

Table of contents

1. Introduction	2
2. File Header	2
3. Data Structures	3
4. Different File Headers and Data Structures	4
4.1 Text Data	4
4.1.1 Header Data	4
4.1.2 Data	4
4.2 SAT Data	4
4.2.1 Header Data	4
4.2.2 Data	4
4.3 Classifier Data	5
4.3.1 Header Data	5
4.3.2 Data	5
4.4 ClassifierCodeCheck Data	5
4.4.1 Header Data	5
4.4.2 Data	6
4.5 Graphics Data	6
4.5.1 Header Data	6
4.5.2 Data	7
4.6 FELDHELL Data	7
4.6.1 Header Data	7
4.6.2 Data	7
4.7 Sonogram Data	7
4.7.1 Header Data	7
4.7.2 Data	8
4.8 Signal Analysis Data	9
4.8.1 Header Data	9
4.8.2 Data	9

.WDA Data File Format

Application Note

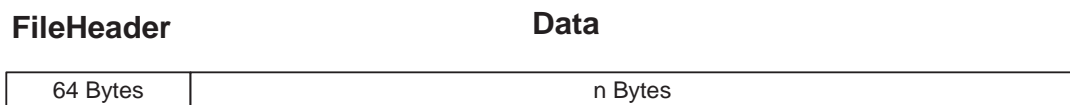
1. Introduction

In this document the .WDA file format is described. The .WDA file stores different kind of data:

- Text data
- Graphics data
- FELDHHELL data
- Signal Analysis data
- SAT data
- Classifier data
- Classifier CodeCheck data
- Sonogram data

The .WDA file contains a header at the beginning that identifies the data stored in the file and provides additional information dependent on the stored data.

The .WDA files are binary files, every file has the following structure:



The data at the begin of the file is the file header, the size of the header is 64 Bytes. Every .WDA file stores the same information in the first 64 Bytes of the file. Depending on the data stored, the file header can have some additional information stored, which extends the size of the Header, but this will be described later.

2. File Header

The file header that is at the beginn of every .WDA file stores the following information:

Signature	FileType	Version	LineCount	Pad	Pad2
8 Bytes	8 Bytes	4 Bytes	4 Bytes	8 Bytes	32 Bytes

The size of the file header is 64 bytes.

The **'Signature'** is always "WDA" (**NOTE:** by older versions it can be "W51PC", "W41PC" or "W40PC").

The **'FileType'** is dependent on the stored data and can have the following values:

- "Text"
- "Graphics"
- "FELDHHELL"
- "SigAnal"
- "SAT"
- "Classifr"
- "CCC"
- "Sonogram"

The **'Version'** is always "0x00020000" (Version A) or "0x00030000" (Version B). **'LineCount'** contains the number of data lines stored. (**NOTE:** the data stored is line based, that is a data package represents the data of a line.)

.WDA Data File Format

Application Note

'Pad' contains additional information to be stored.
'Pad2' contains additional information to be stored.

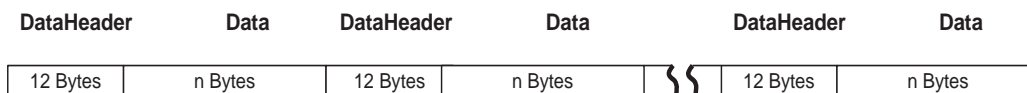
NOTE: the following struct type could be used to extract the file header.

```
struct FileHeader
{
    char Signature[8];
    char FileType[8];
    LONG Version;
    LONG LineCount;
    char Pad[8];
    char Pad2[32];
};
```

The FileHeader struct has a size of 64 Bytes.

3. Data Structures

After the Header the specific data is stored. The data is divided into several data packages, in the file header the number of data packages is stored.



A data package contains a data header and the data itself.



The size of the data header is 12 Bytes.

The data header at the begin of each data package stores the following additional information:



In **PrevSize** the size of the preceding data is stored, **Size** stores the size of the data contained in this data package and in **TIME** the timestamp of the data is stored. The data size value stored in **Size** indicates how many Bytes of data are stored after the data header for this data package.

NOTE: the following struct type could be used to extract the data header.

```
struct LineData
{
    LONG PrevSize;    // size of the data line before
    LONG Size;       // size of the data line
    LONG TIME;       // time of the data received
};
```

The LineData struct is 12 byte long.

