Migration Guide

Migration from DSAPI v. 1 to DSAPI v. 2

Creating a Stream Writer and Writing a Blob to a Stream

Blobs are written to a stream as they are written anywhere else in Tupelo. Instead of creating a StreamWriter, you need to obtain a StreamingContext, create a time-annotated resource to associate the blob with a URI and a timestamp, and then write it to Tupelo using a regular BlobWriter operator and the streaming context. E.g.

```
Contex context = ...// a regular tupelo context

//Create a context with time-annotating capabilities
DataSource dataSource =
  DataSourceFactory.getDataSource(DBType.MYSQL,"jdbc:mysql://host/db","user","password");
Context timeAnnotatingContext = new CompositeContext(context,dataSource,DBType.MYSQL);

//Create a time-annotated resource
Resource streamResource = Resource.uriRef("urn:stream");
Resource dataPointResource = TimeAnnotatedResource.create(streamResource,
      TemporalAnnotation.getInstant(System.currentTimeMillis()));

//Write the blob to Tupelo
BlobWriter bw = new BlobWriter(dataPoint);
bw.setInputStream(new FileInputStream("data000.jpg"));
context.perform(bw);
```

Creating a Stream Reader and Reading a Blob from a Stream

In a similar way, blobs are read from a StreamingContext using regular Tupelo operators such as BlobFetcher. E.g.

```
Contex context = ...// a regular tupelo context

//Create a context with time-annotating capabilities
DataSource dataSource =
  DataSourceFactory.getDataSource(DBType.MYSQL,"jdbc:mysql://host/db","user","password");
Context timeAnnotatingContext = new CompositeContext(context,dataSource,DBType.MYSQL);

//Create a time-annotated resource
Resource streamResource = Resource.uriRef("urn:stream");
Resource dataPointResource = TimeAnnotatedResource.create(streamResource,
                         TemporalAnnotation.getLast()));

//fetch the data using a BlobFetcher
BlobFetcher bf = new BlobFetcher();
bf.setSubject(dataPointResource);
context.perform(bf);
//read the data and do something with it
processBlob(bf.getInputStream());
```