Tags and Queries

Tag and Query functions are in system/support/tagquery.lua. They build upon the functions of the core.

void :tag(tags<string>)

This tags, or detags, the related object with tags (it is a helper function for _setTagState). The parameter is a collection of tag changes separated by commas. There are two tag changes - adding tags, where either +name or just name is used, and removing tags where -name is used. Tag names must be alphanumeric strings.

<instance table>,<count>:query(tags<string>)

This queries the tags. The parameter is a list of one or more tags which must all be set to have the instance in the output list. The function returns a hash where the references are keys and a count like the simple functions in core.lua. If an empty string or nil parameter is provided it returns all objects (e.g. it is functionally equivalent to __getAllObjects).

Queries with zero or one parameter are fairly rapid, those with two or more may be significantly slower as they are O(n^2) so they should not be used in speed scenarios; if this is absolutely required, have a single tag representing the pairing.

As with other functions of this type, it should be assumed that the return table is read only, as it may be a direct reference to an internal table.

<string>,<string>__extract(str<string>) private

This is a helper function for the above two functions. Its purpose is to break down a string separated by commas - effectively it is like the Python split() method with space removal.

Two parameters are returned. The first is the remainder of the string; this returns nil if there is nothing left in the string (e.g. the string was originally empty or only had one entry in it), otherwise it returns the rest of the string. The second is the removed string; this may too be nil if there is nothing to remove.

Effectively, given *“ hello, there , world”* it would return *“there , world”* for the first string and *“hello”* as the second.

First version 25 Nov 2014