There are some changes in Version 1.04 that are not backward compatible with previous versions.

1 Linear Constraints

The most serious backward compatibility issue is that the default value of lin.abs has changed from FALSE to TRUE. The reason for this is the addition of the linear styles like "varfraction" and "varvalue". There may be less confusion thinking of these constraints being absolute rather than net (but really it is best to think of them as outside that dichotomy).

There is a warning in 1.04 when there is the possibility of this change impacting the results. This cure could become worse than the disease—the easiest way to get rid of the warning is to specify lin.abs in the call.

The results of an optimization now include information on the linear constraints in a different way, though this is unlikely to be a compatibility problem. There is now a lin.table component in the results of trade.optimizer if there are linear constraints. An example is:

```
$lin.table
  style trade absolute levels direction rfloc riskfrac.col
Country  varfraction FALSE  TRUE 3 0 0 0 1
Sector  varfraction FALSE  TRUE 4 0 0 0 1
Country@2 weight FALSE   TRUE 3 0 0 0 1
Sector@2 weight FALSE   TRUE 4 0 0 0 1
```

So there is a succinct summary of the linear constraints imposed.

2 Valuation

The computed value for cash in valuation has changed. The new calculation is the gross value minus the net value. Previously it was just the gross value.
That was a bug in the sense that it didn’t do as intended except for special
cases; it was not a bug in the sense that it did what it was documented to do.

The default value of the `collapse` argument for the random portfolio method
of `valuation` used to always be `FALSE`. Now it is `TRUE` if prices is a matrix.

Not really a backward compatibility problem, but the default and optimized
portfolio methods of `valuation` didn’t respect `weight=TRUE` when `collapse`
was `TRUE`. The default value of `weight` has changed in this setting, but the
default behavior remains the same.

3 Trading Costs

In long-short portfolios there is a complication with trading costs when a trade
turns a long position into a short position or vice versa. The complication is
that the cost on the long-side and the short-side are allowed to be different.
Previously any trade that went past zero switched trading costs – as if there
were two separate trades: one to get to zero, and the other to go the rest of the
way.

That was too complicated, and arguably wrong.

The new scheme is:

- For linear costs (vector or single column costs and no `cost.par`) the switch
  is made as before.

- For all other cases there is just one cost evaluation done with the cost
  function for the side where the trade starts.

This change means that the `doubleconst` control argument is no longer used.