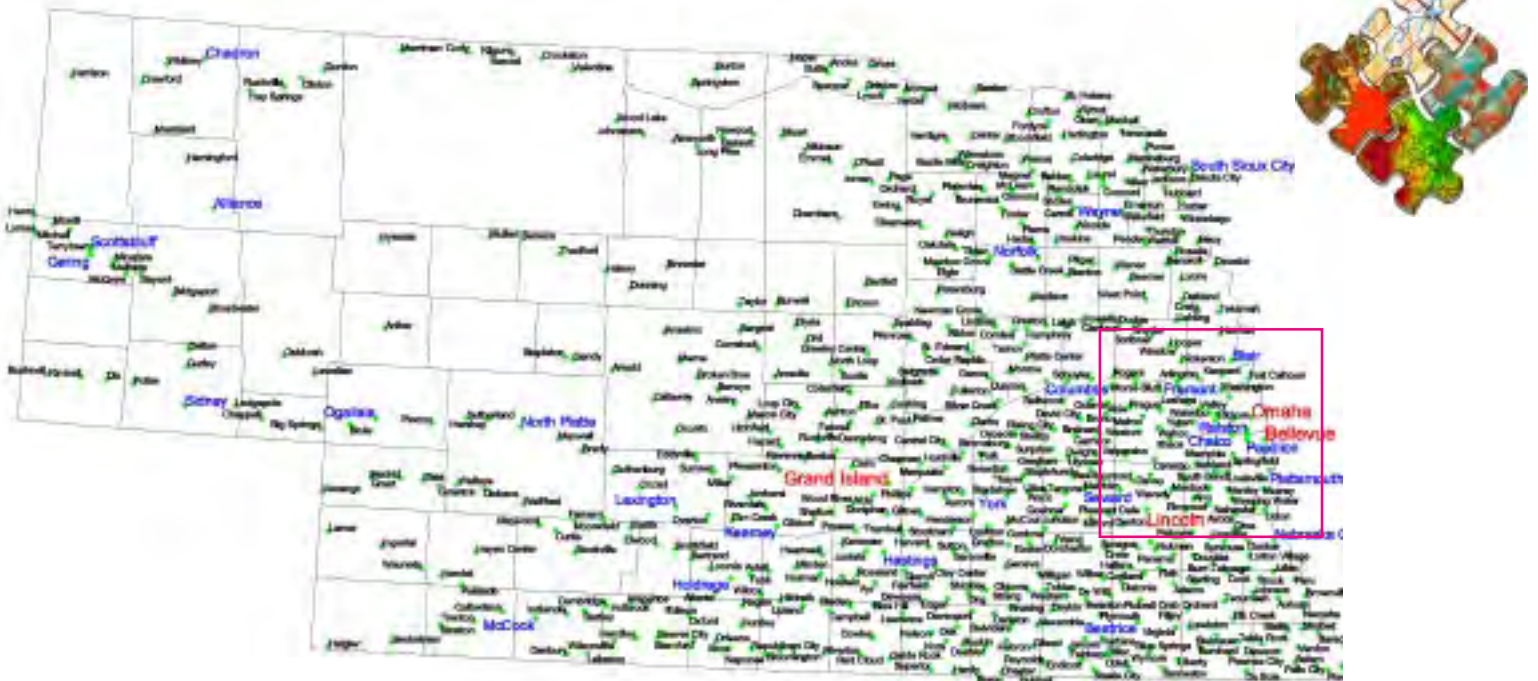# Label Placement Optimization with TNTmips®

Labels are selected for placement using a ranking system (in this case, based on city population) that allows larger cities to be selected for labeling in preference to smaller when overlap occurs (note: also allows other differentiation, such as font and color, based on city size). Label position is adjusted for fewest collisions.
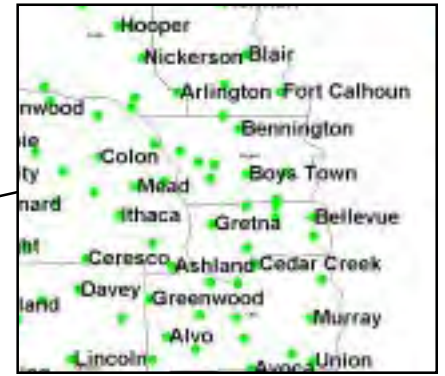


# Automated Label Placement Using TNTmips® and MapInfo®

## MapInfo just can't compare!

TNTmips' optimization and ranking prevent significant labels from being deleted and allow for greater density of labeling.

