Why Not To Use Vector Class In Your Code?

Vector class is often considered as obsolete or “Due for Deprecation” by many experienced Java developers.

They always recommend and advise not to use Vector class in your code.

They prefer using ArrayList over Vector class.

I have tried to list out some points regarding why not to use Vector class in your code.

You can achieve Thread Safety without Vector.

Vector class has only one advantage over ArrayList i.e it is thread safety.

But, you can achieve thread safe ArrayList by using synchronizedList() method of Collections class.

```
 public class Test
{

        public static void main(String[] args)
           {

             ArrayList<Integer> list = new ArrayList<Integer>();

              Collections.synchronizedList(list);

           //It returns Synchronized list backed by original list.

           }
}
```

Thread Safe of Vector class is time consuming.

All methods of Vector class are synchronized. This makes each and every operation on Vector object thread safe. But, it is time consuming.

Because, you need to acquire object lock for each operation you want to perform on vector object. Usually, you need set of operations to be synchronized not each and every operation.

It Doesn't make sense to take the object lock once, perform the operations you want and then release the lock when you are done. Why acquire the lock again and again for each operations?.

This is the time consuming process and decreases the performance of your application.

Enumeration Vs Iterator

Vector class has a method which return Enumeration over the elements of Vector object. Although, Enumerations are faster than the Iterator, but it is not backed by the original collection.
That means, any changes made to the original collection does not reflect in Enumeration object. They ignore the modifications done during iteration. This may cause issues.

Is Vector class poorly designed?

Vector class combines two features – “Re-sizable Array” and “Synchronization“.

This makes poor design. Because, if you need just “Re-sizable Array” and you use Vector class for that, you will get “synchronized Resizable Array” not just re-sizable array.

This may reduce the performance of your application. Therefore, instead of using Vector class, always use ArrayList class.

You will have re-sizable array and whenever you want to make it synchronized, use

Collections.SynchronizedList().