

WCAG Checklist

Last Updated Saturday, 17 September 2005

The following is the W3C's checklist applied to Joomla 1.0. WCAG's requirements can sometimes be open to "interpretation". This checklist applies only to the core Joomla install, not to any 3rd Party Component (this will be addressed in another document).

Many of WCAG's requirements are outside of Joomla! development team. Some requirements must be filled by the template designer. Some other requirements need to be addressed by the content manager. As such, Joomla! will offer the ability to make WCAG compliant sites, but will not guarantee that sites created using Joomla! will be compliant.

We have used the following text to denote the status of each item in the checklist:

- OK: Means this is a requirement that can be achieved via Joomla!
- To Do: Means this is a feature that the development team needs to include in future releases of Joomla!
- Not Core (OK): Means this requirement is outside the control of the development team (e.g. template or content focus)

Note that some requirements have been marked as both To Do and Not Core (OK). Those instances denote that the Joomla! engine may be producing content that needs to comply, but that generally it is outside the hands of the development team.

{mospagebreak title=Priorities explained}
Priority 1

You must comply with those. Without it, you are cutting out a bunch of people from being able to access your site. By complying with Priority 1, you have a mostly accessible site.
Priority 2

You should comply with those. If you don't, you are making the life of many of your visitors much more difficult. By complying with Priority 2, you make a site that is free of significant barriers.
Priority 3

You may comply with those. If you do, you will have a majority of people able to use your site. By complying with Priority 3, you are really increasing access to your site.

{mospagebreak title=Priority 1}

The following checklist was adapted from: <http://www.w3.org/TR/WCAG10/full-checklist.html>
Priority 1 checkpoints

In General (Priority 1)

Yes
No
N/A

1.1 Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated

GIFs), applets and programmatic objects, ascii art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.
OK

2.1 Ensure that all information conveyed with color is also available without color, for example from context or markup.

Not Core (OK)

4.1 Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions).

Not Core (OK)

6.1 Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document.

To Do
Not Core (OK)

6.2 Ensure that equivalents for dynamic content are updated when the dynamic content changes.

Not Core (OK)

7.1 Until user agents allow users to control flickering, avoid causing the screen to flicker.

Not Core (OK)

14.1 Use the clearest and simplest language appropriate for a site's content.

Not Core (OK)

And if you use images and image maps (Priority 1)
Yes
No
N/A

1.2 Provide redundant text links for each active region of a server-side image map.

Not Core (OK)

9.1 Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

Not Core (OK)

And if you use tables (Priority 1)

Yes
No
N/A

5.1 For data tables, identify row and column headers.

To Do
Not Core (OK)

5.2 For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.

To Do
Not Core (OK)

And if you use frames (Priority 1)

Yes
No
N/A

12.1 Title each frame to facilitate frame identification and navigation.

Not Core (OK)

And if you use applets and scripts (Priority 1)

Yes
No
N/A

6.3 Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

Not Core (OK)

And if you use multimedia (Priority 1)

Yes
No
N/A

1.3 Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.

Not Core (OK)

1.4 For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation.

Not Core (OK)

And if all else fails (Priority 1)

Yes

No
N/A

11.4 If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.

Not Core (OK)

{mospagebreak title=Priority 2}

Priority 2 checkpoints

In General (Priority 2)
Yes
No
N/A

2.2 Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. [Priority 2 for images, Priority 3 for text].

Not Core (OK)

3.1 When an appropriate markup language exists, use markup rather than images to convey information.

Not Core (OK)

3.2 Create documents that validate to published formal grammars.

Not Core (OK)

3.3 Use style sheets to control layout and presentation.

To Do
Not Core (OK)

3.4 Use relative rather than absolute units in markup language attribute values and style sheet property values.

Not Core (OK)

3.5 Use header elements to convey document structure and use them according to specification.

To Do

3.6 Mark up lists and list items properly.

Not Core (OK)

3.7 Mark up quotations. Do not use quotation markup for formatting effects such as indentation.

Not Core (OK)

6.5 Ensure that dynamic content is accessible or provide an alternative presentation or page.

Not Core (OK)

7.2 Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off).

Not Core (OK)

7.4 Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.

Not Core (OK)

7.5 Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.

Not Core (OK)

10.1 Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.

To Do

11.1 Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.

Not Core (OK)

11.2 Avoid deprecated features of W3C technologies.

To Do

12.3 Divide large blocks of information into more manageable groups where natural and appropriate.

Not Core (OK)

13.1 Clearly identify the target of each link.

To Do

13.2 Provide metadata to add semantic information to pages and sites.

To Do

13.3 Provide information about the general layout of a site (e.g., a site map or table of contents).

To Do

13.4 Use navigation mechanisms in a consistent manner.

Not Core (OK)

And if you use tables (Priority 2)

Yes
No
N/A

5.3 Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).

To Do

5.4 If a table is used for layout, do not use any structural markup for the purpose of visual formatting.

Not Core (OK)

And if you use frames (Priority 2)

Yes
No
N/A

12.2 Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.

Not Core (OK)

And if you use forms (Priority 2)

Yes
No
N/A

10.2 Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.

OK

12.4 Associate labels explicitly with their controls.

OK

And if you use applets and scripts (Priority 2)

Yes

No

N/A

6.4 For scripts and applets, ensure that event handlers are input device-independent.

Not Core (OK)

7.3 Until user agents allow users to freeze moving content, avoid movement in pages.

Not Core (OK)

8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies [Priority 1 if functionality is important and not presented elsewhere, otherwise Priority 2.]

