noise	TdB10 ::= INTEGER	undefInt	#8001
accMeas	INTEGER	undefInt	#8001
rejMeas	INTEGER	undefInt	#8001
nrLevel	TdB10 ::= INTEGER	undefInt	#8001
minFreq	THertz::= INTEGER	undefInt	#8001
maxFreq	THertz::= INTEGER	undefInt	#8001
validSamples	INTEGER	undefInt	#8001
sample	SEQUENCE OF 512 TdB10	512 x undefInt	512 #8001

2.2.3 Minimum Settings when using an OAE structure

If a structure is left unused, the structure should be filled as shown above in the chapter Initial Measurement Conditions. When a structure is used for saving a measurement, a minimum set of measurement conditions and data should be saved. In this chapter will be listed the minimum settings, it is necessary to save.

2.2.3.1 Probe Fitting Measuring Conditions

Probe Fitting Measuring Conditions
Minimum settings for TProbeFitCurve, Ref. Para. 2.1.3.1 on page 11.

Field	*Type*	*Value*	*Explanation*
timeCurvesCorr	BOOLEAN	[FALSE, TRUE]	mandatory.
probeMic	TprobeMicCurve	(see below)	(see below)
level	TdB10	[-200..1200] centibel	Mandatory.
accMeas	INTEGER	[0..maxInt]	Mandatory. A positive value indicates that a measurement is present.
rejMeas	INTEGER	[0..maxInt]	Not mandatory. If not used, undefInt.
samplePeriod	FLOAT	Refer definition area for float, appendix C	Mandatory.
sample	SEQUENCE OF fittingNSamples (128) float	Refer definition area for float, appendix C	Mandatory.

2.2.3.2 Probe Microphone Curve Measuring Conditions

Probe Microphone Curve Measuring Conditions
Minimum settings for TProbeMicCurve, Ref. Para. 2.1.3.3 on page 13

Field	*Type*	*Value*	*Explanation*
minFreq	THertz::= INTEGER	[0..20 000]	Mandatory.
maxFreq	THertz::= INTEGER	[0..20 000]	Mandatory.
validSamples	INTEGER	[0..probeMicNSamples] i.e. [0..1024]	Mandatory.
sample	SEQUENCE OF probeMicNSamples (1024) TdB10	centibel: [-200..1200]	Mandatory.

2.2.3.3 SOAE Measuring Conditions

SOAE Measuring Conditions, minimum settings for TSOAECurve
Ref. Para. 2.1.4.2 on page 14.

Field	*Type*	*Value*	*Explanation*
maskSignal	TMaskSignal::= INTEGER	[0..5]	Not mandatory. Set to 1 if masking is not used.
maskFreq	THertz::=INTEGER	[0..20 000] Hz	Not mandatory. Set to undefInt if masking is not used.
maskLevel	TdB10::=INTEGER	[-200..1200] centibel or tenths of dB	Not mandatory. Set to undefInt if masking is not used.
accMeas	INTEGER	[0..maxInt]	Mandatory. A positive value indicates that a measurement is present.
rejMeas	INTEGER	[0..maxInt]	Not mandatory. If not used, undefInt.
nrLevel	TdB10 ::= INTEGER	[-200..1200] centibel	Mandatory.
minFreq	THertz::=INTEGER	[0..20 000] Hz	Mandatory.
maxFreq	THertz::=INTEGER	[0..20 000] Hz	Mandatory.
validSamples	INTEGER	[0..sOAENSamples] i.e. [0..1024]	Mandatory.
sample	SEQUENCE OF sOAENSamples (1024) TdB10	[-200..1200] centibel	If the samples are not used, insert undefInt.
markIdx	SEQUENCE OF 10 INTEGER	[0..1024]	Not mandatory. if not used, undefInt.

2.2.3.4 TEOAE Measuring Conditions

TEOAE Measuring Conditions
Minimum settings for TTEOAECurve. Ref. Para. 2.1.5.2 on page 17.

Field	*Type*	*Value*	*Explanation*
maskSignal	TMaskSignal::= INTEGER	[0..5]	Not mandatory. Set to 1 if masking is not used.
maskFreq	THertz::=INTEGER	[0..20 000] Hz	Not mandatory. Set to undefInt if masking is not used.
maskLevel	TdB10::=INTEGER	[-200..1200] centibel or tenths of dB	Not mandatory. Set to undefInt if masking is not used.
stimPar	(see TTEOAEStimPar above)	(see above)	(see table above)
stimLevel	TdB10 ::= INTEGER	[-200..1200] centibel or tenths of dB	Mandatory
stimAdj	INTEGER	[0..3]	Mandatory. Set to 0 if unknown.
stimSuppress	FLOAT	refer definition area for float, appendix C	Mandatory.
linAquisMode	BOOLEAN	[FALSE, TRUE]	Mandatory.
accMeas	INTEGER	[0..maxInt]	Mandatory. A positive value indicates that a measurement is present.
rejMeas	INTEGER	[0..maxInt]	Not mandatory. If not used, undefInt.
nrLevel	TdB10 ::= INTEGER	[-200..1200]centibel or tenths of dB	Mandatory.
samplePeriod	WORD	[0..65535] µs (microseconds)	Mandatory.
sampleA	SEQUENCE OF tEOAENSamples (512) INTEGER	[minParmInt .. maxInt] µPa (micropascal)	Unused samples are filled with undefInt.
sampleB	SEQUENCE OF tEOAENSamples (512) INTEGER	[minParmInt .. maxInt] µPa (micropascal)	Unused samples are filled with undefInt.
qualifier	SEQUENCE OF 4 FLOAT	Refer definition area for floats, appendix C	Not mandatory. If unused fill with 0 (#00 00 00 00)

2.2.3.5 TEOAE Measuring Conditions: Stimulus Parameters

TEOAE Measuring Conditions

Minimum Settings for the structure TTEOAEStimPar, Ref. Para. 2.1.5.4 on page 20.

Field	*Type*	*Value*	*Explanation*
Click:			
polarity	INTEGER	Condensation or rarefaction: [1..2]	The default value is 1. Mandatory.
clktype	INTEGER	Half wave, Full wave or filtered click: [1..3]	Mandatory.
duration	INTEGER	[0..32767] micro s.	Not mandatory. The stimDelay field is measured as the delay between onset of stimulus and onset of averager window.
stimdelay	INTEGER	[0..32767] milli s.	Not Mandatory. The stimulus will easily show if recorded in the averager window.
Tone Burst:			
riseTime	INTEGER	[0..32767] micro s.	Default value: 0. Mandatory.
decayTime	INTEGER	[0..32767] micro s.	Default value: 0. Mandatory.
duration	INTEGER	[0..32767] micro s.	Not mandatory. The stimDelay field is measured as the delay between onset of stimulus and onset of averager window.
stimDelay	INTEGER	[0..32767] milli s.	Not Mandatory. The stimulus will easily show if recorded in the averager window.

