HORSE ID BAYESIAN BELIEF NETWORK MODEL MANUAL

To query the HorseID BBN Server, use **curl** or make **REST API** request as follows:

curl -X < environment > {url_address} / < object > / < function > --data < request >

Where { url_address } may be a constant url address of the HorseID BBN server. In this case, url_address is http://bbn.horseid.com.

The code format above is equivalent to the Python code format:

< environment_class >.< function > (< request >)

Where **< environment >** option is the development environment's variable in curl and it takes any value of :

GET	For Testing	for testing the function without a test case in testing env.
PUT	For Usage	for using function with a request in production or dev env.
POST	For Testing	for testing the function with a desired test case in test env.

The < **environment_class** > option is the development environment's variable in python. which can take any value of :

HorseIDBayesianNetwork	for normal use.
HorseIDBayesianNetworkTest	for testing.

The **< object >** option can be any of:

variables
model (We will focus on model for now)

The **< function > or < activities >** option is a variable that shows the function to call. These can be any of :

build	use_default_values	get_cpds	load_data	check_model
run	declare_variables	load_cpds	prepare_data	get_cardinality
update	update_values	draw_default_graph	train_model	get_local_independencies
initialise_space	load_sizes	draw_graph	update_model	get_active_trail_nodes
set_universe	set_evidences	build_model	test_model	query
clear_values	set_cpds	load_cpd_to_model	describe_node	map_query

The General < **request** > Object Format:

```
request =
{
        'data':
        {
```

```
                #for all nodes, do
                        'node_name': [ value1, value2, value3, ... ],
        },
        'dataset':
        {
                #for all nodes, do
                        'node_name': [ value1, value2, value3, ... ],
        },
        'graph':
        [
                #for all edges, do
                        ('from node_name', 'to node_name')
        ],
        'node'          :                       'value',
        'variables'     :                       [‘valuesl’]
        'observed'      :                       'values',
        'evidence:
        {
                #for all nodes, do
                        node.node_name :    (‘evidence_value’),
        },
        evidence_card:
        {
                #for all nodes, do
                        node.node_name :    'evidence_card_value',
        }
        'elimination_order':    [values]
}
```

Explain Variable Format ASAP!!!

For Example:

To call function build (None) on the Horse Identification BBN model, that is

HorseIDBayesianModel.build({‘node’:‘value’}) in python is equivalent to

curl -X **PUT** http://bbn.horseid.com/**model/build** --data **{ ‘node’ : ‘value’ }**

and to test if it works (for admin use only) use the REST request :

curl -X **GET** http://bbn.horseid.com/**model/build** for quick check. For more detailed testing with a test case such as { ‘node’ : ‘chip_work’, ‘result’: True } use the REST request:

curl -X **POST** http://locahost:8000/**model/build** --data **{ ‘node’ : ‘chip_work’, ‘result’: True }**

General work flow in the development and production environment is given by:

1. START THE SYSTEM

    Python Code:
    ```
    from bbn import HorseIDBayesianNetwork
    model = HorseIDBayesianNetwork( );
    model.start( );
    ```

    Curl/REST API code:
    ```
    curl -X PUT http://bbn.horseid.com/model/start  --data { }
    ```

2. USE SYSTEM

    Python Code:
    ```
    model.set_graph( request );
    ```

    Curl/REST API code:
    ```
    curl -X PUT http://bbn.horseid.com/model/set_graph --data $request
    ```

NOTE: Step 1 and Step 2 are very import to start up the BBN system. All other activities are done in Step 3.

Hence, we have the general procedure is as follows:
Python code:
```
from bbn import HorseIDBayesianNetwork;
model = HorseIDBayesianNetwork

#start up the system ADMIN ACCESS ONLY
request=None
model.start(request)

#feel free to call any function here PUBLIC ACCESS
request = {…}
model.query(request);
```
Curl/REST API code:
```
#set up the system ADMIN ACCESS ONLY
curl -X PUT http://bbn.horseid.com/model/start  --data { }
#feel free to call any function here PUBLIC ACCESS
request={ }
curl -X PUT http://bbn.horseid.com/model/query --data $request
```