



Topaz Systems, Inc.

Topaz Systems, Inc.
650 Cochran Street, Unit 6
Simi Valley, CA, 93065

www.topazsystems.com
tech support: 805 520-8286
support@topazsystems.com

©1995-2006, all rights reserved, US patent 6,307,955, and pending

OVERVIEW

The SigPlus J2ME implementation allows the developer to pass the raw data obtained over a serial port or bluetooth connection from the signature pad in, and output as Topaz SIG file format data, which can be used later with the full version of the SigPlus java bean for further work.

You are responsible for obtaining the data coming over the bluetooth connection, which can then be added to the bean through the `addToInputBuffer()` method. You'll see the demo provided opens a DAT file and gets the data to add to the `addToInputBuffer()` method. Of course, you will get the data via bluetooth.

Then, the `writeSigCompressed()` returns a Topaz SIG file that you can return to the full java bean (available at http://www.topazsystems.com/Software/sigplusjava2_47.zip) or to any of the other software development tools (see <http://www.topazsystems.com/Software/proindex.htm> in the upper right...all of those tools accept SIG files). A good example showcasing the methods below is the `MicroSigPlusDemo.jav`. Please refer to it for code examples.

FUNCTION CALL SUMMARY

SigPlus MicroUtl is comprised of 4 classes total:

1. `MicroTabletInterface` - opens the `microUtl`, to allow data to be added, has the function to add the data, and closes the `microUtl` allowing it to do other work
2. `MicroSigWriter` - writes out the data as a sig file
3. `MicroParameter` - sets up the `microUtl` for the right tablet model
4. `MicroSigDataType` - a helper class for `MicroSigWriterClass`.

MicroTabletInterfaceMethods:

Constructor Method

MicroTabletInterface (MicroTabletParamaters, MicroSigDataType)

- Function: To create an instance of the MircoTabletInterfaceClass you need an instance of the MicroTabletParamaters class, and the MicroSigDataType Class.
- Example: “tablet = new MicroTabletInterface(tabParams, sig)”

OpenTablet ()

- Function: Ready SigPlus MicoUtl to accept the data buffer.
- Return Value: Void
- Example: “tablet.openTablet()”

AddToInputBuffer (byte() Buffer, int BufferSize)

- Function: To add the data you’ve collected from the pad
- Byte () Buffer – Data accumulated from the pad
- IntBufferSize – Size of the buffered data.
- ReturnValue: Void
- Example: “tablet.addToInputBuffer(inputData, inputData.length)”

CloseTablet ()

- Function: Called when you are done passing in the data buffer.
- ReturnValue: Void
- Example: “tablet.closeTablet()”

BytesInInputBuffer ()

- Function: Checks whether the buffer is empty and whether we should continue.
- Return Value: Integer
- Example: “if (tablet.bytesInInputBuffer() == 0)
 {
 break;”

MicroSigWriterMethods:

Constructor

MicroSigWriter (intCompressionMode)

- There is only one mode available, and that is 1.
- Responsible for writing out the SIG file format data
- Example: “writer = new MicroSigWriter(1)”

WriteSigCompressed (MicroSigDataType)

- Function: Returns the sigfile as a byte array that can be written out to file.
- Return Value: Byte()
- Example: “outputData = writer.writeSigCompressed(sig)”

MicroTabletParameters Methods

SetTabletModel (String TabletModel)

- Sets up the proper model used for signature capture. Please refer to the SigPlusJava documentation for complete listing of Tablet Model Strings.
- Get Tablet Model Strings here:
 - <http://www.topazsystems.com/software/javadocs/class.html#setTabletModel>
- Example: "tabParams.setTabletModel("SignatureGem4X5")"

MicroSigDataType

This class is used as a helper class for MicroSigWriterClass.

FAQ regarding the J2ME API:

1. What is the meaning of the input parameter for MicroSigWriter constructor?

```
//MicroTabletParameters  tabParams = new MicroTabletParameters();  
    //MicroSigDataType      sig      = new MicroSigDataType();  
//MicroTabletInterface  tablet = new MicroTabletInterface(tabParams,sig);  
    //MicroSigWriter        writer  = new MicroSigWriter( 1 );
```

The parameter is the compression factor, MicroSigWriter only produces compressed sig strings, and if 0 is used, it is forced to 1.

2. What model should I specify if I'm using DeskGem1x5 pad(with Bluetooth //connectivity)?

```
tabParams.setTabletModel( "SignatureGem1X5" );
```

3. What's happening when I call openTablet() ?

```
//tablet.openTablet();
```

A thread is started to process the raw tablet data as it is added to the threads input queue.

```
tablet.addToInputBuffer( _receiveBuffer, _receiveBuffer.length );
```

4. I have some raw signature data in _receiveBuffer, got it via Bluetooth

```
//from the pad. Will the method above copy the data into some internal  
    //MicroSigWriter's buffer?
```

```
tablet.addToInputBuffer( _receiveBuffer, _receiveBuffer.length );
```

The raw tablet data is placed into a queue to be processed into point and stroke data. A separate thread takes the data from this queue and processes it.

5. What is going on in that method?

```
//tablet.closeTablet();
```

closeTablet shuts down the input data processing thread.

6. How do you get the signature as an image from the compressed SIG file format?

You can return the points/strokes and redraw the signature. Here's the methods with which he can gain access to the points/strokes:

```
        tablet.closeTablet();
        SigDataPoint point = new SigDataPoint(0,0);

        int num_points = 0 ;
        int num_strokes = sig.strokes.size();
        Status.show("num_strokes=" + num_strokes);
        for(int i = 0; i < num_strokes; i++ )
        {
            num_points = sig.getNumberOfPointsInStroke(i);
            Status.show("num_points=" + num_points);
            for(int j = 0; j < num_points; ++j )
            {   point = sig.getPoint(i, j);
                .....
            }
        }
    }
```

Depending on processor speed and loading, there may be a brief delay between the data being consumed from the input buffer, and the end of processing to produce the stroke data. I would suggest a short sleep once the input buffer is empty to allow for the rest of the processing. This is due to things happening in different threads.

For more information regarding products and support
Email: support@topazsystems.com
Phone: (805) 520-8286

www.topazsystems.com
650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282; Fax: 805 520-0867