

The actual code

```
public class TraxDemo {
    public static void main (String argv[]){
        if (argv.length != 3) {
            System.out.println("usage: TraxDemo xslSource xmlSource outFile");
            System.exit(2);
        } else new TraxDemo(argv[0],argv[1],argv[2]);
    }
    TraxDemo(String xslSourceFile, String xmlSourceFile, String outFile){
        Source xslSS=new StreamSource(xslSourceFile);
        StreamSource xmlSS=new StreamSource(xmlSourceFile);
        TransformerFactory tf = TransformerFactory .newInstance();
        Transformer t=null;
        try {
            t = tf.newTransformer(xslSS);
        } catch (TransformerConfigurationException ex) {ex.printStackTrace();}
        try {
            t.transform(xmlSS, new StreamResult(new FileOutputStream(outFile)));
        } catch (FileNotFoundException ex) {ex.printStackTrace();}
        } catch (TransformerException ex) { ex.printStackTrace(); }
    }
}
```

xml.transform packages

DOM SAX

<i>Package</i>	<i>Description</i>
javax.xml.transform	Defines the TransformerFactory and Transformer classes, which you use to get a object capable of doing transformations. After creating a transformer object, you invoke its transform() method, providing it with an input (source) and output (result).
javax.xml.transform.dom	Classes to create input (source) and output (result) objects from a DOM.
javax.xml.transform.sax	Classes to create input (source) from a SAX parser and output (result) objects from a SAX event handler.
javax.xml.transform.stream	Classes to create input (source) and output (result) objects from an I/O stream.

TrAX main classes

DOM SAX

javax.xml.transform.Transformer
transform(Source xmls, Result output)

javax.xml.transform.sax.SAXResult implements Result
javax.xml.transform.sax.SAXSource implements Source

javax.xml.transform.stream.StreamResult implements Result
javax.xml.transform.stream.StreamSource implements Source

javax.xml.transform.dom.DOMResult implements Result
javax.xml.transform.dom.DOMSource implements Source