PNNL-17614

Pacific Northwest
National Laboratory
Operated by Battelle for the
U.S. Department of Energy

GridLAB-D Technical Support Document: Tape Modules Version 1.0

DP Chassin

May 2008

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm


This document was printed on recycled paper.
(9/2003)

PNNL-17614

GridLAB-D Technical Support Document: Tape Modules Version 1.0

DP Chassin

May 2008

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Acronyms and Abbreviations

CSV	comma-separated values
ODBC	open database connectivity

Contents

1.0 Introduction

A tape module implements objects that can be used to establish and change the boundary condition on a model, and observes the properties of individual objects or the aggregate properties of a group of objects. Player and shaper tapes are used for updating the model at specified times from a file. Recorder and collector tapes are used for collecting information from the model.

2.0 Tape Modules

Tape modules consist of players, shapers, recorders, and collectors.

2.1 Players

A player provides the ability to update a single object variable at specified times. The values are read from a file formatted like comma-separated value (CSV) file or other source (for example, open database connectivity [ODBC] or Matlab). The source data must have timestamps (or time changes) in the first column, and the values to be posted in the second column.

In defining a player in a model input file, the variable to which the value is written must be specified. The variable to be updated must exist in the player's parent, which must also be specified in the input model. A loop count can also be specified that will allow the source to be played more than once. For example, the following lines in a model input file will use the player in a file named lightingDemand.txt to update the *demand* variable in the *lights* object.

```
object player {
   name player;
   parent lights;
   property demand;
   file lighting.player;
   loop 1000;
}
```

2.2 Shapers

A shaper provides the ability to update a single variable for a group of objects at specified times. Shapers produce boundary conditions based on a shape, either by playing a scaled result that conforms to the defined shape, or producing a series of pulse-width modulated events of a set amplitude that aggregate over time to the given shape. Shapers differ from other tapes in that they must define the conditions that give rise to the shape.

2.3 Recorders

A recorder provides the ability to collect a recording of one of more properties of an object. It can specify the sampling interval, triggers, and other properties affecting the recording. For example, the following lines in a model input file will record the values of the *energy* and *power* variables in the *meter* object to a file called meter.csv at 3600-second intervals.

```
object recorder{
   name MeterCorder;
   parent meter;
   property energy,power;
   file meter.csv;
   interval 3600;
   limit 1000;
}
```

2.4 Collectors

Collectors are different from recorders in that they aggregate multiple object properties into a single value. They do not use the parent property but instead use the group property to form a collection of objects over which the aggregate is taken. The group property specifies the grouping rule for creating the collection. Groups may be specified using any registered property of the object, such as class, size, parent, id, or rank. The property value is aggregated as a minimum, maximum, count, average, standard deviation, mean, variance (2^{nd} moment), mean bias error (1^{st} moment), or kurtosis (3^{rd} moment). If the property is a complex number, the property must be specified in the form

```
   property.part
```

where part is real, imaginary, magnitude, angle (in degrees), or angular component (in radians).

For example, a collector over all water heater objects might aggregate the power property using “count(power),min(power),max(power),std(power)”, which would print the number of water heaters, the minimum power used by any one water heater, the maximum power used, and the standard deviation of the power used by the set of water heaters.