

# UseCocoonXMLSerializerCode

```
#FORMAT Java
public InputStream getInputStream(...my params...)
throws IOException, SourceNotFoundException {

    Serializer serializer = createSerializer();

    ByteArrayOutputStream outputStream = null;
    try {
        outputStream = new ByteArrayOutputStream();
        serializer.setOutputStream(outputStream);

        // fill the serializer with SAX Events (it's a ContentHandler)
        MyBrilliantClass.toSAX(serializer, ...my params...);

        return new ByteArrayInputStream(outputStream.toByteArray());
    } catch (SAXException e) {
        throw new CascadingIOException(e);
    } finally {
        if (null != outputStream)
            outputStream.close();
    }
}

private Serializer createSerializer() {
    // we use Cocoon's XMLSerializer here which is configurable in terms
    // of the Transformer Implementation to use (some are buggy, so we
    // want to ensure which one is used)

    DefaultConfiguration transformerConfig =
        new DefaultConfiguration("transformer-factory", null);
    transformerConfig.setValue(
        "org.apache.xalan.xsltc.trax.TransformerFactoryImpl");

    DefaultConfiguration serializerConfig =
        new DefaultConfiguration("serializer", null);

    // here we use cocoon's serializer
    serializerConfig.setAttribute("class",
        "org.apache.cocoon.serialization.XMLSerializer");

    serializerConfig.addChild(transformerConfig);

    XMLSerializer serializer = new XMLSerializer();
    serializer.enableLogging(new NullLogger());
    try {
        serializer.configure(serializerConfig);
    } catch (ConfigurationException e) {
        throw new InitializationException(
            "could not configure XMLSerializer", e);
    }

    return serializer;
}
```