2.2.3.6 DP Gram Measuring Conditions

DPOAE Measuring Conditions: DP Gram Measurement ***Minimum settings for TDPGram, Ref. Para. 2.1.7.2 on page 24.***			
Field	*Type*	*Value*	*Explanation*
maskSignal	TMaskSignal::= INTEGER	[0..5]	Not mandatory. Set to 1 if masking is not used.
maskFreq	THertz::=INTEGER	[0..20 000] Hz	Not mandatory. Set to undefInt if masking is not used.
maskLevel	TdB10::=INTEGER	[-200..1200] centibel or tenths of dB	Not mandatory. Set to undefInt if masking is not used.
norm	TDPNormName ::= SEQUENCE OF 32 CHARACTER STRING	31 printable ASCII characters [#20..#7F] plus a NULL termination	Not mandatory. If not used, insert 31 ASCII SP (#20) plus a NULL termination.
point	SEQUENCE OF dPNGramPoint (9) TDPPoint	(see TDPPoint)	Not mandatory. If not used use dpPointInitialCond.

2.2.3.7 Distortion Product Input-Output Curve Measuring Conditions

DPOAE Measuring Conditions: Input-Output Curve Measurement

Minimum settings for TDPIOCurve, Ref.para. 2.1.8.2 on page 26.

Field	*Type*	*Value*	*Explanation*
maskSignal	TMaskSignal::= INTEGER	[0..5]	Not mandatory. Set to 1 if masking is not used.
maskFreq	THertz::=INTEGER	[0..20 000] Hz	Not mandatory. Set to undefInt if masking is not used, or if some noise is used for masking.
maskLevel	TdB10::=INTEGER	[-200..1200] centibel or tenths of dB	Not mandatory. Set to undefInt if masking is not used.
norm	TDPNormName	32 printable ASCII characters i.e. [#20..#7F]	Not mandatory. If not used, insert ASCII SP (#20)
freq	THertz::= INTEGER	[0..20 000] Hz	Mandatory
nPoint	INTEGER	[0..10]	Mandatory. Use 0 if unknown.
f1StartLevel	TdB10 ::= INTEGER	[-200..1200] centibel or tenths of dB	Mandatory
f2StartLevel	TdB10 ::= INTEGER	[-200..1200] centibel	Mandatory
f1Inc	TdB10 ::= INTEGER	[-200..1200] centibel	Mandatory
f2Inc	TdB10 ::= INTEGER	[-200..1200] centibel	Mandatory
point	SEQUENCE OF dPIONPoint (10) TDPPoint	See TDPPoint Minimum Settings on the next page.	

2.2.3.8 DPOAE Measuring Conditions: Common Distortion Product Definitions

DPOAE Measuring Conditions: Common Distortion Product Definitions
Minimum settings for TDPPoint, Ref. Para. 2.1.9.1 on page 27.

Field	*Type*	*Value*	*Explanation*
stimAdj	INTEGER	[0..3]	Use 0 if unknown. Mandatory.
timeWindow	TTimeWindow::= INTEGER	[0..14] OR [20..24]	Use 0 if unknown. Mandatory.
f1	THertz ::= INTEGER	[0..20000] Hz	Mandatory
f2	THertz ::= INTEGER	[0..20000] Hz	Mandatory
f1Level	TdB10 ::= INTEGER	[-200..1200]centibel or tenths of dB	Mandatory.
f2Level	TdB10 ::= INTEGER	[-200..1200]centibel	Mandatory
selectDP	INTEGER	[0..6]	Use 0 if unknown. Mandatory.
dp1Level	TdB10 ::= INTEGER	[-200..1200]centibel	Mandatory
dp1Phase	TdG3600	[-3600..3600] deciDegrees	Not Mandatory. If unused, undefInt.
dp1Noise	TdB10 ::= INTEGER	[-200..1200]centibel	Mandatory
dp2Level	TdB10 ::= INTEGER	[-200..1200]centibel	Mandatory
dp2Phase	TdG3600	[-3600..3600] deciDegrees	Not Mandatory. If unused, undefInt.
dp2Noise	TdB10 ::= INTEGER	[-200..1200]centibel	Mandatory
accMeas	INTEGER	[0..maxInt]	Mandatory. A positive value indicates that a measurement is present.
rejMeas	INTEGER	[0..maxInt]	Not mandatory. If not used, undefInt.
nrLevel	TdB10 ::= INTEGER	[-200..1200]centibel	Not mandatory. Use undefInt if not used.
minFreq	THertz::=INTEGER	[0..20 000] Hz	Mandatory.
maxFreq	THertz::=INTEGER	[0..20 000] Hz	Mandatory.
validSamples	INTEGER	[0..dPNSamples] i.e. [0..512]	Mandatory.
sample	SEQUENCE OF dPNSamples (512) TdB10	[-200..1200] centibel or tenths of dB	Mandatory. Fill up unused samples with undefInt

Appendix A: Vocabulary and Abbreviations

A

ASN.1	Abstract Syntax Notation No. 1. ITU and ISO defined language for specification of protocol message content.
Accepted measurements. *See* AccMeas	Number of accepted measurements, i.e. measurements where the cross correlation coefficient towards the averaged sum of previous samples was adequately high.
AccMeas	(see explanation above)
Acquisition Mode *See* linAcquisMode *See* BOOLEAN	linAcquisMode is defined TBool: FALSE 0 Non Linear TRUE 1 Linear
Amplitude characteristic	The probe fit curve is used to check that the probe is properly inserted in the patient's ear. The probe-fitting curve consists of a time curve with 128 points. An FFT and an amplitude characteristic for the system probe - ear can be derived from the time curve. Refer para. 2.1.3.1: TProbeFitCurve on page 11.
Amplitude spectrum	An amplitude spectrum typically comes from an FFT analyser.

B

BOOLEAN

In [Framework] the Boolean type is defined as

```
TBool ::= INTEGER {
   FALSE   0,
   TRUE    1
}
```

The definition is placed in the header file NOAHdef.h which is included by OAEdef.h.

C

CHARACTER STRING	This ASN.1 builtin type is used for defining a 32 character ASCII string, see normName. Note that only printable ASCII characters are allowed, and that the field is initialised with ASCII #20 (Space). On the last position in the array is inserted a zero-termination. (#00).
CHOICE	This ASN.1 construct is used to describe the "C" union: "A union is a variable that may hold (at different times) objects of different types and sizes, with the compiler keeping track of size and alignment requirements. Unions provide a way to manipulate different kinds of data in a single area of storage. They are analogous to variant records in Pascal", Refer [ANSI-C]

clkType

1	Half Wave click	Condensation click
2	Full Wave click	Condensation-Rarefaction click
3	Filtered click	A square wave passed through a band pass filter (e.g. 1/1 or 1/3 octave). The square wave causes the filter to "ring" and the acoustic signal will be determined by the filter's impulse response. Refer [AECP] para. 3.

components
Used in ASN.1 for the fields in a structured type (a "C" structure). The components are given Identifiers, i.e. a field name, in "C" referred to as the member.

Condensation.
See polarity
The click stimulus polarity can be condensation (maximum pressure) or rarefaction (minimum pressure).

Cosine Bel (cos*cos).
See tw_Hanning
Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.