.WDA Data File Format

Application Note

NOTE: the data packages in a .WDA file can vary in length. Because of this the size of the data is stored in the data header.

4. Different File Headers and Data Structures

In the following chapters the file headers and data packages of the different .WDA file types are described.

4.1 Text Data

The 'FileType' stored in the file header is "Text".

4.1.1 Header Data

This file type uses the 'File Header' described in chapter 2.

The following additional data is stored in the Header

- In Pad[0] the Font type
- In Pad[1] the Left to right Font type
- In Pad[2] if Unicode characters are stored

4.1.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

Every single line of a text document is stored in a separate data package, the length of the line is stored in the data header of the data package. The text data is stored in character values.

The characters stored in the file are from the type wchar_t, that is a character with the size of 2 bytes! That is each character of the Text lines occupies two bytes in the file.

4.2 SAT Data

The 'FileType' stored in the file header is "SAT".

4.2.1 Header Data

This file type uses the 'File Header' described in chapter 2.

The following additional data is stored in the Header:

In Pad[0] if the normal data or the extended data is stored in the file.

4.2.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

A data package in the .WDA file contains the data of one SAT session. The SAT session data is written to a single string. The following values are written to the string

- The Mode string value
- The Service string value
- The MES ID value
- The File string value, of the file that contains the session data
- The DateTime value
- The Duration value
- The Size value
- The Termination string value
- The Wavecom Card ID
- The computer address
- The flag value, that indicates if the session data file has already been displayed
- The Transmitter ID value

.WDA Data File Format

Application Note

NOTE: the values are written to the string in the same order as listed above!

If not stored with `_UNICODE` enabled, then the string contains normal 8-bit characters values, from type `char`, otherwise the string contains 16-bit characters, from type `wchar_t`. A `char` value occupies 1 Byte and a `wchar_t` value occupies 2 Bytes in the binary .WDA file.

4.3 Classifier Data

The 'FileType' stored in the file header is "Classifr".

4.3.1 Header Data

This file type uses the 'File Header' described in chapter 2.

The following additional data is stored in the Header:

- In `Pad[0]` the Timestamp format is stored

4.3.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

A data package in the .WDA file contains the data of one classified signal.

The signal data of a signal detected by the classifier is written to a single string. The following values are written to the string:

- The Signal name string value
- The Mode string value
- The Modulation string value
- The Center Frequency string value including the unit label
- The Shift string value including the unit label
- The Baudrate string value including the unit label
- The Bandwidth string value including the unit label
- The Confidence string value including the unit label
- The Mode value
- The Center Frequency value
- The Shift value
- The BaudrateA value
- The BaudrateB value
- The Bandwidth value
- The Time stamp value

NOTE: the values are written to the string in the same order as listed above!

If not stored with `_UNICODE` enabled, then the string contains normal 8-bit characters values, from type `char`, otherwise the string contains 16-bit characters, from type `wchar_t`. A `char` value occupies 1 Byte and a `wchar_t` value occupies 2 Bytes in the binary .WDA file.

4.4 ClassifierCodeCheck Data

The 'FileType' is "CCC".

4.4.1 Header Data

This file type uses the 'File Header' described in chapter 2.

The following additional data is stored in the Header:

- In `Pad[0]` the Timestamp format is stored

.WDA Data File Format

Application Note

4.4.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

A data package in the .WDA file contains the data of one classified signal or the CodeCheck data of a classified signal. The data stored for the Classifier CodCheck is divided into two parts, the Classifier part and the CodeCheck part. A .WDA file contains at first the Classifier data and then the CodeCheck data.

The Classifier data is stored in the way described in chapter 4.3. The data package which contains the text "CodeCheck Data Section" marks the begin of the CodeCheck data section.

The CodeCheck data of a classified signal is written to a single string. The following values are written to the string:

- The index value of the classified signal
- The string value with all predicted modes
- All detected modes(can be more than one), for a detected mode the following values are written to the string:
 - The Mode name string value
 - The SubMode name string value
 - The string value with the additional description
 - The Hits value

NOTE: the values are written to the string in the same order as listed above!

If not stored with _UNICODE enabled, then the string contains normal 8-bit characters values, from type char, otherwise the string contains 16-bit characters, from type wchar_t.

A char value occupies 1 Byte and a wchar_t value occupies 2 Bytes in the binary .WDA file.

