PROCITEC®

every signal is a message

DESCRIPTION

WAV FILE FORMAT

Version 1.1.1

Date: 4. November 2020

PROCITEC GmbH
Rastatter Straße 41
D-75179 Pforzheim

Imprint

WAV File Format

PROCITEC GmbH
Rastatter Straße 41
D-75179 Pforzheim
Germany
http://www.procitec.com/
info@procitec.de

Subject to technical modifications

Printed:

Table of Contents

1 Changes

Version	Date	Pages Modified	Author	Change due to
1.1.1	27-10-2020	5	AM	schema: replace xs:sequency with xs:all for complexType baseType
1.1	2017			layout updated
1.0				Initial version

2 General

2.1 Introduction

This document describes WAV file format in general and the extension for the storage of additional meta information about the signal.

2.2 Reference documents

No	Name	Comment
1	Multimedia Programming Interface and Data Specifications 1.0	Standard WAV file format specification, publicly available

3 WAV Filformat

Standard WAV file format is a RIFF-based format containing „fmt“ and “data” chunks. For the description of these chunks and the general RIFF format, please see /1/.

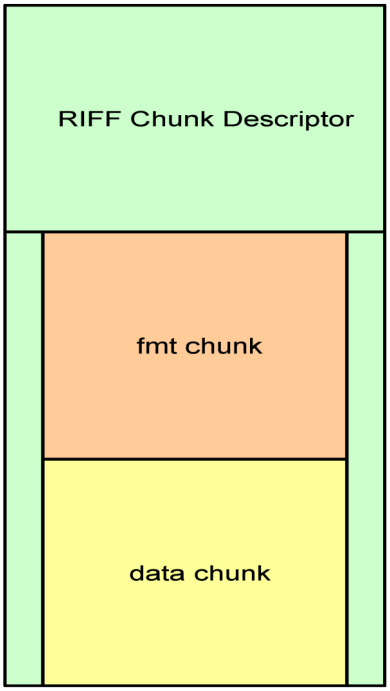

Figure 1: Standard WAV-format

The PROCITEC applications support the following formats:

- PCM 8/16/24/32 bit integer
- PCM 32 bit float
- A-law
- µ-law

Multichannel files are supported. For complex signals see the following section.

3.1 WAV-format extension to include custom data

Standard WAV format includes various meta information in its “fmt” chunk to describe the signal. For the use in COMINT systems, additional information regarding signal time, signal frequency etc. has to be added

To store this information in the WAV file, an additional ***meta*** chunk is added. The data in this chunk are stored in the XML format. This additional chunk will be ignored by standard tools which can load WAV files.

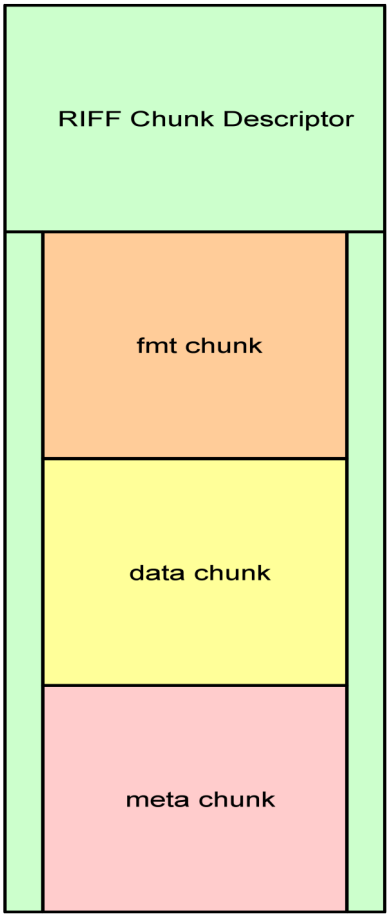

Figure 2: Extended WAV file format

The meta data chunk has the following structure:

Length [Byte]	Content	Description
4	`meta`	Header-Signature
4	<length>	Length oft the Meta-Header
	XML	Meta data in XML