D

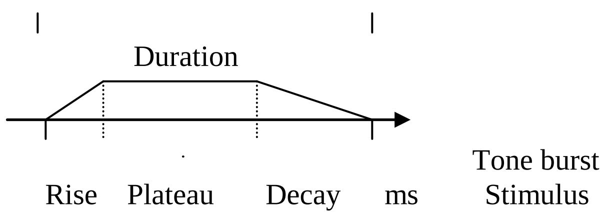
data qualifiers
See qualifiers
The qualifiers of type float still needs to be defined in order to be used across modules of different brand. One practical use could be the cross correlation demanded before a recorded frame of samples is accepted.

data structure
OAEDEF.H describes the data structure for interchange of data with the NOAH ver. 2.0 database.

DataFmtCodeStd
Data Format Code Standard, see explanation in the document preface on page 2. For this OAEdef.h, DataFmtCodeStd = 200.

DataTypeCode
Different Data Type Codes are allocated for the three different OAE methods, refer para. 1.1: A few words about programming with OAEDEF.H on page 5.

decayTime
Tone bursts are defined by 4 parameters: RiseTime, DecayTime, Duration and StimDelay (averaging window offset to stimulus in ms):



The averager window offset is measured from the start of the Rise time. The Duration is defined as

Duration = RiseTime + PlateauTime + DecayTime

DevTypeCode
Defined as Integer in Noahdef.h. Identifies a particular device or instrument type to a NOAH module. Defined individually by NOAH modules. Ref. [Framework].

Distortion product OAE data	Acoustic distortion products (Acoustic DPs) result from the interaction of two simultaneously presented puretones (the primaries). In humans, the most prominent distortion product is the cubic difference tone. Specifically, if two tones of frequencies F1 and F2 (F2 > F1) are presented externally, a third tone of frequency (2F1-F2) will be produced internally. This tone is the most prominent distortion product.
DPGram	Distortion Product Diagram.
dPGramData	Distortion product DP-Gram OAE data. DP-Gram data consists of up to MaxMeasNo (6) DP-grams. Each DP-Gram consists of DPGramNPoint (9) points with amplitude spectrums. Refer para. 2.1.7.1: TDPGramData on page 24.
DP-Gram point	Distortion Product Diagram Point. Refer para. 2.1.9.1: TDPPoint on page 27.
dPIONPoint	The maximum number of curvepoints in a Distortion Product OAE IO-curve. Refer para. 2.1.8.3: dPIONPoint on page 27.
DPLevel	Cochlea generated Distortion Product SPL level in centibel or dB x 10. Refer para. 2.1.9.1: TDPPoint on page 27.
dPNSamples	The maximum number of curve points in a Distortion Product OAE Measurement curve – The number is common for DP-grams and DP IO-curves.
DPOAE data	Distortion Product OAE data.
duration	Stimulus duration in microsec (µsec) Refer Para. 2.1.5.4: TTEOAEStimPar on page 20.

E

endCurve	The sequence of curve points in an OAE Curve is not necessarily filled with data. It is recommended to save an endCurve marker after the curve points with actual data. The unused curve points can be endCurve or null-filled. See para. 2.1.12 Reading and writing curve points on page 32.
Evoked Otoacoustic emissions	Transient Evoked Otoacoustic Emissions (TEOAEs) also referred to as click evoked OAEs are frequency dispersive responses following a brief acoustic stimulus, such as a click or tone burst. Because this was the first emission type reported in the litterature by D. T. Kemp in 1978, the term *evoked otoacoustic emissions* is often applied specifically to transient evoked emissions. They are also known as *Kemp echoes*, and *delayed evoked otoacoustic emissions*

F

f1, f2 — Supplying two stimuli of two different frequencies F1 and F2 provokes the distortion Product OAE.

f1Inc, f2Inc — Increment of f1, f2. Since the Distortion Product Input-Output Curve often is measured starting from higher levels with a gradual decrease of the tones f1 and f2, the sign if the increment is often negative. This means in effect a *decrement* of the level. If the interval [70 .. 25] dB SPL is to be covered, suggested common levels for f1 and f2 could be as shown below, f1Inc = f2Inc = - 50 centibel (- 5 dB):

0	1	2	3	4	5	6	7	8	9
70	65	60	55	50	45	40	35	30	25

f1Level, f2Level — The measured SPL level for the stimuli in centibel or dB x 10. Refer para. 2.1.9.1: TDPPoint on page 27.

f1StartLevel, f2StartLevel — The structure makes it mandatory to start at one end of the IO-curve, e.g. from low stimulus levels. The examples in [HOCA-4] suggest the same levels for the stimulus frequencies f1 and f2, but different levels are legal. Refer para. 2.1.8.2: TDPIOCurve on page 26.

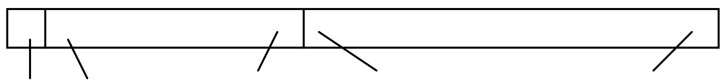
FFT analysis — Analysis by using Fast Fourier Transform: Transformation of curves, data from time to frequency domain and vice versa.

fittingNSamples — The probe fit curve is used to check that the probe is properly inserted in the patient's ear. The probe fitting curve consists of a time curve with fittingNSamples (128) samples or measuring points. An FFT and an amplitude characteristic for the system probe - ear can be derived from the time curve.

float — The floating point builtin type is defined as a "C" 32 bit Float with range [-3.4E38 .. –1.18E-38 OR 0.0 OR 1.18E-38 .. 3.4E38] and 7-digit precision. For further explanation refer appendix C.

Bit No.		***Usage***
[31]	(1 bit)	sign
[23..30]	(8 bit)	Biased exponent (incl. exponent sign)
[0..22]	(23 bit)	Significand (Normalised Mantissa)

Borland 16-bit "C": 32 bit float



S (30)-exponent-(23) (bit 22) Significand ...(bit0)

freq *see* Reference Frequency — The Reference Frequency is in OAEdef.h defined as SQRT(F1*F2) in a Distortion Product measurement

H

Half Wave click. *See* clkType	The click stimulus consists in this case of a half wave Sinusoidal.

I

Input-Output Curve	The Distortion Product Input-Output Curve. Refer para. 2.1.8: Distortion Product Input-Output Curve on page 25.
int	The "C" 16 bit integer is mapped to the ASN.1 builtin type INTEGER. Its usage is described in para. 2.1.1: The Integer type used in OAEDEF.H on page 10.

K

Kaiser-Bessel. *See* tw_Kaiser	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.

L

level	Stimulus level when recording a Probe Fitting Curve. Refer para. 2.1.3.1: TProbeFitCurve on page 11.
linAquisMode	The Linear Acquisition Mode: A Boolean value. Refer para. 2.1.5.2: TTEOAECurve on page 17.
Acquisition Mode *See* linAcquisMode *See* BOOLEAN	linAcquisMode is defined TBool: FALSE 0 Non Linear TRUE 1 Linear

M

Manufacturer codes	A code that identifies the manufacturer of a NOAH measurement or fitting module. Refer [Framework] para. 3 for TManufCode.
markIdx	The 10 Marked frequencies Index (fix points) are normally used to indicate local maxima, but the lack of a precise definition makes the use of these codes manufacturer dependant. Refer para. 2.1.4.2: TSOAECurve on page 14.
maskFreq	Frequency of the masking signal applied.
maskLevel	Level of the masking signal applied.
maskSignal	Masking signal type applied to the other ear (contra lateral ear) .

