NAME

getPolicyFileInfo − Read information from the header of a policy file

SYNOPSIS

#include <qnet.h>

int getPolicyFileInfo(string *path***, int** *maxdim***, int*** *ndim***, int*** *decn***, int** *trunc[]***);**

DESCRIPTION

The **getPolicyFileInfo**() function reads the header information from the file whose name is *path* and that is expected to contain policy information for a state space that has dimension less than or equal to *maxdim*.

Upon return, *ndim* contains the dimension of the state space (the number of classes or queues) and *decn* contains the number of decisions that are specified at each state. The state space truncation values are returned in the array *trunc[]*, which should contain space for at least *maxdim* elements.

RETURN VALUE

On success, the value **QNET_NORMAL** (defined to be zero) is returned, otherwise one of the following negative values is returned:

QNET_BAD_MAGIC
The input file is not recognized as a policy data file.

QNET_BAD_DIMENSION
The state space dimension stored in the file exceeds *maxdim* or *maxdim* is negative.

QNET_BAD_FILE
The input file could not be opend or read from.

EXAMPLE

The following sequence can be used to read the header information from a policy file:

```
int maxdim = 3;    // largest state space dimension expected
int ndim;          // actual dimension of state space
int decn;          // number of decisions
int N[maxdim + 1]; // state space truncations

getPolicyFileInfo(string("policy.u"), maxdim, &ndim, &decn, &N[1]);
```

Here N[0] is not used; this follows the convention used in most of the QNET DP code. Once the header information is available an array to hold the policy data can be allocated and the data read in. If *ndim*=3 and *decn*=2 then the code to do this would be

```
int**** action;
action = (int****) makePolicyArray(ndim, decn, &N[1]);
readPolicyFile(string("policy.u"), ndim, decn, &N[1], action);
```

SEE ALSO

readPolicyFile(3), **writePolicyFile**(3), **makePolicyArray**(3)

AUTHOR