Caution!
This is no official format description.
The information is derived from input data of the device and may be not correct.

# Wacom Inkling WPI file format description.

The pen input is divided layers and strokes. A stroke starts when the pen hits the paper and ends when the pen leaves the paper. A stroke consists of points with the following data

1. Pen x/y position
2. Pen pressure
3. Pen x/y tilt

The first 2059 Bytes are unknown data. It seams that they do not belong to the pen data and can be ignored.
After the first 2040 (((2059))) Bytes there are the pen data. They consist of different data blocks. A data block has the following format:

| | |
|---|---|
| Block Descriptor | 1 Byte |
| Block Size | 1 Byte |
| Block Data | Blocksize -2 Bytes |

The following block descriptors are known:

| | | |
|---|---|---|
| 1. | Stroke/Layer Descriptor | 241 |
| 2. | Pen x/y Data | 97 |
| 3. | Pen Pressure | 100 |
| 4. | Pen Tilt | 101 |
| 5. | Unknown | 197 Size 17 |
| 6. | Unknown | 194 Size 4 |
| 7. | Unknown | 199 Size 28, Size 24, Size 20 |

Block descriptions:
Layer descriptor (241):

| | | |
|---|---|---|
| 241 | 3 | Type |

The following types are used:

| | |
|---|---|
| 00 | End of Stroke |
| 01 | Start of Stroke |
| 128 | New Layer |

Pen x/y Data (96)

| | | | | | |
|---|---|---|---|---|---|
| 97 | 6 | X High | X Low | Y High | Y Low |

The resolution of the Y Position is the half of the X position.

The Value of the Y position is multiplied by 2 and then incremented by 5. This is done that the output values matches the values of the .WAC output file.

Pen pressure 96

| | | | | | |
|---|---|---|---|---|---|
| 100 | 6 | not used | not used | P High | P Low |

The resolution of the Y Position is the half of the X position.

Pen tilt 101

| | | | | | |
|---|---|---|---|---|---|
| 101 | 6 | Tilt X | Tilt Y | not used | not used |

The file ist organized in the following way:
Start of Stroke

| | | |
|---|---|---|
| Pen x/y Data | Pen Pressure | Pen Tilt |
| End of Stroke | | |
| | | |
| Start of Stroke | | |
| Pen x/y Data | Pen Pressure | Pen Tilt |
| End of Stroke | | |
| New Layer | | |
| Start of Stroke | | |
| Pen x/y Data | Pen Pressure | Pen Tilt |
| End of Stroke | | |

Class to read Wacom inkling WPI Files.
The class PaperInkConverter reads the strokes in a Wacom inkling WPI File.

Constructror:
```
PaperInkConverter::PaperInkConverter(wchar_t* Filename,
                                     int StrokeThreshold,
                                     int AllThreshold,
                                     int SlidingThreshold);
```

The constructor opens a .WPI File and reads the first stroke.
Parameter:

| | |
|---|---|
| Filename | Filename of the input File |
| StrokeThreshold | Relative pressure threshold within a strokes in %.<br>The maximum pressure within a stroke is calculated. Points with a pressure less than<br>StrokeThreshold * Max pressure in stroke / 100<br>are ignored. |
| AllThreshold | Relative pressure threshold of all strokes in %.<br>The maximum pressure of all strokes until this point calculated. Points with a pressure less than<br>AllThreshold * Max pressure of all strokes / 100<br>are ignored. |
| SlidingThreshold | Not used |

The thresholds improve the handwriting recognition. There is a better separation between two characters.

If the file cannot be opened, or the stroke start cannot be found than the Number of Points is negative. (see GetNumberOfPoints)
The first stroke is available immediately after construction. No ReadNextStroke is necessary to get the data of the first stroke

```
int PaperInkConverter::GetNumberOfPoints()
```
Get the number of points for the actual stroke
Parameter:
Return Value:
    Number of Points in the current stroke
If End of File is reached or if the file cannot be opened by the constructor, than the return value is -1

`short* PaperInkConverter::GetPenPositionX();`

Get the pen X positions of the current stroke

Parameter:

Return Value:

Array of short values with the X-Values of the pen position of the stroke

`short* PaperInkConverter::GetPenPositionY()`

Get the pen Y positions of the current stroke

Array of short values with the Y-Values of the pen position of the stroke

`unsigned char * PaperInkConverter::GetPenPressure()`

Get the pen pressure of the current stroke

Array of short values with the X-Values of the pen pressure of the stroke

`unsigned char * PaperInkConverter::GetPenTiltX()`

Get the pen X tilt of the current stroke

Array of short values with the pen tilt X-Values of the stroke

`unsigned char * PaperInkConverter::GetPenTiltY()`

Get the pen Y tilt of the current stroke

Array of short values with the pen tilt Y-Values of the stroke

`int PaperInkConverter::ReadNextStroke();`

Read the next stroke

Parameter:

None

Return Value:

| | |
|---|---|
| 0 | More Strokes found in layer |
| <> 0 | No more Strokes found in Layer (End of File or Next Layer) |

When the functions returns a value <> 0 than either end of file has been reached or the layer has changed (test it with EOF_Found). If the layer changed, the values for the new layer are still available.

`int PaperInkConverter::EOF_Found()`

Check if end of file has been reached

Return Value:

| | |
|---|---|
| MANAGED_TRUE | End of file found |
| MANAGED_FALSE | More strokes available |

`void PaperInkConverter::SetAllSensitivity(int Sensitivity);`

set a new value for AllSensitivity (see constructor)

`void PaperInkConverter::SetStrokeSensitivity(int Sensitivity);`

set a new value for StrokeSensitivity (see constructor)

If a sensitivity value is used, than an original stroke is split into substrokes. In this case ReadNextStroke reads the next substroke.