	Refer para.2.1.10.2: TMaskSignal on page 30.
maxFreq	The Maximum Frequency specifies the frequency of index ValidSamples-1 in the array. Refer para. 2.1.3.3: TProbeMicCurve on page 13.
maxInt	Highest positive value for the Integer Type = 32767 (#7FFF hex). Refer [Framework] and Para. 2.1.1: The Integer type used in OAEDEF.H on page 10.
maxMeasNo	Up to maxMeasNo (6) measurements can be saved. This goes for SOAE, TEOAE, DP-Gram and IO-Curve
measCond	Measuring Conditions. Refer to the paragraphs 2.2.3.1 through 2.2.3.6.
minFreq	The Minimum Frequency specifies the frequency of index 0 in the "Sample" array. Refer para. 2.1.3.3: TProbeMicCurve on page 13.
Minimum Settings	The recommended minimum of Measurement Conditions that must be saved with a measurement in order to make it valuable when retrieved at a later stage.
minInt	Integers are stored using 2's complement in a two-byte store. This means that minInt = -32768 or #8000 hex. Refer para. 2.1.1.

N

nBN *See* TMaskSignal	Narrow band noise applied as masking signal.
Noise rejection level. *See* nrLevel.	Noise rejection level measured in centibel or dB x 10. Refer para. 2.1.4.2: TSOAECurve on page 14 or Para. 2.1.9.1: TDPPoint on page 27.
Noise Floor	Sound Pressure Level of the Noise Floor. The more accepted measurements, the lower the noise floor. Refer Para. 2.1.9.1: TDPPoint on page 27.
norm	Up to 32 characters are reserved in the structure to save the name of the norm applied at the DP-IO curve recording. The Norm Name can be saved in the structures 2.1.7.2: TDPGram and 2.1.8.2 TDPIOCurve.
noSignal See TMaskSignal	Masking signal not applied.
nPoint	Number of points in the saved IO-curve. The structure restricts the number to max. 10. Refer. para. 2.1.8.2: TDPIOCurve on page 26.
nrLevel	Noise rejection level measured in centibel or dB x 10.

O

oaest_Click	OAE Stimulus Type = Click stimulus (1)
oaest_ToneBurst	OAE Stimulus Type = Tone Burst (2)
Output DP level. *See* DPLevel	Cochlea generated Distortion Product SPL level in centibel or dB x 10. Refer Para. 2.1.9.1: TDPPoint on page 27.
Output DP phase. *See* Phase	Cochlea generated Distortion Product SPL phase in "decidegrees" or degrees x 10. (Full circle is 3600). Refer Para. 2.1.9.1: TDPPoint on page 27.

P

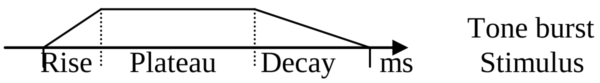
Phase	(Refer Output DP phase above.)
pN *see* TMaskSignal	Pink Noise. Refer 2.1.10.2: TMaskSignal on page 30.
point	This identifier is used in two different curves: 1) The saved DPGram consists of up to dPNSamples (512) curve points. Refer para. 2.1.7.2: TDPGram on page 24. 2) An IO-curve consists of up to dPIONPoint (10) curve points. Refer para. 2.1.8.2: TDPIOCurve on page 26.
polarity	1: Condensation (maximum pressure) 2: Rarefaction (minimum pressure)
probeMicNSamples	Up to probeMicNSamples (1024) samples can be saved as curve points. The actual number of valid samples may be lower.

Q

Qualifier	The four qualifiers of the "C" builtin type float are used for validation of data and could be correlation coefficients. Qualifiers are for manufacturer-internal purposes until a proper definition has been agreed. Refer para. 2.1.5.2: TTEOAECurve on page 17.

R

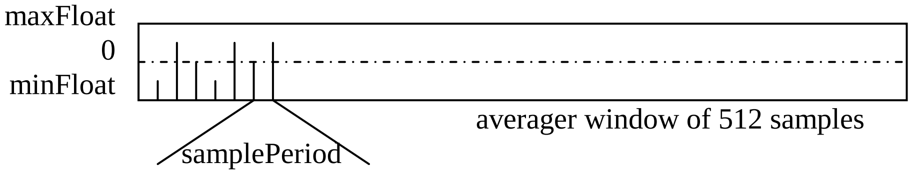
RareFaction. *See* polarity	Polarity of stimulus leads to a Minimum pressure.
Rejected measurements. *See* RejMeas	Number of rejected measurements. The dominant reason for reject is noise induced by the person under test. (Muscle activity by movement etc.) Refer Para. 2.1.9.1: TDPPoint on page 27.
Reference Frequency	The Reference Frequency is in OAEdef.h defined as SQRT(F1*F2) in a Distortion Product measurement. It can also be defined as F1 or F2. The purpose of the Reference Frequency is solely for display of a Distortion Product Measurement. Note that both F1 and F2 are accessible in the structure TDPPoint. The acoustic-electric activity at the basilar membrane takes place at several points while measuring a Distortion Product: There is activity at F1, at F2 and at their differential products. It's a matter of taste how the result of a such measurement is displayed.
rejMeas	Refer Para. 2.1.9.1: TDPPoint on page 27.
riseTime	Tone bursts are defined by 4 parameters: RiseTime, DecayTime, Duration and StimDelay (averaging window offset to stimulus in ms):



Duration = RiseTime + PlateauTime + DecayTime

S

Sample	A sample i.e. a momentary measurement of a signal value, in the OAE case a (small) measured sound pressure that converted to binary format and represented in an integer in µPa.
Sample period in µs. *See* SamplePeriod	TTEOAECurve example from OAEdef.h: - Defined SEQUENCE i.e. an ordered collection of data separated by the sample period in milliseconds (ms).



samplePeriod = The time between samples is saved in millisec.

Note that TProbeFitCurve also contains a samplePeriod. Refer para. 2.1.3.1: TProbeFitCurve on page 11.

SampleA, SampleB	512 samples of unit micropascal (µPa). Refer para. 2.1.5.2: TTEOAECurve on page 17.
SamplePeriod	The time measured between taking samples of an analogue signal is called T_s . The sampling frequency f_s is defined $f_s = \frac{1}{T_s}$ The sample rate is another term for sampling frequency.
Samples offset to stimulus in millisec. *See* stimDelay	Start of averager window (Sample array) offset to stimulus in millisec. A negative value indicates that the stimulus is ahead in time of the averager window.
SOAE data	Spontaneous Oto Acoustic Emissions. SOAE data consists of an amplitude spectrum and 10 Marked Frequencies Index to indicate responses. The structure also includes measurement conditions. The amplitude spectrum typically comes from an FFT analyser, so 1024 samples is a good number.
sOAENSamples	1024. See explanation to SOAE data above.
SPL Level for F1. *See* F1Level	Distortion Product OAE Input level. Refer Para. 2.1.9.1: TDPPoint on page 27.
SPL Level for F2. *See* F2Level	Distortion Product OAE Input level. Refer Para. 2.1.9.1: TDPPoint on page 27.
Spontaneous OAE data. *See* SOAE data	Spontaneous Oto Acoustic Emissions (SOAE) are more or less continuos narrowband signals emitted by about 50 pct. of human ears even in the absence of external acoustic stimulation. Refer para. 2.1.4.1: TSOAEData on page 14.
stimAdj	Stimulus adjustment defined INTEGER. The actual level will change from the desired level if e.g. the ear volume is not the same as when calibrating the probe in a coupler (normally 2cc). 1 The stimulus level is adjusted using Coupler 711, ref. [IEC-60711], i.e. it is not compensated for the actual acoustical conditions. 2 The stimulus level is Cavity Corrected i.e. it is adjusted to compensate for the different volume actually used at the measurement. 3 The stimulus level is In Situ Corrected, i.e. it is adjusted by using the probe microphone placed in the test persons ear so the actual level can be measured.
stimDelay	(see Samples offset above)
StimLevel	Sound pressure level of the stimulus measured in centibel or dB x 10. Refer para. 2.1.5.2: TTEOAECurve on page 17.

