

Developer's How To Guide

[How do I create a graph component?](#)

[Which renderers does MSEpedite support. How do I use them?](#)

[How do I plot data points?](#)

[How do I add zooming capability?](#)

[How do I add the default tooltip?](#)

[How do I implement a custom tooltip?](#)

[How do I create a graph component?](#)

Include MSEpedite.jar as a library in your project. The quickest way to create a graph component is to instantiate a SessionGraph. The code snippet below creates a graph and places it in a frame.

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.JFrame;
import org.proteomecommons.MSEpedite.app.SessionGraph;

public class Main {

    /** Creates a new instance of Main */
    public Main() {
        //Create the Graph
        SessionGraph graph = new SessionGraph();

        //Create a frame to contain the graph
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(new BorderLayout());
        frame.getContentPane().add(graph, BorderLayout.CENTER);
        frame.setSize(new Dimension(800,800));
        frame.setVisible(true);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        new Main();
    }
}
```

[Which renderers does MSEpedite support. How do I use them?](#)

Three renderers currently exist: bar, line, dot. The constructor takes three parameters. A string which will become a label for the renderer; the instance of a graph object; an instance of a DataSource. In the vast majority of the implementations, a graph object will also be an instance of DataSource. In the code snippet below shows the

construction of the three renderers. Note that MSExpedite will draw using all renderers that have been added to the graph object. In the snippet below, although three renderers are constructed, only one is added to the graph object and used.

```
//Construct an instance of SessionGraph
SessionGraph graph = new SessionGraph();
//Instantiate a LineDrawer to render spectra as lines
LineDrawer lineRenderer = new LineDrawer("Line", graph,(DataSource) graph);
//Instantiate a BarDrawer to render spectra as bars
BarDrawer barRenderer = new BarDrawer("Bar", graph,(DataSource)graph);
//Instantiate a DotDrawer to render spectra as dots
DotDrawer dotRenderer = new DotDrawer("Dot",graph,(DataSource)graph);

//Add only one renderer to the graph
graph.clearDrawers (); //clear any previous renderers
graph.addRenderer(lineRenderer);
```

How do I plot data points?

Construct an Array2D object. In the example that follows, the pair (1,1)...(10,10) are plotted as a Bar graph.

```
SessionGraph graph = new SessionGraph();
BarDrawer bar=new BarDrawer("Bar", graph,(DataSource)graph);
graph.add(bar);
float[] x={1,2,3,4,5,6,7,8,9,10};
float[] y={1,2,3,4,5,6,7,8,9,10};
Array2D a2d = new Array2D(x,y);
graph.set("MyDataLabel",a2d);
```

How do I add zooming capability?

Construct a ZoomController object as follows.

```
SessionGraph graph = new SessionGraph();
BarDrawer bar=new BarDrawer("Bar", graph,(DataSource)graph);
graph.add(bar);
float[] x={1,2,3,4,5,6,7,8,9,10};
float[] y={1,2,3,4,5,6,7,8,9,10};
Array2D a2d = new Array2D(x,y);
graph.set("MyDataLabel",a2d);
ZoomController zoom = new ZoomController((IOjectDrawer)graph,graph);
zoom.enable(true);
```

How do I add the default tooltip?

The tooltip does a calculation of the signal to noise ratio for the data point near the location of the cursor. The only parameter required to be set is the window half size

centered at the point of the cursor. The signal to noise will be calculated using the datapoints in this window as the background.
For MALDI a value of 10 is good.

```
SessionGraph graph = new SessionGraph();
MetaData metaData=graph.getMetaData();
MSLevelAttr mslevel = metaData.getMsLevelAttr();
//Here is where the window half size is set to 10
mslevel.getPeakDetectionDefinition().getSignalToNoise().setWindowHalfSize(10);
```

```
DatapointSensor datapointSensor = new DatapointSensor((IObjectDrawer) graph,
graph,new ToolTipController(graph,metaData));
```

```
datapointSensor.dataProbeOn(true);
```

How do I implement a custom tooltip?

Implement the interface `org.proteomecommons.MSExpedit.Grahp.ToolTipWriter` and pass the resulting object to the constructor of a `DatapointSensor()`; The workhorse method is

```
public String[] getText(int index);
```

The index into the datapoint array (`Array2D`) supporting the graph is passed to the method. This index represents the (mz,Int) pair closest to the cursor.

Example:

This tooltip will display the mass and intensity that is closest to the cursor.

```
class MassIntensityTTW implements ToolTipWriter{
    Graph gr = null;

    public MassIntensityTTW(SessionGraph gr){
        set(gr);
    }

    public void set(Graph chart) {
        gr = chart;
    }

    public String[] getText(int index) {
        Array2D a2d = gr.getDataPoints();
        float fmz = a2d.x[index];
        float fintensity=a2d.y[index];

        String mz = ""+fmz;
```

```
String intensity="" + fintensity;  
  
String s[] = {"Intensity = " + intensity, "Mass = " + mz};  
return s;  
}  
}
```

To use the resulting object instantiate a DatapointSensor object as follows

```
DatapointSensor datapointSensor = new DatapointSensor((IOjectDrawer) graph,  
graph, new MassIntensityTTW(graph));  
datapointSensor.dataProbeOn(true);
```