Subject: Re: NEW: Tree<T> container
Posted by kohait00 on Fri, 30 Jul 2010 14:59:31 GMT
View Forum Message <> Reply to Message

i'looked inside your code and borrowed the GetRoot idea
i personally prefer the Node storage beeing pure Array, without the way complicated next/prev stuff of linked lists. if i really need to iterate through them, i could GetParent and access all children (todo: know one's own position

now there is 2 flavors of the hole thing.

Tree<T>, beeing organized through a Node<T> as root (where T can easyly be One<T> if needed). making direct deriving of the object unneeded, but still possible if one wants to one can setup own Node with the needed elements directly stored.

Branch<T>, beeing organized directly from a root instance of Branch<T>. having a T instance inside. but this reduces the flexibility a bit (in terms of pick_ Add()'s to the Array, compared to Tree<T>, where T *are* the elements of Tree, here, the Branch only has a T memeber. so Branch is actually a subset of Tree and could be esiely done with Tree<T>. i'll provide another example soon.

thanks for help. if that all works, the algorithms are to be implemented, binary, binary balanced, ...

File Attachments

1) TreeTest.rar, downloaded 226 times