

# Quels objets groupés par classes occupent de la mémoire

```
jmap -histo <pid>
```

Exemple de résultat :

num	#instances	#bytes	class name (module)
1:	1362240	282208168	[B (java.base@11.0.5)
2:	951832	105719672	[C (java.base@11.0.5)
3:	1062742	50457440	[Ljava.lang.String; (java.base@11.0.5)
4:	31119	34863792	[I (java.base@11.0.5)
5:	1070615	34259680	org.apache.cocoon.xml.SaxBuffer\$StartElement
6:	1296689	31120536	java.lang.String (java.base@11.0.5)
7:	214401	28913920	[Ljava.lang.Object; (java.base@11.0.5)
8:	1070740	25697760	org.xml.sax.helpers.AttributesImpl (java.xml@11.0.5)
9:	1070615	25694760	org.apache.cocoon.xml.SaxBuffer\$EndElement
10:	578713	18518816	java.util.HashMap\$Node (java.base@11.0.5)
11:	916100	14657600	org.apache.cocoon.xml.SaxBuffer\$Characters

Pour voir l'évolution :

```
jmap -histo 1234 > jmap-histo1.txt
# bcp plus tard
jmap -histo 1234 > jmap-histo2.txt
./compare-jmap-histo jmap-histo1.txt jmap-histo2.txt
```

avec le script

## compare-jmap-histo

```
#!/usr/bin/perl

foreach (`cat $ARGV[0]`) {
    my ($instances, $sizes, $name) = /^s*\d+:\s+(\d+)\s+(\d+)\s+(.*)/ or next;
    $prev{$name} = { sizes => $sizes, instances => $instances } if !$prev{$name}
}
foreach (`cat $ARGV[1]`) {
    my ($instances, $sizes, $name) = /^s*\d+:\s+(\d+)\s+(\d+)\s+(.*)/ or next;
    my $prev = delete $prev{$name} or print("unknown prev $name\n"), next;
    printf "%4dM =>%4dM\t\t%7d => %7d\t%s\n", $prev->{sizes} / 1e6, $sizes / 1e6, $prev->{instances},
$instances, $name;
}
```