Constructing Image Maps

Here's how to construct an image map, such as the screen maps in Section 2.1 of requirements docs:

1. Draw the figure in idraw;
    a. Each image-map thumbnail is a scaled-down version of the corresponding full-size figure.
    b. The x/y scaling in idraw is .25/.25.
2. Open it in Photoshop at 80DPI.
3. Add a plain white background.
4. Apply the Sharpen filter to the entire file.
5. Possibly tweak the image in other ways to fix any ps-to-gif conversion weirdnesses.
    a. E.g., for the Rolodex Tool image map, some of the Helvetica Bold thumbnail titles were not properly center-aligned above the thumbnails.
    b. The titles were grabbed and horizontally moved in Photoshop to align them properly.
    c. Note that at present we elect not to do this tweakage, since it's a pain and an extra manual step in the conversion process. Such tweaks are best left to the bitter end.
6. Flatten the layers and save as a GIF.
7. Use Dreamweaver or xv (or something else) to define image map hot spots.
    a. The obvious advantage of Dreamweaver is its ease of point and click.
    b. However, in terms of total time, xv is probably as good or better, since we have to ftp and then groom the html that Dreamweaver produces.
    c. With xv, we have to do a bit of arithmetic (adding width and height to get x2 and y2), but then we can type the number straight into the image-map.me file.
8. If Dreamweaver is used, extract and legibly reformat the image map from the HTML file it creates.

Here are specific tweakages for image maps thus far generated:

1. In the Rolodex screen map, a bunch of the labels above the image thumbnails were nudged varying amounts to the left, because the center justification that appears correct (pretty much) in idraw is off when Photoshop has converts the image from ps to gif. This is unfortunate but pretty well known in the realm of ps font justification within idraw-generated images.

Here's what to do on the Postscript side:

1. When the full idraw picture is larger than a page, break it up into individual files, each fitting on a single page.
2. Name the files *x*-part*n*, for root file name *x* and 1<=*n*<= *number of pages needed*.
3. Clearly, this page break up should be done only after the full-size image map is pretty darn stable, since the parts should ***always*** be generated from the whole, to maintain consistency between the HTML and PostScript versions.