

Description of the Format of the Network Files

Thomas Brinkhoff, IAPG, FH Oldenburg/Ostfriesland/Wilhelmshaven

<http://www.fh-ooow.de/institute/iapg/personen/brinkhoff/>

March 2000 / September 2003

A network for generating network-based spatiotemporal datasets is described by two binary files. The first file describes the nodes; its name ends with `.node`. The second file (`*.edge`) consists of the edges.

Format of the node file

The file consists of a set of nodes. Each node consists of

- the length of the node name (0-127) (java byte)
- the name as byte array
- the identifier of the node (java long)
- the x-coordinate of the node (java int)
- the y-coordinate of the node (java int)

In the following, you'll find a code fragment writing such nodes to a file:

```
public boolean writeNode (java.io.DataOutputStream out, String name,
                        long id, int x, int y) {
    try {
        byte l = (byte)name().length();
        out.writeByte(l);
        if (l > 0)
            out.write(name.getBytes());
        out.writeLong(id);
        out.writeInt(x);
        out.writeInt(y);
        return true;
    } catch (IOException ex) {
        return false;
    }
}
```

Format of the edge file

The file consists of a set of edges. Each edge consists of

- the identifier of the first node (java long)
- the identifier of the second node (java long)
- the length of the name of the edge (0-127) (java byte)
- the name of the edge as byte array
- the identifier of the edge (java long)
- the class of the edge (java int)

In the following, you'll find a code fragment writing such edge to a file:

```
public boolean writeEdge (java.io.DataOutputStream out, long nodeId1,
                        long nodeId2, String name, long id, int edgeClass) {
    try {
        out.writeLong(nodeId1);
        out.writeLong(nodeId2);
        byte l = (byte)name.length();
        out.writeByte(l);
        if (l > 0)
            out.write(name.getBytes());
        out.writeLong(id);
        out.writeInt(edgeClass);
        return true;
    } catch (IOException ex) {
        return false;
    }
}
```