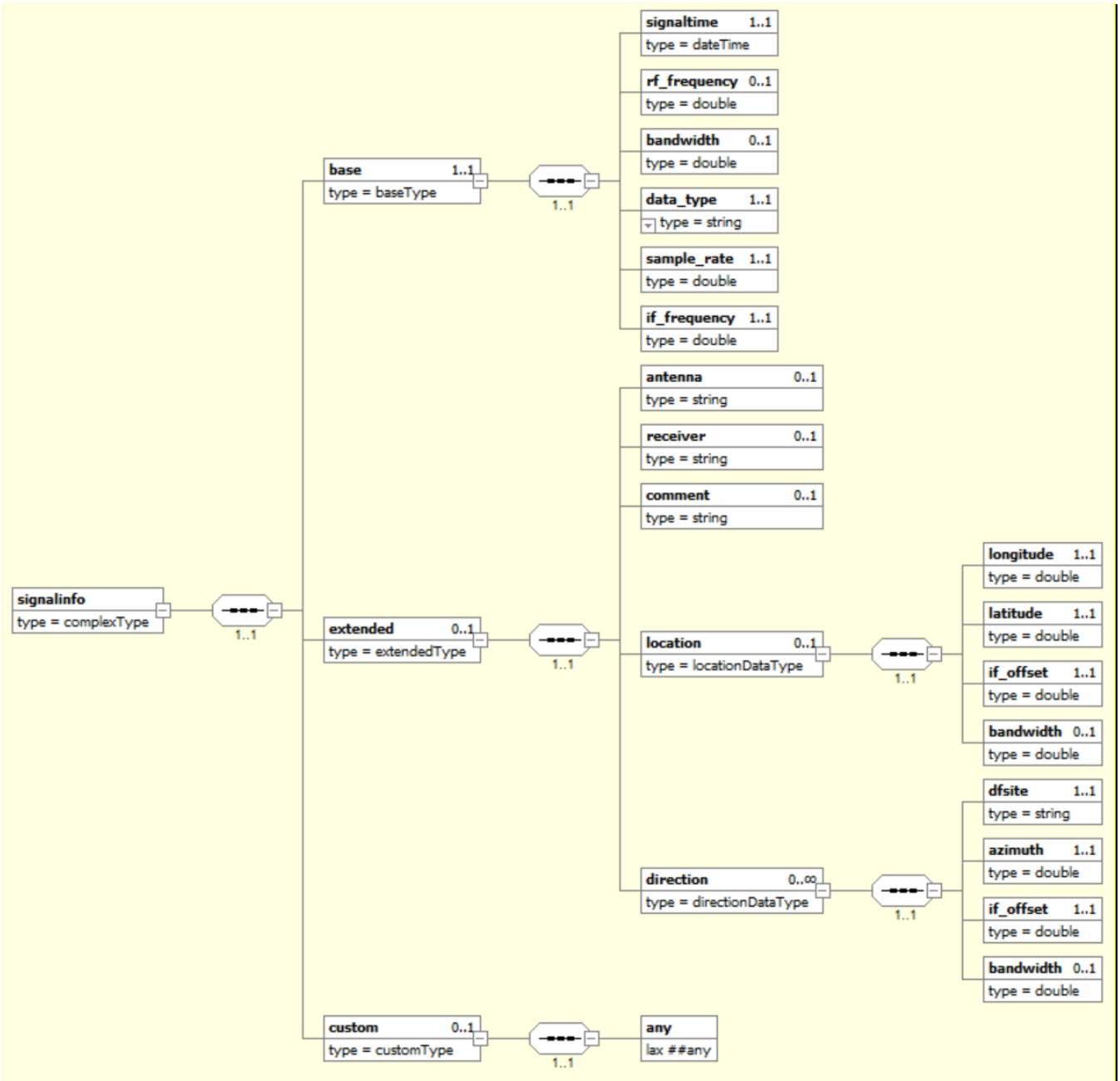

Figure 3: XML Schema

This is the definition of the XML-schema for the content of the meta chunk:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="quali-
fied" attributeFormDefault="unqualified">
	<xs:element name="signalinfo">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="base" type="baseType"/>
				<xs:element name="extended" type="extendedType"
minOccurs="0" maxOccurs="1"/>
				<xs:element name="custom" type="customType" minOccurs="0"
maxOccurs="1"/>
			</xs:sequence>
		</xs:complexType>
	</xs:element>
	<xs:complexType name="customType">
		<xs:sequence>
			<xs:any maxOccurs="unbounded" minOccurs="0"
processContents="lax"/>
		</xs:sequence>
	</xs:complexType>
	<xs:complexType name="extendedType">
		<xs:sequence>
			<xs:element name="antenna" type="xs:string" minOccurs="0"/>
			<xs:element name="receiver" type="xs:string" minOccurs="0"/>
			<xs:element name="comment" type="xs:string" minOccurs="0"/>
			<xs:element name="location" type="locationDataType"
minOccurs="0"/>
			<xs:element name="direction" type="directionDataType"
minOccurs="0" maxOccurs="unbounded"/>
		</xs:sequence>
	</xs:complexType>
	<xs:complexType name="baseType">
		<xs:all>
			<xs:element name="signaltime" type="xs:dateTime"/>
			<xs:element name="rf_frequency" type="xs:double" minOccurs="0"/>
			<xs:element name="data_type" default="real">
				<xs:simpleType>
					<xs:restriction base="xs:string">
						<xs:enumeration value="real"/>
						<xs:enumeration value="complex"/>
					</xs:restriction>
				</xs:simpleType>
			</xs:element>
			<xs:element name="sample_rate" type="xs:double"/>
			<xs:element name="if_frequency" type="xs:double" minOccurs="1"
default="0"/>
			<xs:element name="bandwidth " type="xs:double" minOccurs="0"/>
		</xs:all>
	</xs:complexType>
```

```
      <xs:complexType name="locationDataType">
            <xs:sequence>
                  <xs:element name="longitude" type="xs:double"/>
                  <xs:element name="latitude" type="xs:double"/>
                  <xs:element name="if_offset" type="xs:double"/>
                  <xs:element name="bandwidth" type="xs:double" minOccurs="0"/>
            </xs:sequence>
      </xs:complexType>
      <xs:complexType name="directionDataType">
            <xs:sequence>
                  <xs:element name="dfsite" type="xs:string"/>
                  <xs:element name="azimuth" type="xs:double"/>
                  <xs:element name="if_offset" type="xs:double"/>
                  <xs:element name="bandwidth" type="xs:double" minOccurs="0"/>
            </xs:sequence>
      </xs:complexType>
</xs:schema>
```

The length of the meta chunk must be multiple of 4 bytes.

