Array access - Property vs field vs indexer

This benchmark shows the relative speed of a large property array vs field array vs indexer array. The benchmark uses this "timer" macro.

The benchmark has since been updated to include tests of lists as well as rawArrayIndexing.

It now shows there is little difference between accessing array fields or properties or indexed members, but you can get a dramatic speedup if you use the rawArrayIndexing macro. However, rawArrayIndexing does not speed up accessing members of a list because the Boo.Lang.List class normalizes the index value on its own.

```
[System.Reflection.DefaultMember("h")]
class Y:
 private _size = 10000000
 public _a as (int) = array(int, _size)
 public a as (int):
  get:
   return _a
 h(index as int):
  get:
   rawArrayIndexing:
    return _a[index]
  set:
   rawArrayIndexing:
    _a[index] = value

 public _l as List

x = Y()
x._a[0] = 0
print "filling the list, please wait..."
x._l = List(x._a)
print

timer "Overall elapsed time: ":
 print "First raw array indexing is only used in the class's default
member:"
 timer "Property a:  ":
  i = 1
  while i < len(x._a):
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   ++i

 timer "Array _a:  ":
  i = 1
  while i < len(x._a):
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
  ++i

timer "List _l:  ":
 i = 1
 while i < len(x._l):
  x._l[i] = x._l[i-1]
  x._l[i] = x._l[i-1]
  x._l[i] = x._l[i-1]
  x._l[i] = x._l[i-1]
  x._l[i] = x._l[i-1]
  ++i

timer "Default member h: ":
 i = 1
 while i < len(x._a):
  x[i] = x[i-1]
  x[i] = x[i-1]
  x[i] = x[i-1]
  x[i] = x[i-1]
  x[i] = x[i-1]
  ++i

print
print "Now raw array indexing when accessing the class members too:"

timer "Property a:  ":
 rawArrayIndexing:
  i = 1
  while i < len(x._a):
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   x.a[i] = x.a[i-1]
   ++i

timer "Array _a:  ":
 rawArrayIndexing:
  i = 1
  while i < len(x._a):
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   x._a[i] = x._a[i-1]
   ++i

timer "List _l:  ":
 rawArrayIndexing:
  i = 1
  while i < len(x._l):
   x._l[i] = x._l[i-1]
   x._l[i] = x._l[i-1]
```

```
     x._l[i] = x._l[i-1]
     x._l[i] = x._l[i-1]
     x._l[i] = x._l[i-1]
     ++i

  timer "Default member h: ":
   rawArrayIndexing:
    i = 1
    while i < len(x._a):
     x[i] = x[i-1]
     x[i] = x[i-1]
     x[i] = x[i-1]
     x[i] = x[i-1]
     x[i] = x[i-1]
```

```
      ++i
   print
```

The output I got: (I compiled to exe first and then ran it)

```
First raw array indexing is only used in the class's default member:
Property a:  00:00:02.2187500
Array _a:  00:00:02.2031250
List _l:  00:00:02.4531250
Default member h: 00:00:00.1250000

Now raw array indexing when accessing the class members too:
Property a:  00:00:00.1875000
Array _a:  00:00:00.1406250
List _l:  00:00:02.4218750
Default member h: 00:00:00.1250000

Overall elapsed time: 00:00:09.8750000
```