Do it in an Inspector!

Never underestimate the power of inspecting and changing things with Inspectors. Perhaps your GemStone database is messed up: a bunch of People have excess Parts in certain Bins.

Grab an Inspector on some root object you can get a hold of. Send it a message, #people, and Inspect It. In the Inspector on the people, type an expression and Inspect It:

```
self select:
    [ :each | | found |
    found := each bins
        detect: [ :eachBin | eachBin size > 1]
        ifNone: [nil].
    found notNil]
```

In the Inspector on the collection of affected people, write a loop to fix each one:

```
self do:
    [ :each |
    (each bins select: [ :eachBin | eachBin size > 1]) do:
        [ :eachBad |
        (eachBad parts copyFrom: 2 to: eachBad parts size) do:
            [ :eachPart | eachBad removePart: eachPart]]]
```

The problem is resolved. Close your inspectors and carry on with your day.

Note: the code example above is not correctly formatted by our standards. In a couple of inspectors it is hard to get to well-factored code. Furthermore, what you have done to the system will be documented nowhere. These are rather troubling issues: what do they suggest to you?

[Home] [XP] [XP Practices Frame]

ronjeffries@acm.org
http://www.armaties.com