4.5 Graphics Data

The 'FileType' stored in the file header is "Graphics".

4.5.1 Header Data

This file type uses the 'File Header' described in chapter 2.

The Phase value, the AspectRatio (could be stored in old Wavecom Data Files, for the actual Software it is not stored anymore) and the Flip Left/Right/Top/Bottom value are stored additionally to the default header.

The following additional data is stored in the Header, following the default header:

FileHeader	Phase	Pad	AspectRatio	FlipHV	Pad
64 Bytes	4 Bytes	4 Bytes	8 Bytes	4 Bytes	4 Bytes

- Phase: the Phase value
- Pad: padding bytes for alignment (not used)
- AspectRatio: the AspectRatio value
- FlipHV: the Flip Left/Right/Top/Bottom value
- Pad: padding bytes for alignment (not used)

To the default file header with 64 bytes additional a block of 24 bytes is added that contains the Phase, the AspectRatio, the FlipHV and some padding bytes for alignment. The Phase value occupies 4 Bytes followed by 4 padding bytes. The Aspect Ratio occupies 8 bytes. The FlipHV (Flip horizontal vertical) value occupies 4 bytes followed by 4 padding bytes.

NOTE: the following struct type could be used to extract the data header.

.WDA Data File Format

Application Note

```
struct GraphicsData  
{  
    long   Phase;  
    double AspectRatio;  
    int    FlipHV;  
};
```

4.5.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

A data package in the .WDA file contains the data of all pixels of one line in the graphic. The pixel data is stored in BYTES (8-bit type) values, that is one Byte in the .WDA file contains the value of an Pixel in the graphic.

4.6 FELDHELL Data

The 'FileType' is "FELDHELL".

4.6.1 Header Data

This file type uses the 'File Header' described in chapter 2.

No additional data is stored in the Header.

4.6.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

A data package in the .WDA file contains the data of all pixels of one line in the graphic. The pixel data is stored in unsigned char values (8-bit type), that is one Byte in the .WDA file contains the value of an Pixel in the graphic.

4.7 Sonogram Data

The 'FileType' stored in the file header is "Sonogram".

4.7.1 Header Data

This file type uses the 'File Header' described in chapter 2.

The following additional data is stored in the Header, following the default header:

FileHeader	DocSizeX	DocSizeY	CursorA	CursorB	Bandwidth
64 Bytes	4 Bytes	4 Bytes	8 Bytes	8 Bytes	4 Bytes
	4 Bytes	4 Bytes	4 Bytes	2 Bytes	2 Bytes
	4 Bytes				4 Bytes
Translation	LowFreq Limit	HighFreq Limit	LowMag Limit	HighMag Limit	VideoFilterOn

- DocSizeX: the document width
- DocSizeY: the document height
- CursorA: the value of the cursorA
- CursorB: the value of the cursorB
- Bandwidth: the Bandwidth value

.WDA Data File Format

Application Note

- Translation: the Translation value
- LowFreqLimit: the lower frequency limit value
- HighFreqLimit: the higher frequency limit value
- LowMagLimit: the lower magnitude limit value
- HighMagLimit: the higher magnitude limit value
- VideoFilterOn: the VideoFilter On/Off flag

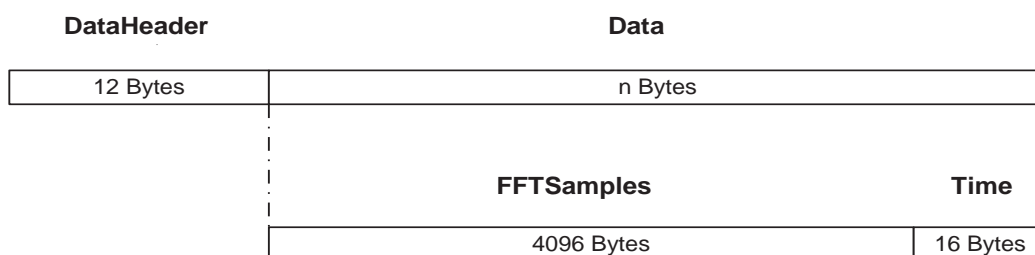
To the default file header with 64 bytes additional a block of 48 bytes is added. The additional block contains display parameter used by the Wavecom GUI for displaying the stored data correctly.