The meta chunk will be added at the end of the WAV file to allow for adding the meta content without moving the data part of the WAV file.

The format of the meta chunk contains three detail-levels: *base, extended and custom*.

In the „base“ level, all elements important for the loading and processing of the IQ-file are stored:

Element	Description
signaltime	Time of the first sample in the file, UTC
rf_frequency	Receiver frequency in Hz. If signal input was not a real receiver (for example for file or stream input), or the frequency information for the input was not available, this field may be equal to 0Hz or some other relative frequency value.
data_type	"real" or "complex" signal
bandwidth	Signal bandwidth in Hz
sample_rate	Sampling rate (as floating point value). Sampling rate value in the standard WAV-header can store only integer values. Therefore, in case of non-integer sampling rate, there will be a small difference between actual sampling rate stored in a custom chunk and the value in the standard WAV header. To ensure correct signal handling, the value from custom chunk should be used. In case of sample rates higher than max 32-bit integer value (~2GHz), only the sampling rate from custom chunk should be used. The value in the standard WAV-header is not defined in that case.
if_frequency	IF-Frequency of the signal in Hz, always 0 for complex signals

In case of a complex signal, the WAV file will always contain 2 data channels.

In the ***extended*** level, additional signal information will be stored:

Element	Description
antenna	Antenna description
receiver	Receiver description

comment	Free comment
location	Location information
location/longitude	Longitude (-180° … +180°)
location/latitude	Latitude (-90° … +90°)
location/if_offset	Offset relative to if_frequency, used for the location info
location/bandwith	Bandwidth around if_offset frequency, used for the location info
direction	One or multiple directions
direction/dfsite	Direction finder name
direction/azimuth	Direction azimuth (0-360°)
direction/if_offset	Offset relative to if_frequency, used for the direction info
direction/bandwith	Bandwidth around if_offset frequency, used for the direction info

The ***custom*** level contains any further user-defined fields.

This is an example of the valid XML meta chunk content:

```
<?xml version="1.0" encoding="utf-8"?>
<signalinfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="signalinfo.xsd">
 <base>
  <signaltime>2009-02-01T05:56:45.234Z</signaltime>
  <rf_frequency>6075000</rf_frequency>
  <data_type>real</data_type>
  <sample_rate>20000</sample_rate>
  <if_frequency>12500</if_frequency>
  <bandwidth>16000</bandwidth>
 </base>
 <extended>
  <antenna>Antenne 1</antenna>
  <receiver>EM010</receiver>
 </extended>
```