

Comparing Sakai Display Technologies

Information

This page contains info and comparisons of the various Sakai compatible display technologies.

What do we want out of a display technology?

- Easy to use for the developer
 - well structured, low learning curve, good documentation
- Easy for the designer (Code Separation)
 - easy to use for the UI designer, can change templates without changing code and vice versa
- Good integration with Sakai
 - portal, widgets, tools, etc...

What display technologies can we use in Sakai?

- Virtually any Java related display technology can work in Sakai
 - However, there are some caveats
 - Sakai uses a special Servlet to handle redirects and processing of HTTP request variables
 - Portal and widget interaction have to be developed for each technology
- What about non-Java tech?
 - Things that can connect to Java code via some kind of bridge (like Ruby or PHP) can be used but good integration is unlikely

Current Sakai Display Technologies

[ComparativeDisplayTechnologies.ppt](#) - Powerpoint on Comparative Display Technologies

Key:	 =Yes /Good	 =In middle/Sorta /Neutral	 =No /Bad	 =Positive	 =Negative
------	---	--	---	---	---

Java Servlets

-  Easy for Developer - *Difficult to work with since everything must be done manually*
-  Easy for Designer - *Code and HTML are mixed in one Java file*
-  Integration with Sakai - *No real integration with Sakai other than the fact that it is functional*

Other notes:

-  Almost all Java based web technologies are based on Servlets
-  Tried and tested technology (1997) with strong community and docs
-  Pure Java code
-  HTML is placed in the Java class, impossible for UI designers to work with
-  Development this way is painful and slow compared to more modern methods
-  Average AJAX/Javascript integration

Velocity

-  Easy for Developer - *Good docs and fairly easy to get started using it*
-  Easy for Designer - *Separation is mediocre and not easy for UI designers to work with*
-  Integration with Sakai - *Currently good but is not really being worked on much anymore*

Other notes:

-  Decent docs and community
-  Most used display technology in Sakai
-  New development in Velocity is discouraged by the community, mostly carried over from CHEF
-  Average AJAX/Javascript integration

JSP

-  Easy for Developer - *One of the easiest technologies to pick up and start using*
-  Easy for Designer - *Code and UI are mixed together in one file, no separation but does resemble html*
-  Integration with Sakai - *No real integration with Sakai other than the fact that it is functional*

Other notes:

- + Dynamic Java web content scripting language, similar to PHP
- + Can be changed without restarting the server
- i Well tested technology with large community, excellent docs, and many books
- Compiled when accessed which can cause a noticeable delay when accessing a page for the first time
- Has many of the same weaknesses that other scripting languages have
- i Average AJAX/Javascript integration

JSF

- ✗ Easy for Developer - *High learning curve and difficult to work with and get used to*
- ✗ Easy for Designer - *Does not use HTML in templates and can include code with the UI, hard to work with for UI designers*
- ✓ Integration with Sakai - *Has widgets and best integration with Sakai (currently) that is being kept up to date*

Other notes:

- i The second most used display technology in Sakai (after Velocity)
- i MyFaces and Oracle ADF Faces (other JSF implementations), pretty good docs and books
- Pretty heavyweight and has problems in the current version
- Poor AJAX/Javascript integration
- Does not work with browser navigation

RSF

- ✓ Easy for Developer - *Learning curve is moderate but is easy to use once familiarity is gained*
- ✓ Easy for Designer - *Excellent separation of Code and UI, uses pure XHTML templates*
- i Integration with Sakai - *Current average but improving rapidly, new widgets being developed*

Other notes:

- + Good docs and code samples available on the RSF Wiki
- i Greater separation of UI and code can require more files and lines of code
- i Not used in Sakai core yet but is fully functional in it and integrates somewhat
- Small user community, no books yet
- Still in development
- + Good AJAX/Javascript integration

Wicket

- + Easy for Developer - *Uses pure Java and no XML configuration required*
- + Easy for Designer - *Excellent separation of Code and UI, uses pure XHTML templates*
- i Integration with Sakai - *Excellent, see the existing tools for the minimal setup required*

Other notes:

- + Good docs and code samples available on the Wicket Wiki
- + Several core tools are now Wicket based and have rich AJAX UI's
- + Large user community, several excellent books
- + Has been around for a number of years and has a large developer base. Very stable.
- + Built in AJAX integration using pure Java