NOTE: the following struct type could be used to extract the data header.

```
struct SonogramData
{
    long    IDocSizeX;
    long    IDocSizeY;
    double  dCursorA;
    double  dCursorB;
    long    lBandwidth;
    long    lTranslation;
    long    lLowFrequencyLimit;
    long    lHighFrequencyLimit;
    short   sLowMagnitudeLimit;
    short   sHighMagnitudeLimit;
    BOOL    boVideoFilterOn;
};
```

4.7.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data. The stored sonogram data has the following structure:



- FFTSamples: the first 4096 bytes of the stored data contain the 2048 sample values for the FFT.
- Time: the last 16 bits of the stored data contains the systemtime timestamp value of the received data (the additional timestamp is necessary because the default timestamp in the data header does not support milliseconds).

The data for the 'Sonogram' file types occupies 4112 bytes inside a data package after the data header.

A data package in the .WDA file contains the datapoints of one FFT sample and its timestamp value. The FFT sample data points are stored in short values (2-bit type), that is 2 Bytes in the .WDA file contains the value of one FFT sample data point.

NOTE: the following struct type could be used to extract the data.

.WDA Data File Format

Application Note

```
typedef struct ST_SONOGRAM_DATA  
{  
    short asFFTSamp[2048];  
    SYSTEMTIME Time;  
} ST_SONOGRAM_DATA;
```

4.8 Signal Analysis Data

The 'FileType' stored in the file header is "SigAnal".

4.8.1 Header Data

This file type uses the 'File Header' described in chapter 2. The following additional data is stored in the Header, following the default header:

FileHeader	CursorHide	LabelX	LabelY	Pad
64 Bytes	1 Byte	40 Bytes	40 Bytes	3 Bytes
	4 Bytes	8 Bytes	8 Bytes	8 Bytes
	XSizePhys	YSizePhys	XSizeUser	YSizeUser

- CursorHide: flag that indicates if the cursor are hidden
- LabelX: the x-axis unit label string
- LabelY: the y-axis unit label string
- Pad: padding bytes for alignment (not used)
- XSizePhys: the physical width of the data range
- YSizePhys: the physical height of the data range
- XSizeUser: the width of the data range in the user interface
- YSizeUser: the height of the data range in the user interface

To the default file header with 64 bytes additional a block of 112 bytes is added.

The additional block contains display parameter used by the Wavcom GUI for displaying the stored data correctly.

NOTE: the following struct type could be used to extract the data header.

```
struct SigAnalData
```

```
{  
    char    m_nCursorHide;  
    char    m_strLabelX[40];  
    char    m_strLabelY[40];  
    long    m_Phys_xSize;  
    double  m_Phys_ySize;  
    double  m_User_xSize;  
    double  m_User_ySize;  
};
```

4.8.2 Data

This 'FileType' uses the data structure described in chapter 3 to store the data.

A data package in the .WDA file contains the data of all datapoints of one line in the graphic. The data points are stored in long values (4-bit type), that is 4 Bytes in the .WDA file contains the value of one data point in the graphic.

.WDA Data File Format

Application Note

Since more than thirty years Wavecom Elektronik AG has developed, manufactured and distributed high quality devices and software for the decoding and retrieval of information from wireless data communication in all frequency bands. The nature

of the data communication may be arbitrary, but commonly contains text, images and voice. The company is internationally established within this industry and maintains a longstanding, world-wide network of distributors and business partners.

Product Information

Products	http://www.wavecom.ch/product-summary.php
Datasheets	http://www.wavecom.ch/brochures.php
Specifications	http://www.wavecom.ch/product-specifications.php
Documentation	http://www.wavecom.ch/manuals.php
Online help	http://www.wavecom.ch/content/ext/DecoderOnlineHelp/default.htm
Software warranty	One year free releases and bug fixes, update by DVD
Hardware warranty	Two years hardware warranty
Prices	http://www.wavecom.ch/contact-us.php

System Requirements

	<i>Minimum</i>	<i>Recommended</i>
CPU	P4 Dual-Core 2.4 GHz	Core i5 or Core i7 2.8 GHz
Memory	2 GB RAM	4 - 8 GB RAM
OS	Windows XP	Windows 7 32-bit or Windows 7 64-bit

Distributors and Regional Contacts

You will find a list of distributors and regional contacts at <http://www.wavecom.ch/distributors.php>