StimPar	The Transient Evoked OAE stimulus parameter is described in this document in para.2.1.5.4: TTEOAEStimPar on page 20.
StimSuppress	Number of milliseconds to suppress after the Stimulus onset. Refer para. 2.1.5.2: TTEOAECurve on page 17.

StimType	The ERA stimulus type is adapted for TEOAE use in OAEdef.h:
	TTEOAEStimType ::= INTEGER {
	oaest_Click 1,
	oaest_ToneBurst 2
	}

Stimulus duration in microsec (µsec). *See* duration	Stimulus duration in microsec (µsec) Refer Para. 2.1.5.4: TTEOAEStimPar on page 20.

T

TdB10	Sound Pressure Level expressed in dB x 10 or centibel.
TDevTypeCode	Device Type Code defined as Integer in noahdef.h.
TDg3600	Degrees x 10 or tenths of a degree. Refer para. 2.1.10.4: TDg3600 on page 32.
TDPGramPoint	A curve point in the Distortion Product Diagram. Refer para. 2.1.9.1: TDPPoint on page 27.
TDPOAE	Distortion Product OAE.
TEOAE	Transient Evoked Oto Acoustic Emissions.
TEOAE response curve	Transient evoked OAE - The response curve is the output from cochlea in µPa sampled in an averager window.
tEOAENSamples	A TEOAE response curve consists of TEOAENSamples (512) curve points recorded at a given sample rate. Refer para. 2.1.5.3: TEOAENSamples on page 19.
TERAStimPar	Electrical Response Audiometry Stimulus Parameter. This parameter was earlier imported to OAEdef.h from ERAdef.h. In the present edition of OAEdef.h, TTEOAEStimPar replaces it.
THertz	Frequency is measured in Hertz (cycles per second). This type is imported from NOAHdef.h.
Time	Time is measured between samples in microseconds (µs). Refer samplePeriod.

time response curves	The outer structure for transient evoked OAE called TEOAEData consists of 6 time response curves.
timeCurvesCorr	If the time curve is corrected for the microphone frequency characteristic, the boolean “TimeCurvesCorr” will indicate this by the value ‘TRUE’. Refer para. 2.1.3.1: TProbeFitCurve on page 11.
TimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
TDPIOCurve	Refer para. 2.1.8.2: TDPIOCurve on page 26.
TMaskSignal	Refer para. 2.1.10.2: TMaskSignal on page 30.
tone	A pure tone applied as masking signal. Refer para. 2.1.10.2: TMaskSignal on page 30.
toneBurst	Identifier in the TTEOAEStimPar structure. Ref para. 2.1.5.4: TTEOAEStimPar on page 20.
TProbeMicCurve	Refer para. 2.1.3.3: TProbeMicCurve on page 13.
Transient evoked OAE data	These OAE are so-called echoes of click or tone burst stimuli.
TSOAECurve	Refer para. 2.1.4.2TSOAECurve14
TSOAEData	Refer para. 2.1.4.1: TSOAEData on page 14.
TTEOAEData	Refer para. 2.1.5.1: TTEOAEData on page 16.
TTEOAECurve	Refer para. 2.1.5.2: TTEOAECurve on page 17.
TTimeWindow	Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_Blackman. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_Gaussian. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_Hamming. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_Hanning. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_Kaiser. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_Rectangle. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.

tw_Triangular. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.
tw_User[1..5]. *See* TTimeWindow	Window function. Refer para. 2.1.10.3: TtimeWindow on page 31.

U

undefInt	The Integer value -32767. (#8001 hex). Used to indicate that a value is undefined. This value is assigned to the constant undefInt. Ref. [Framework].
unknown.	The Integer value 0. (#0000 hex). When used as a parameter value it means that the parameter is **defined** however, to an unknown value.

V

validSamples	The actual size of a curve array limits the max no. of valid samples. The actual number of valid samples may be lower. Therefore, the samples in the array are equally spaced with a frequency distance of (maxFreq – minFreq) / (validSamples – 1). Refer for ex. para. 2.1.3.3: TProbeMicCurve on page 13.

W

wN	White Noise applied as masking signal Refer para. 2.1.10.2: TMaskSignal on page 30.
WORD	The "C" unsigned integer is in the ASN.1 text of this document and in the header file ERAdef.h named WORD. The legal values are [0.. 65535] , Refer [Framework].

