Subject: Re: Esc: maps nested in arrays and vice versa - underdevelopment or a bug?
Posted by fudadmin on Fri, 12 May 2006 22:23:17 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 12 May 2006 22:54fudadmin wrote on Fri, 12 May 2006 17:36Quote:
This is not multimap and they all have the same key value - void...

I had been using these kind of structures for 5 years with Dialect interpreter but I have never imagined that someone will try to invent square wheels...

Actually, this is the same as maps in any other scripting language. (Now I wonder how look Dialect maps like

...
Quote:
Frames

Frames store data by equating a string, which becomes the frame’s localized key, to a value. Thus a frame is essentially a storage system of key-value pairs, often called "hash tables" or "dictionaries". A frame key must be a string (ex. x or myVariable), while the value can be of any data type. Frames are created using curly braces, {}. To set the variable x to an empty frame: x = {}. To create a frame with two initial key-value pairs, also known as slots, separate the values by commas (ex. myFrame = {age:27, eyeColor:"Brown"}, or, myFrame = {"age":27, "eyeColor":"Brown"}). Values are retrieved from a frame using either the dot or bracket operator. Thus both

x = myFrame.age and x = myFrame["age"] yield the value 27.

If a key is referenced that doesn't contain a value, the result will be nil. Once a frame has been created, new slots can be added using the same operators: myFrame.hairColor = "Blonde", or, myFrame["hairColor"] = "Blonde". To completely remove a slot fro m a frame, use the remove function. For example: remove(myFrame, "hairColor")

Two frames can be combined using the concatenation operator. If a key exists in both frames, then the value contained in the right operand is used in the result. For example: {a:1, b:2} ~ {b:3, c:4} results in the frame {a:1, b:3, c:4}.

When evaluated as a Boolean value, all frames return true.