General Description

The goal is to allow Tiki sites to be accessed by each and every man, woman, and child using the necessary assistive technology. Such technology can include screen readers, mouth balls, touch sticks, or one of a miriad of other devices. To provide users of these assistive technology devices with equal access to information, Tikiwiki must present the necessary information in a specific manner. In doing so, Tikiwiki will automatically adapt the presentation of any given page to the limits and abilities of the user, and that user's user agent. Ideally, this process should put as little burden as possible on Tikiwiki, theme developers, administrators, and users. It should also ideally be completely transparent to all, while being handled by & integrated into the core, while also preserving the environment & vision of Tikiwiki.

Achieving WAI-508 Compliance

Notice: This document is intended to serve as a Roadmap for what needs to be done for WAI-508 accessibility in Tikiwiki. It is not a how-to documentation. If you would like to help make Tikiwiki WAI-508 compliant, please contact musus for specifics.
Below is a step-by-step process for building the first of Tiki-Certified WAI-508 Compliant Themes. The theme will be called *508*. If you would like to join the WAI-508 Team, checkout #tikiaccess on irc.freenode.net & get in on the action. All are welcome.

Steps:

1. Forms
   1. Ensure all form fields are labeled with a label tag, and tether it to its corresponding form field using a specific convention.
   2. Ensure labels are placed to the left of their assigned form field
   3. Modify *name* attribute for form fields to match *ID* attribute (some browsers don't yet support the *ID* attribute)
   4. Remove local styling (P1)
   5. Format input fields & buttons for height, background color, width, etc
2. Hyperlinks
   1. Remove local styling (P1)
   2. Remove ALT attribute (P1)
   3. Place *title* attribute immediately after tag, using the convention, "Click here to..."
   4. Duplicate *link* class in CSS documents renamed to "a"
   5. Remove "link" class from templates (it is unnecessary & redundant as a default class)
   6. Remove hyperlinks from labels (?)
3. Tables
   1. Change table column headers from td tags to th tags (for data tables only) (P1)
   2. Remove local styling (P1)
   3. Duplicate even & odd classes in CSS documents, renamed to ". td" to keep formatting for borders & such consistent
   4. In the templates, move even & odd classes to tr tags
   5. Implement TBODY, THEAD, TFOOTER, etc
4. Develop Theme Manager for:
   1. Note: Brian & musus have begun this part
   2. Themes & Icon Themes:
      1. Customizing & Tweaking
      2. Building completely new themes (includes CSS, templates & icons)
      3. Managing alternate text for media, TITLE text for links, etc.
5. Replace hardcoded images with Tikicon
6. Ultimately eliminate dependence on JavaScript. In the meantime, add the NOSCRIPT element for every JS.
7. Images
   1. Make ALT text **meaningful** (P1)

    2. Add hyperlinked D for every image or use 1px transparent gif as described on WebAccessibilityInitiativeDev
    3. Replace ALT attribute with LONGDESC when applicable (P1)
    4. Remove TITLE attribute (P1)
8. Add strategically placed "Jump to ..." links & anchors (P2)
9. Develop))TikiAccessibility((Certified icon (P1)
10. Expand any & all abbreviations & acronyms (P3)
11. Set logical tab order (P3)

Who is doing what?

CSS and WAI expert: musus
CSS expert: beerlounge.de
PHP: btodoroff
Wisdom: mose

Timescale

- *508* is scheduled to be complete by 1.9rc1