Appendix B: The header file OAEDEF.H

```
// ======================================================================
//
// File        : OAEDef.h
//
// Project     : NOAH 2.x
//
// Version     : 1.00
//
// Purpose     : Contains global constant and type definitions for OAE
//               data (Otoacoustic emissions) to be used by HIMSA
//               modules.
//
// Copyright   : HIMSA A/S, 1999
//
// Written By  : HIMSA A/S, 1999
//
//==========================================================================
// FORMAT:     DataTypeCode  =dtc_SOAE_L or dtc_SOAE_R (9/10)
//             DataFmtCodeStd=200
// and         DataTypeCode  =dtc_TEOAE_L or dtc_TEOAE_R (11/12)
//             DataFmtCodeStd=200
// and         DataTypeCode  =dtc_DPOAE_L or dtc_DPOAE_R (13/14)
//             DataFmtCodeStd=200
// and         DataTypeCode  =dtc_DPIOOAE_L or dtc_DPIOOAE_R (25/26)
//             DataFmtCodeStd=200
// and         DataTypeCode  =dtc_PROBEFITOAE_L or dtc_PROBEFITOAE_R (27/28)
//             DataFmtCodeStd=200
//
// ==========================================================================
// Prevent multiple includes
#ifndef __OAEDEF_H
#define __OAEDEF_H

#include <windows.h>
#include "noahdef.h"         // include HIMSA defines
#include "noahadd.h"         // additional oae data type defines

// The above mentioned data formats dtc_DPIOOAE_L, dtc_DPIOOAE_R,
// dtc_PROBEFITOAE_L, dtc_PROBEFITOAE_R are defined in noahadd.h
// The data formats dtc_DPOAE_L and dtc_DPOAE_R defined in noahdef.h
// are superseeded by dtc_DPGRAMOAE_L and dtc_DPGRAMOAE_R defined in noahadd.h

// definition of max number of measurements of each type

#define MaxMeasNo 6

typedef int TMaskSignal;     // Description of signal
//-----------------------------------------------------------------------
#define NoSignal 1           // Channel without any signal
#define Tone     2           // Pure Tone
#define NBN      3           // Narrow Band Noise
#define WN       4           // White Noise
#define PN       5           // Pink Noise
//-----------------------------------------------------------------------
```

```
// If correcting spectra (TEOAE and probefit) with the microphone amplitude
// characteristic, it is necessary to save the microphone curve.
//
// MinFreq specifies the frequency of index 0 in the "Sample" array.
//
// MaxFreq specifies the frequency of index ValidSamples-1 in the array.
//
// The samples in the array are equally spaced with a frequency distance of
// (MaxFreq - MinFreq) / (ValidSamples - 1).

#define       ProbeMicNSamples            1024


typedef struct
{
  THertz      MinFreq;      // Freq corresp to first sample
  THertz      MaxFreq;      // Freq corresp to last sample
  int         ValidSamples; // Number of valid samples
  TdB10       Sample[ProbeMicNSamples]; // Amplitude in dB10 SPL
} TProbeMicCurve;


// Definition of fitting curve
// The probe fit curve is used to check that the probe is properly inserted
// in the patients ear.
//
// The probe fitting curve consists of a time curve with 128 points.
// An FFT and an amplitude characteristic for the system probe - ear can be
// derived from the time curve.
//
// The flag "TimeCurvesCorr" is TRUE if the time curve has been corrected for
// influence from probe microphone amplitude characteristic.
//
// If the flag "TimeCurvesCorr" is FALSE the system amplitude characteristic
// must be corrected for the probe microphone amplitude characteristic.
//
// If the probe fit is leaky, the amplitude characteristic will have a low level
// at the low frequencies.
//
// The record includes measurement parameters.

// ---------------------------------------------------------------------------
// FORMAT:    DataTypeCode = dtc_PROBEFITOAE_L or dtc_PROBEFITOAE_R (27/28)
//            DataFmtCodeStd = 200
// ---------------------------------------------------------------------------

#define       FittingNSamples             128


typedef struct {
  TBool           TimeCurvesCorr;
  TprobeMicCurve  ProbeMic;                // Probe mic amplitude characteristic
  TdB10           Level;                   // Stimulus level used
  int             AccMeas;                 // Accepted measurements
  int             RejMeas;                 // Rejected measurements
  float           SamplePeriod;            // Sample Period in milliseconds (ms)
  float           Sample[FittingNSamples];  // Amplitude in micropascal (uPa)
} TProbeFitCurve;
```

```
// Definition of Spontaneous OAE curve
//


// SOAE data consists of an amplitude spectrum and 10 fix point
// (frequencies) to indicate responses. The amplitude spectrum typically
// comes from an FFT analysis, so 1024 samples is a good number.
//


// MinFreq specifies the frequency of index 0 in the "Sample" array.
// MaxFreq specifies the frequency of index ValidSamples-1 in the array.
// The samples in the array are equally spaced with a frequency distance of
// (MaxFreq - MinFreq) / (ValidSamples - 1).


// -------------------------------------------------------------------------
// FORMAT:     DataTypeCode = dtc_SOAE_L or dtc_SOAE_R (9/10)
//             DataFmtCodeStd = 200
// -------------------------------------------------------------------------


#define       SOAENSamples  1024



typedef struct {
  TmaskSignal MaskSignal;   // Masking signal type applied to other ear
  THertz      MaskFreq;     // Masking signal frequency
  TdB10       MaskLevel;    // Masking level
  int         AccMeas;      // Accepted measurements
  int         RejMeas;      // Rejected measurements
  TdB10       NRLevel;      // Noise rejection level
  THertz      MinFreq;      // Freq corresp to first sample
  THertz      MaxFreq;      // Freq corresp to sample ValidSamples-1
  int         ValidSamples; // Number of valid samples
  TdB10       Sample[SOAENSamples];    // Amplitude in dB SPL
  int         MarkIdx[10];  // marked frequencies
} TSOAECurve;





// Definition of SOAE data
//

// SOAE data consists of up to MaxMeasNo measurements
//


typedef       TSOAECurve    TSOAEData[MaxMeasNo];
```

```
// Definition of stimulus type for TEOAE
//

typedef int TTEOAEStimType;
#define oaest_Click         1     // Click stimulus
#define oaest_ToneBurst     2     // Tone Burst stimulus

// Definition of Stimulus parameter record for TEOAE measurements. There are
// different sets of parameters for click stimulus and for tone burst
// stimulus.
//

typedef struct {
  TTEOAEStimType StimType;
  union U {
    struct {
      int Polarity;  // 1=Condensation, 2=RareFaction
      int ClkType;   // 1=Half Wave click,
                     // 2=Full Wave click,
                     // 3=Filtered click
      int Duration;  // Stimulus duration in usec.
      int StimDelay; // Samples offset to stimulus in msec.
    } Click;
    struct {
      int RiseTime;  // usec.
      int DecayTime; // usec.
      int Durationr; // Stimulus duration in usec.
      int StimDelay; // Samples offset to stimulus in msec.
    } ToneBurst;
  } U;
} TTEOAEStimPar;
```

```
// Definition of TEOAE curve
// Each time response curve consists of the actual samples, data
// qualifiers and some parameters describing the measurement. The
// curve represents 512 points at a given sample rate. SampleA and
// SampleB are measured alternately. A+B is the resulting curve and
// A-B is the noise. The qualifiers are used for validation of data and
// could be the correlation coefficient in the time interval 5-20 ms.
// The use of qualifiers is an alternative or supplement to the use of
// A and B buffers. The field StimLevel contains the desired level.
//
// The actual level will change from the desired level if e.g the ear volume
// is not the same as when calibrating the probe in a coupler (normally 2cc).
// If StimAdj is 1 (coupler) the stimulus level is not compensated for the
// actual acoustical conditions.
//
// If StimAdj is 2 (cavity) the stimulus level is adjusted to compensate for
// a different volume.
//
// If StimAdj is 3 (insitu) the stimulus level is adjusted by using the probe
// microphone to measure the actual level.

// --------------------------------------------------------------------------
// FORMAT:     DataTypeCode = dtc_TEOAE_L or dtc_TEOAE_R (11/12)
//             DataFmtCodeStd = 200
// --------------------------------------------------------------------------

#define       TEOAENSamples 512

