Java - Logical_Topology_Algorithm/src/Logical_Topology_Algorithm.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | Java

Package Explorer

- Grooming_minimizeWaveleng
- Grooming_shortestPath
- Grooming_Solver
- Grooming1_1
- lineMatrix
- Logical_Topology_Algorithm
- nodeInformation
- Optical_Network_addSecondar
- Optical_Network_joinTrafficMa
- Optical_Network_networkCost
- Optical_Network_shortestPath
- Optical_Network_shortestPath_
- Optical_Network_shortestPath_
- Optical_Network_shortestPath_
- Report_networkDesignModifie
- saveDataToFile
- simplifiedReport

Logical_Topology_Algorithm.java

```
1  /*************************************************************************
11
12 import java.util.ArrayList;
24
25 /**
29 public class Logical_Topology_Algorithm implements IAlgorithm
30 {
31     @Override
32     public String executeAlgorithm(NetPlan netPlan, Map<String, String> algorithmParameters, Map<String, String> net2planParameters)
33     {
34
35         final int N = netPlan.getNumberOfNodes();
36         if (N == 0 ) throw new Net2PlanException("This algorithm requires a topology with nodes");
37
38         int nodes[] = new int[N];
39
40         //Validate arguments
41         netPlan.setLayerName(0, "Physical Topology");
42         String logicalTopology = algorithmParameters.get("logicalTopology");
43         if (!logicalTopology.equalsIgnoreCase("Opaque") && !logicalTopology.equalsIgnoreCase("Transparent")
44             && !logicalTopology.equalsIgnoreCase("Translucent"))
45         {
46             throw new Net2PlanException("'logicalTopology' must be 'Opaque' , 'Transparent' or 'Translucent'");
47         }
48
49         Set<Long> nodeIds = netPlan.getNodeIds();
50         Map<Long, Double> linkCosts = netPlan.getLinkLengthInKmMap(0);
51         Map<Long, Pair<Long, Long>> linkTable = netPlan.getLinkMap(0);
52
53         Set<Long> demandIds = netPlan.getDemandIds(0);
54
55         //Check if there is no second layer
56         //Opaque physical layer = logical layer
57         //Add new layer logical topology transparent
58         boolean active = netPlan.isLayerActive(1);
59         if(active!=true)
60         {
61
62         if (logicalTopology == "Opaque")
63             {
64                 netPlan.copyLayer(0);
65             }
```

Problems | Javadoc | Declaration

0 items

Description	Resource	Path	Location	Type

Writable | Smart Insert | 1 : 81