You can see it with your own eyes; the proof is in the numbers.
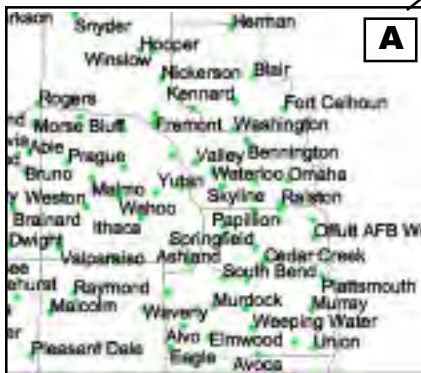


0 deleted statewide

281 deleted statewide

unenhanced labeling
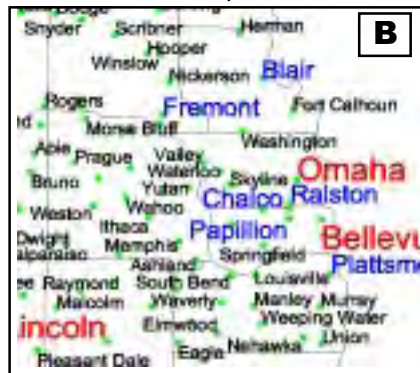
MapInfo's deletion

52 deleted statewide
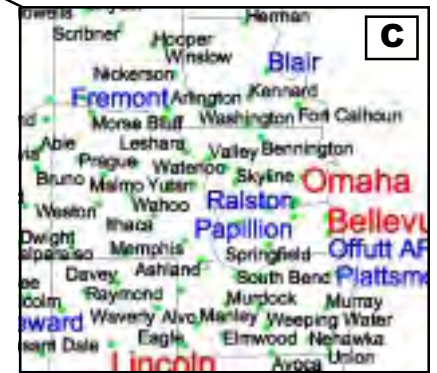
56 deleted statewide

14 deleted statewide



**A**



**B**



**C**

TNTmips using basic deletion

TNTmips using ranked labels and deletion

TNTmips full optimization and ranking, method used above

---

# Automated label placement using cartographic scripts in TNTmips®

## Overlapping and unreadable labels?
## Tedious manual placement?  Not anymore!

Automate your label placement and get even better, more consistent results using TNTmips.

### Simple Deletion

This most basic label placement management (your only option with MapInfo) deletes labels as they overlap.

### Ranked Deletion

Using simple deletion, important labels may be removed in favor of less important ones because deletion is based on the order in which labels are generated. This problem is solved by utilizing ranking, which allows overlapping labels to be selected for removal intelligently.

### Optimization

This process adjusts the positions of labels relative to the point they label in order to minimize label collision.

TNTmips' automated label placement uses deletion, ranking, and optimization to create the cleanest, densest labels without lost information or the tedium of manual placement.

### How to do it...

1. Style by query
2. Open the desired script
3. Assign size and ranking variables highlighted in red to values appropriate for your data
4. Designate a field for use in defining rank when deleting (in the example city.population is used)
5. Select a field to provide the label text (in the example city.city_name is used)

If you wish to increase the number of different ranks, simply extend the number of if-then-else loops in the script and associate rank order and values.

---

MicroImages, Inc.  •  TNTgis - Advanced Software for Geospatial Analysis
(402)477-9554 • FAX (402) 817-0151
email: info@microimages.com
Web Site: www.microimages.com

---

## Auto Label Generation Scripts

### A  Deletion of overlapping labels

```
# This script uses deletion only

dodel = 1   doopt = 0
size = 5000 #size of labels in object units
rank = 1
LineStyleSetFont("arial.ttf")
#find placement for labels
LineStyleTextNextPosition(city.city_name,size,0.0,0,
                                nextx,nexty,length)
LineStyleAddToOptimizer(Internal.x,Internal.y,Internal.x
+ length,Internal.y + size,rank,doopt,dodel)
#draw labels
func FuncDrawLabel () {
      size = small
      LineStyleSetTextColor(0,0,0,0,0,0)
      LineStyleSetFont("arial.ttf")
      LineStyleDrawText(city.city_name,size,0.0,0)
      }
```

### B  Ranked  and  C  Full Optimize

```
# This script uses deletion and ranking based on
# population to select which points are to be labeled
# and where to allow maximum density
small = 5000   mid = 7000   big = 9000
# the above values control text size at each rank
dodel = 1   doopt = 1

# For  B  - only ranked set doopt = 0

if (city.population < 5000) { #determine rank
      size = small
      rank = 1
      }
else {
      if (city.population < 25000) {
            size = mid
            rank = 2
            }
      else {
            size = big
            rank = 3
            }
      }
LineStyleSetFont("arial.ttf")
#find placement for labels
LineStyleTextNextPosition(city.city_name,size,0.0,0,
                                nextx,nexty,length)
LineStyleAddToOptimizer(Internal.x,Internal.y,Internal.x
+ length,Internal.y + size,rank,doopt,dodel)
#draw labels
func FuncDrawLabel () {
      small = 5000  mid = 7000  big = 9000
      if (city.population < 5000) {
            size = small
            LineStyleSetTextColor(0,0,0,0,0,0)
            }
      else {
            if (city.population < 25000) {
              size = mid
              LineStyleSetTextColor(0,0,255,0,0,255)
                  }
            else {
                  size = big
                  LineStyleSetTextColor(255,0,0,0,0,0)
                  }
            }
      LineStyleSetFont("arial.ttf")
      LineStyleDrawText(city.city_name,size,0.0,0)

      }
```