typedef struct {
  TmaskSignal   MaskSignal;      // Masking signal type applied to other ear
  THertz        MaskFreq;        // Masking signal frequency
  TdB10         MaskLevel;       // Masking level
  TTEOAEStimPar StimPar;         //
  TdB10         StimLevel;       // SPL stimulus level
  int           StimAdj;         // 1=coupler, 2=cavity corrected, 3=insitu
  float         StimSuppress;    // msec. to suppress after stimulus
  TBool         LinAcquisMode;   // TRUE=Linear FALSE=Non linear
  int           AccMeas;         // Accepted measurements
  int           RejMeas;         // Rejected measurements
  TdB10         NRLevel;         // Noise rejection level
  float         SamplePeriod;    // Sample Period in milliseconds (ms)
  float         SampleA[TEOAENSamples];   // Unit:micropascal (uPa)
  float         SampleB[TEOAENSamples];   // Unit:micropascal (uPa)
  float         Qualifier[4];    // First can be correlation, second S/N-ratio
} TTEOAECurve;


// Definition of TEOAE data
// TEOAE data consists of MaxMeasNo time response curves
// The probe microphone correction curve is assumed to be the same
// for all measurements in a data set.
// The flag "TimeCurvesCorr" is TRUE if time curves has been corrected
// for influence from probe microphone characteristic.

typedef struct
{
  TBool           TimeCurvesCorr;
  TprobeMicCurve  ProbeMic;
  TTEOAECurve     Data[MaxMeasNo];
```

```
} TTEOAEData;
// Definition of time windows for Amplitude spectrums
typedef int TTimeWindow;

//--------------------------------------------------------------------------------
#define tw_Rectangle        1
#define tw_Triangular       2
#define tw_Gaussian         3
#define tw_Hanning          4              // Also called Cosine bell (cos*cos)
#define tw_Hamming          5
#define tw_Blackman         6
#define tw_Kaiser           7              // Also called Kaiser-Bessel, a=2.5
#define tw_Bartlett         8
#define tw_Welch            9
#define tw_Riemann          10
#define tw_Cauchy           11
#define tw_Chebyshev        12
#define tw_Cos10Percent     13
#define tw_FlatTop          14
#define tw_Parzen           15
#define tw_User1            21             // Depends on manufacturer codes
#define tw_User2            22             // do.
#define tw_User3            23             // do.
#define tw_User4            24             // do.
#define tw_User5            25             // do.
//--------------------------------------------------------------------------------


typedef int TDg3600;

// Definition of a DP point
// Using a 1024 point FFT will give 512 points in the frequency range
// from 0 to FSample / 2.
// The frequency resolution will be (FSample / 2) / 512.
//
// Using 512 samples will allow any kind of cubic DP and any frequency
// ratio F1/F2.
// If using a normal ratio and the normal DP (2*F1-F2 and 2*F2-F1)
// The frequency resolution can be increased by using a bigger FFT and
// only saving points in the interesting part of the frequency range.
//
// E.G:
// At measure frequency 8kHz necessary bandwidth is 6 kHz (saving the
// range from 5kHz to 11kHz) when F1 / F2 ratio is 11 / 9.
//
// With a sample frequency of 26kHz the necessary points is 6/26*2048=472
// which can fit in the structure

#define       DPNSamples     512

// The field StimLevel contains desired level.
// The actual level will change from the desired level if e.g the ear volume
// is not the same as when calibrating the probe in a coupler (normally 2cc).
// If StimAdj is 1 (coupler) the stimulus level is not compensated for the
// actual acoustical conditions.
// If StimAdj is 2 (cavity) the stimulus level is adjusted to compensate for
// a different volume.
// If StimAdj is 3 (insitu) the stimulus level is adjusted by using the probe
// microphone to measure the actual level. In case of a high frequency pure tone
// stimulus (DP measurement) this can lead to misadjustment because of
```

```
// standing waves.
```

```
// The field SelectDP describes which DP product the fields DP1xxxxx and
// DP2xxxxx contain.
// SelectDP = 1 or 2 means that the fields DP1xxxxx and DP2xxxxx contains
// values for the normal DP products
// which is 2*F1-F2 and 2*F2-F1.
// SelectDP = 3 or 4 means DP product 3*F1-F2 and 3*F2-F1
// SelectDP = 5 or 6 means DP product 3*F1-2*F2 and 3*F2-2*F1

// An odd value of SelectDP means that the values in the DP1xxxxx fields
// should be used for the DP-Gram.

//
// An even value of SelectDP means that the values in the DP2xxxxx fields
// should be used for the DP-Gram.
//
//---------------------------------------------------------------------------
// Table of values for SelectDP
// 1 = 2*F1-F2
// 2 = 2*F2-F1
// 3 = 3*F1-F2
// 4 = 3*F2-F1
// 5 = 3*F1-2*F2
// 6 = 3*F2-2*F1
//---------------------------------------------------------------------------

// MinFreq specifies the frequency of index 0 in the "Sample" array.
//
// MaxFreq specifies the frequency of index ValidSamples-1 in the array.
//
// The samples in the array are equally spaced with a frequency distance of
// (MaxFreq - MinFreq) / (ValidSamples - 1).

typedef struct {
  int         StimAdj;      // 1=coupler, 2=volume corrected, 3=insitu
  TtimeWindow TimeWindow;   // Time window used
  THertz      F1;           // Input freq 1
  THertz      F2;           // Input freq 2
  TdB10       F1Level;      // SPL Level for F1
  TdB10       F2Level;      // SPL Level for F2
  int         SelectDP;     // See comments and table above
  TdB10       DP1Level;     // Output DP1 level
  TDg3600     DP1Phase;     // Output DP1 phase
  TdB10       DP1Noise;     // SPL Noise floor for DP1
  TdB10       DP2Level;     // Output DP2 level
  TDg3600     DP2Phase;     // Output DP2 phase
  TdB10       DP2Noise;     // SPL Noise floor for DP2
  int         AccMeas;      // Accepted measurements
  int         RejMeas;      // Rejected measurements
  TdB10       NRLevel;      // Noise rejection level
  THertz      MinFreq;      // Freq corresp to first sample
  THertz      MaxFreq;      // Freq corresp to last sample
  int         ValidSamples; // Number of valid samples
  TdB10       Sample[DPNSamples]; // Amplitude in dB SPL
} TDPPoint;


// DP norm name
```

```
typedef char TDPNormName[32];
```

```
// Definition of a DP-gram as 9 DP-gram points

#define       DPGramNPoint  9


typedef struct {
  TmaskSignal MaskSignal;   // Masking signal type applied to other ear
  THertz      MaskFreq;     // Masking signal frequency
  TdB10       MaskLevel;    // Masking level
  TDPNormName Norm;         // Norm name
  TDPPoint    Point[DPGramNPoint];
} TDPGram;

// Definition of DP-Gram OAE data
// DP-Gram data consists of up to MaxMeasNo DP-grams.
// Each DP-Gram consists of DPGramNPoint points with amplitude spectrums.

// ----------------------------------------------------------------------------
// FORMAT:    DataTypeCode = dtc_DPGRAMOAE_L or dtc_DPGRAMOAE_R (13/14)
//            DataFmtCodeStd = 200
// ----------------------------------------------------------------------------

typedef       TDPGram         TDPGramData[MaxMeasNo];

// Definition of an I/O-graph as a reference freq. and 10 I/O-graph points

#define       DPIONPoint    10

typedef struct {

  TmaskSignal MaskSignal;   // Masking signal type applied to other ear
  THertz      MaskFreq;     // Masking signal frequency
  TdB10       MaskLevel;    // Masking level
  TDPNormName Norm;         // Norm name
  THertz      Freq;         // Ref. Freq. Typical SQRT(F1*F2) or F2
  int         NPoint;       // number of points in IO-curve
  TdB10       F1StartLevel; // Start level of F1
  TdB10       F2StartLevel; // Start level of F2
  TdB10       F1Inc;        // Increment of F1
  TdB10       F2Inc;        // Increment of F2
  TDPPoint    Point[DPIONPoint];
} TDPIOCurve;


// Definition of DP-I/O OAE data
// DP-I/O data consists of up to MaxMeasNo I/O-graphs (input/output graph).
// Each I/O-graph is measured at a specific frequency and
// it consists of up to 10 points with amplitude spectrums.

// ----------------------------------------------------------------------------
//  FORMAT:   DataTypeCode = dtc_DPIOOAE_L or dtc_DPIOOAE_R (25/26)
//            DataFmtCodeStd = 200
// ----------------------------------------------------------------------------

typedef       TDPIOCurve    TDPIOData[MaxMeasNo];
```

```
#endif // Prevent multiple includes
```

