Subject: Re: Esc: maps nested in arrays and vice versa - underdevelopment or a bug?
Posted by mirek on Fri, 12 May 2006 21:54:49 GMT
View Forum Message <> Reply to Message

fudadmin 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:
Nevertheless, are these structures both maps or none, or only the second one?

menus1={ menu :"MainMenu", submenus :"MainSubmenu1" };
menus2={ "menu":"MainMenu", "submenus":"MainSubmenu1" };

How many keys and values are in each of them?

Sure, if they are initialize as { x:y, ... }, they are maps.

Quote:
And does it mean that you can have only one key in a map???

You can have one value per one key. In other words, each key has unique value.

In this example, for menus1, value of menu and submenus is void. Therefore, your initiazation could also be written as

menus1[void] = "MainMenu";
menus1[void] = "MainSubmenu1";

menus2["menu"] = "MainMenu";
menus2["submenus"] = "MainSubmenu1";

I hope this helps...

BTW, there is also other equivalent "structure" notation for menus2:

menus2.menu = "MainMenu";

menus2.submenu = "MainSubmenu1";

Quote:
And otherwise you have to create a multimap(what's this?)?

Well, I used STL termnilogy here. std::map is map like this (unique keys). There is also std::multimap - in that case, keys are not unique - you can have more keys with the same value in the map (but then single operator[] is obviously not enough for dealing with such ADT).

Mirek
