



Multi-file support:

Our goal was to achieve multi-file support for the Data Tag Project.

We thought about lots of ideas in order to do so as well as we could.

At first we thought about merging object implicitly but due to lots of examples of other developers who tried to implement multi-file support this way we understood we better come up with a better idea.

Assumptions that were taken into consideration while trying to find a more suitable solution:

1. We did not want the support of multi-file system to change the logic of the decision tree.

In order to achieve this goal we added two new input options:

- a. A file named Root.slot that will contain the name of all wanted files.

For an example:

cat Root.slot →

Animals, Food_v1_beta-15, CarsVersionOne

- b. A direct input into command line

For an example:

CliRunner ... Dsektop/Animals.ts RootSlot=Animals, Food_v1_beta-15, CarsVersionOne ...

Now while parsing process the tree will contain a root leaf that will be used as a father to all other files defined by a wanted name.

Doing so allowed us to achieve another goal – by changing the Root Slot one can manipulate all wanted files in different combinations without the need to change any of the *.ts files.

2. We did not want any assumptions of the uniqueness of any name that was given to any *.ts file.

Trying to achieve a unique id we understood file path is one-one correspondence.

Having this fact minded we knew file names were not an option to define the subject that is handle in a specified file.

Thus we decided to add a new keyword to Data Tags Project – “exports”.

Syntax of *.ts file was changed to:

cat Animals.ts→

exports Animal[comment 1]: consists of Family, Cover, Food_v1_beta-15/Type.

Family: one of insect, mammal.

Cover: some of skin, scaels, fur.

```
cat Food_v1_beta-15.ts→  
exports Food[comment 2]...
```

```
cat Food_v2_beta-15.ts→  
exports Food[comment 3]...
```

Now subject can be defined with any name and file and subject can be named without the need of uniqueness (paragraph 3 holds the explanation to how both subjects will differ in runtime).

3. Allowing usage of different tags in two different files:

We wanted the system will have an option of “importing” tags between two files.

By the definitions:

```
cat Animals.ts →  
exports Animal[lalala]: consists of Family, Cover, Food_v1_beta-15/Type.  
Family: one of insect, mammal;  
Cover: some of skin, scaels, fur.
```

```
cat Food_v1_beta-15.ts →  
exports Food[aflaskdjflks]...  
Family: some of protein, carbonite, fiber.  
Type: one of cereal, soup, mush, chunks.
```

Adding a reference of file name to compound tags allowed us to do so:

```
Animals.ts →  
exports Animal[lalala]: consists of Family, Cover,/ Food_v1_beta-15/Type.  
/Animals/Family: one of insect, mammal;  
/Animals/Cover: some of skin, scaels, fur.  
Food_v1_beta-15/Type: one of cereal, soup, mush, chunks.
```

4. Combining all of these approaches together gave us the ability to get id to any input file – the path and define a one and only Decision Graph.

Sincerely,

Roal Fitoussi and Tomer Chechik