

# StrutsSellMeOnTiles

Describe the new page here.

Why should we use tiles as opposed to anything else? I am told that this question comes often, so the list below has been compiled from this mailing list archive and other published articles to consolidate the arguments for using tiles. If you have experience with tiles, please add, modify, or subtract from this list with the goal of creating a list that people can use when evaluating whether to use tiles or not on their project.

1. Inheritance. Why don't we just use the `<jsp:include/>` tag or `@include` to include common page elements? tiles definitions support inheritance. If you have 60 jsps, you have `@include header.jsp` repeated in every jsp. Next, imagine a new requirement dictates that the menu should be moved from the right side to the left. You now have to change 60 jsps. If your application uses tiles, you change only one jsp, all of your jsps inherit from this layout jsp.
2. If you have a standard site with a common layout (header, footer, menu, bodycontent), tiles does what you want to do very cleanly, integrates with Struts very nicely, and is a standard solution that you don't have to develop or document, and that your team members may already have experience with or at least be more excited about learning than an in-house, one-off system since that experience will have more value in the marketplace.
3. Using `jsp:include` or `@include` you cannot internationalize parameterized content very easily.
4. `Tiles-defs.xml` offers a central document to determine which jsps are used in which places. If the developers follow mvc properly, a new person new to the project can get a sense of site flow and structure by looking at only `struts-config.xml` and `tiles-defs.xml`
5. You can use a tiles controller rather than write a custom tag. This is especially useful for creating a tile such as Amazon's "books you've recently browsed". The tile can plug in to any page without considering the Struts action code for that URL.
6. Tiles scope offers another namespace where attribute names will not collide.