Appendix C: The Floating Point type used in NOAH

IEEE Floating Point Standard

The IEEE Floating Point Standard is the most widely accepted standard representation for floating point numbers. The standard provides definitions for **single precision** and **double precision** representations:

Type of storage:	Length of storage in bits	Sign bit	Exponent	Mantissa
character	8	1	(not applicable)	7
integer	16	1	(not applicable)	15
long integer	32	1	(not applicable)	31
float	32	1	8	23
double	64	1	11	52
long double	80	1	15	64

The single precision IEEE Floating Point Standard format (in the table simply named “float”) is composed of 32 bits, divided into a 23 bit mantissa, M, an 8 bit exponent, E, and a sign bit, S:

<table>
<tr><td colspan="2">Byte at highest address</td><td colspan="2"></td><td></td><td>Byte at lowest address</td></tr>
<tr><td colspan="2">31 Byte 3 24</td><td colspan="2">23 Byte 2 16</td><td>15 Byte 1 8</td><td>7 Byte 0 0</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td></td><td></td></tr>
<tr><td>31</td><td colspan="2">30 … Exp … 23</td><td colspan="3">22 … Mantissa … 0</td></tr>
<tr><td>S</td><td colspan="2">E</td><td colspan="3">M</td></tr>
</table>

The mantissa is stored in a form called a "normalized mantissa". This means that the leftmost bit of the mantissa by definition is a one, and the IEEE format exploits this. For example:

5.375 = 4 + 1 + 1/4 + 1/8

in the binary form would look like:

101.011

The "normalised" form is obtained by adjusting the exponent until the decimal point is to the right of the most significant one:

$1.01011 * 2^2$

and the upper one IS NOT stored, except in the case of a long double. By not storing the most significant one, a greater range can be obtained.

The exponent is stored in a form called a "biased" exponent. The exponent field specifies the power of 2 by which the mantissa must be multiplied to obtain the value of the floating-point number. In order to accommodate negative exponents, the exponent field contains the sum of the actual exponent and a positive constant called the "bias".

This bias insures that the exponent field will always be a positive integer. The actual "bias" for floats is 127, doubles is 1023, and for long doubles it is 16383. Using a float for example, suppose the exponent field contained 132:

132 - 127 = 5

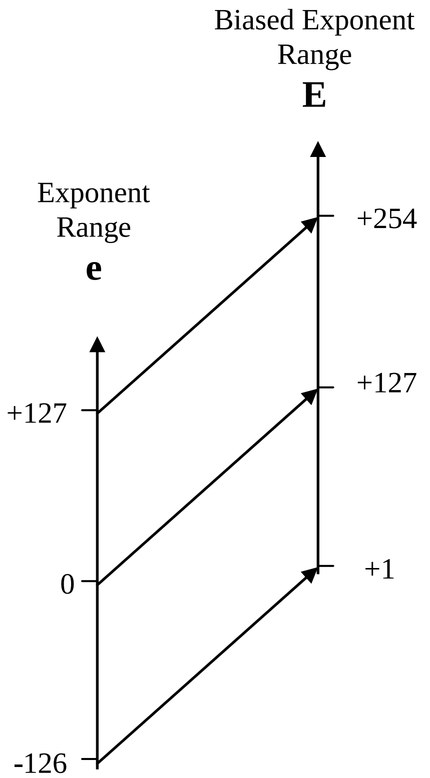
o in this scenario, the power by which the mantissa must be multiplied is 2^5. If the mantissa contained 122:

122 - 127 = -5

In this case the mantissa must multiplied by 2^{-5} to obtain the correct value. The 8087 chip reserves the highest and lowest exponents for handling errors, so the largest exponent is 127 and the lowest is (-126).

For the 32 bit Float, the bias is 127 and thus E = e + 127:

e	E
(Reserved)	255
127	254
…	…
1	128
0	127
-1	126
…	…
-126	1
(Reserved)	0



The sign bit, S, indicates the sign of the mantissa, with S = 0 for positive values and S = 1 for negative values. Zero is represented by E = M = 0. Since S may be 0 or 1, there are different representations for +0 and -0.

The maximum value of E = 255 is reserved to indicate **overflow** values (usually the result of floating point arithmetic) with exponents that are too large or too small to be represented. The special interpretations for E = 255 and M = 0 are PLUS_INFINITY for S = 0 and MINUS_INFINITY for S = 1. Floating point division by zero produces a number with E=255 and nonzero M called NaN (Not a Number).

The repeating Binary Fraction

In the above example, the fraction .375 was easy to represent in binary format. In this next example, the conversion leads to a repeating binary fraction:

To convert decimal 17.15 to IEEE Floating Point Standard:

Convert decimal 17 to binary 10001.

Convert decimal 0.15 to the repeating binary fraction

$0.00\overline{1001}$ Explanation: Repeat forever the $\overline{\text{binary period}}$

Combine integer and fraction to obtain binary

$10001.00\overline{1001}$

Normalise the binary number to obtain

$$1.000100\overline{1001} * 2^{4}$$

The number is positive, so S=0. The exponent e = 4, so E = 127 + 4 = 131 or binary 1000 0011. Align the values for M, E, and S in the correct fields (see figure below).

The hexadecimal value becomes hereby Hex 41 89 33 33:

31	30 23	22 0
	Exponent	Mantissa
S	E = 127 + 4	M = .000100 $\overline{1001}$ or:
0	100 0001 1	000 1001 0011 0011 0011 0011

The range of values for the mantissa, m, is [1 .. (2 – 2^{-23})] since delta_m = (2^{-23}) is the smallest decimal that can be represented in the mantissa.

delta_m = 2^{-23} approximates $1.192 * 10^{-7}$.

The mantissa thus represents a 24 bit binary fraction which corresponds to approximately 7 significant decimal digits.

The largest mantissa is:

31	30 23	22							0
S	Exponent	Mantissa = (2 – 2^{-23}) = MAX_MANTISSA: 2^{-1}	2^{-2}	2^{-2}	…	…	2^{-21}	2^{-22}	2^{-23}
		1	1	1	…	…	1	1	1

Because E = 0 and E = 255 are reserved, the range of values for the exponent, e, is [-126..127], see the table on the previous page.

The largest positive number that can be represented is approximately
maxFloat = $(2 * 2^{127}) = 2^{128}$

The decimal value of maxFloat is approximately $3.4 * 10^{38}$ or 3.4E38.

The smallest positive number that can be represented is accordingly
smallestPosFloat = $(1.0 * 2^{-126}) = 2^{-126}$

The decimal value of smallestPosFloat is approximately $1.18 * 10^{-38}$ or 1.18E38.

The resulting range for the 32-bit Float: [-3.4E38 .. –1.18E-38 OR 0.0 OR 1.18E-38 .. 3.4E38]
The resulting precision is 7 digits as shown above.

This text was prepared on the basis of web pages by Mitch Roth and Inprise Borland.

Index

A

B

C

D

E

F

I

K

L

M

N

O

P

Q

R

S

T

U

V

W