Not Core (OK)

9.2 Ensure that any element that has its own interface can be operated in a device-independent manner.

Not Core (OK)

9.3 For scripts, specify logical event handlers rather than device-dependent event handlers.

Not Core (OK)

{mospagebreak title=Priority 3}

Priority 3 checkpoints

In General (Priority 3)

Yes

No

N/A

4.2 Specify the expansion of each abbreviation or acronym in a document where it first occurs.

Not Core (OK)

4.3 Identify the primary natural language of a document.

To Do

9.4 Create a logical tab order through links, form controls, and objects.

To Do

9.5 Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.

To Do

10.5 Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.

To Do

Not Core (OK)

11.3 Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.)

OK

13.5 Provide navigation bars to highlight and give access to the navigation mechanism.

13.6 Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.

To Do

13.7 If search functions are provided, enable different types of searches for different skill levels and preferences.

To Do

13.8 Place distinguishing information at the beginning of headings, paragraphs, lists, etc.

Not Core (OK)

13.9 Provide information about document collections (i.e., documents comprising multiple pages.).

To Do

13.10 Provide a means to skip over multi-line ASCII art.

Not Core (OK)

14.2 Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.

Not Core (OK)

14.3 Create a style of presentation that is consistent across pages.

OK

And if you use images and image maps (Priority 3)

Yes

No

N/A

1.5 Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.

Not Core (OK)

And if you use tables (Priority 3)

Yes

No

N/A

5.5 Provide summaries for tables.

To Do

Not Core (OK)

5.6 Provide abbreviations for header labels.

To Do

Not Core (OK)

10.3 Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.

And if you use forms (Priority 3)

Yes

No

N/A

10.4 Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.

To Do

{mospagebreak title=Explanations / To Do}

Priority 1

6.1 Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document

- Organise the content so it has a logical reading structure without stylesheets.
- Core: Ensure that generated content is rendered using structural markup (such as div).

5.1 For data tables, identify row and column headers.

- Properly use td for data cell, and th for header cells.
- Core: Using tables to deliver the list of content items in Category section is an appropriate use of tables, particularly if dates, authors, hits, etc are listed with each Item Title. This is considered "tabular data". Ensure that table headers are using th.

5.2 For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.

- Use markup to associate cells with their relevant headers
- Core: Assign ids for each th, and associate the same id to each td in the same column. See <http://www.w3.org/TR/WCAG10-HTML-TECHS/#identifying-table-rows-columns> for example.

Priority 2

3.3 Use style sheets to control layout and presentation.

- Visual design and layout should be handled through the use of CSS. Content should be completely separated from its presentation.
- Core: Develop alternative template to deliver content without tables. Best suggestion at this point would be to create a set of tableless content pages, using patTemplates. These alternative files could be packaged in the core, and the user could select to use either the "old" tabled content, or the "new" tableless. The advantage of this is that older templates relying on legacy code won't break.

3.5 Use header elements to convey document structure and use them according to specification.

- Use proper semantic markup to determine document content hierarchy.
- Core: Assign elements such as moduleheader, contentdeading, etc by appropriate h1, h2, h3, etc. Please refer to suggested use of headers for appropriate of "translation".

10.1 Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.

- Don't force links to open in new windows without warning the user.
- Core: Provide a way to let users know links open in a new browser window. e.g. "The following link opens in a new browser window". Some places uses a small popup notification along the same lines.

11.2 Avoid deprecated features of W3C technologies.

- Don't use deprecated elements such as the font tag
- Core: Go through and make sure that old tags don't get generated. In particular: font, b, i

13.1 Clearly identify the target of each link.

- Several links with the same name are confusing (such as "read more"). Also, links saying simply "click here" are not meaningful.
- Core: Change the behaviour of the link added for "Read more". Use the title alias (or title if alias doesn't exist). Instead of hard coding "read more", provide the user with a variable to edit the text. Provide another variable to go after the title alias, so users can build links such as "Read more about the title alias article"

13.2 Provide metadata to add semantic information to pages and sites.

- Some metadata is already used, such as title, and doctype is up to the template developer. Providing users the ability to include the link tag in the document's head would allow greater accessibility. Link can be used to build navigation systems or provide alternative versions (e.g. text-only). See <http://www.w3.org/TR/WCAG10-HTML-TECHS/#document-meta> for more details.
- Core: Explore the possibility of adding link tag generation to the head of individual content items.

13.3 Provide information about the general layout of a site (e.g., a site map or table of contents).

- Particularly on complex sites, a sitemap is really helpful to find what you're looking for. This is a usability factor as much as an accessibility factor.
- Core: While there are 3PD components to create sitemaps, we should incorporate a sitemap in the core if we want to be able to offer WCAG compliant sites. Sitemaps should give the flexibility to users to select which bits appear on the sitemap, in which order. A possible solution would be to list all sections, categories, static content, components (and

perhaps even an option for content items), and let the user select which ones are to appear on the sitemap, and in which order. This may become clumsy with sites with 1,000's of items, but it would be the most flexible solution. Using the menu structure to deliver the sitemap (as some components do), is too limiting.

5.3 Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).

- If we use tables for layout, the content must make sense when you read it from left-to-right and top-to-bottom (for most languages), as this is how a screen reader goes through a table. Nested tables cause more problems. Simple solution to this iss the oft-requested tableless content generation.

- Core: Develop alternative template to deliver content without tables, as already addressed by point 3.3.