OpenEye Scientific Software

Best Practices for Java

Online documentation: http://www.eyesopen.com/docs

How to avoid creating garbage: Reuse molecules and call *Clear()*.

```
// Re-using molecule objects makes less
// garbage
OEMol mol = new OEMol();
for(int i=0; i!=10; i++) {
    oechem.OESmilesToMol(mol, smiles[i]);
    ...
    mol.Clear();
}
```

```
// Creating a new molecule each iteration
// makes more garbage
for(int i=0; i!=10; i++) {
    OEMol mol = new OEMol();
    oechem.OESmilesToMol(mol, smiles[i]);
    ...
}
```

When to call close()

Call *close()* on any object that has the method, when it is done being used. Despite the fact that java is garbage collected, calling close on any object with an underlying file handle is necessary. The reason for this is that garbage collector (GC) is non-deterministic. If the GC were to handle closing files then the stream might be waiting to flush for several GC cycles, causing corruption if the same file were written to again by another object.

When to call delete()

Call *delete()* as soon as possible on objects that have it defined from these packages:

oegrid oezap oespicoli oeszybki oeshape

The reason the *delete()* method is provided is due to memory management incompatibilities with the underlying C source code and the JVM.

OEFloatArray vs float[]

Often it is faster to use an OEFloatArray when the contents do not need to be manipulated between OpenEye API calls. This is best explained with the following examples:

```
// fastest
OEFloatArray fa = new OEFloatArray(...);
mol.GetCoords(conf, fa);
mol.SetCoords(conf, fa);
```

```
// fast
float[] a = new float[...];
mol.GetCoords(conf, a);
mol.SetCoords(conf, a);
```

```
// slow
float[] a = new float[...];
mol.GetCoords(conf, a);
for (int i = 0; i != a.length; i++)
    a[i] += 1.0f;
mol.SetCoords(conf, a);
```

```
// slowest
OEFloatArray fa = new OEFloatArray(...);
mol.GetCoords(conf, fa);
for (int i = 0; i != fa.getLen(); i++)
    fa.setItem(i, fa.getItem(i) + 1.0f));
mol.SetCoords(conf, fa);
```

OpenEye defines

-Doejava.libs.debug=[1|0]	Display debugging output to the console while loading shared libraries.
-Doejava.libs.path=*path*	Load OE shared libraries directly from *path*. Makes no attempt to unpack them from the jar and fails if no libraries are found.

Tips

Prefer *oechem.OESmilesToMol()* to *oechem.OEParseSmiles()*.
Prefer *float[]* to *double[]* when there is a choice.
Calling any method on an object after .delete() will cause a hard crash.

Keep in mind

"You can be sure only about which objects are marked for garbage collection. You can never be sure exactly when the object will be garbage collected." [1]

Garbage collection happens in a low priority thread. [1]

"... the cost of automatic dynamic memory management is highly dependent on application behavior." [2]

References

1. OCA Java SE 7 Programmer I Certification Guide. Mala Gupta. 2013. Pages 120-124. Manning Publications Company. ISBN: 9781617291043

2. The Garbage Collection HandBook. Richard Jones, Antony Hosking, Eliot Moss. 2012. Page 9. CRC Press

Support

OpenEye Scientific Software, Inc.
9 Bisbee Ct. Suite D
Santa Fe, NM 87508
www.eyesopen.com
support@eyesopen.com