

LIFT for Dreamweaver - Nielsen Norman Group Edition

Test Package: W3C WCAG P.1 Accessibility

UsableNet Inc.

83 Mercer Street, New York City, NY 10012

ph. 212 9655388

fax: 212 96553991

support@usablenet.com

August 28, 2002

Contents

1	Introduction	2
2	WCAG 1.0 checkpoint 1.1	3
2.1	Image with valid ALT [WCAG 01.1]	3
2.2	Composition image with valid ALT [WCAG 01.1]	5
2.3	Banner image with valid ALT [WCAG 01.1]	7
2.4	Spacer image with valid ALT [WCAG 01.1]	10
2.5	Bullet image with valid ALT [WCAG 01.1]	11
2.6	Decorative image with valid ALT [WCAG 01.1]	13
2.7	Repeated image with valid ALT [WCAG 01.1]	14
2.8	Repeated image with consistent ALT [WCAG 01.1]	16
2.9	Image with equivalent ALT [WCAG 01.1]	18
2.10	Composition image with equivalent ALT [WCAG 01.1]	19
2.11	Banner image with equivalent ALT [WCAG 01.1]	20
2.12	Image with valid LONGDESC/D-LINK [WCAG 01.1]	22
2.13	No LONGDESC for spacer image [WCAG 01.1]	24
2.14	Link image with valid ALT [WCAG 01.1]	25
2.15	Link image with equivalent ALT [WCAG 01.1]	28
2.16	Map image with valid ALT [WCAG 01.1]	29
2.17	Thumbnail image with valid ALT [WCAG 01.1]	31
2.18	Thumbnail image with equivalent ALT [WCAG 01.1]	34
2.19	Thumbnail with valid LONGDESC/D-LINK [WCAG 01.1]	35
2.20	Button image with valid ALT [WCAG 01.1]	37
2.21	Button image with equivalent ALT [WCAG 01.1]	40
2.22	Hidden link with valid ALT [WCAG 01.1]	41

2.23	Hidden link image with equivalent ALT [WCAG 01.1]	43
2.24	INPUT with valid ALT [WCAG 01.1]	44
2.25	INPUT with equivalent ALT [WCAG 01.1]	46
2.26	Form Button with valid ALT [WCAG 01.1]	47
2.27	Form Button with equivalent ALT [WCAG 01.1]	49
2.28	SCRIPT with valid NOSCRIPT [WCAG 01.1]	50
2.29	SCRIPT with equivalent NOSCRIPT [WCAG 01.1]	53
2.30	Hotspot with valid ALT [WCAG 01.1]	54
2.31	Hotspot with equivalent ALT [WCAG 01.1]	57
2.32	APPLET with valid ALT [WCAG 01.1]	59
2.33	APPLET with equivalent ALT [WCAG 01.1]	60
2.34	APPLET with valid CONTENT [WCAG 01.1]	61
2.35	Image OBJECT with valid CONTENT [WCAG 01.1]	63
2.36	Image OBJECT with equivalent CONTENT [WCAG 01.1]	65
2.37	Audio/video OBJECT with valid CONTENT [WCAG 01.1]	66
2.38	Audio/video OBJECT with equivalent CONTENT [WCAG 01.1]	67
2.39	OBJECT with valid CONTENT [WCAG 01.1]	68
2.40	OBJECT with equivalent CONTENT [WCAG 01.1]	70
2.41	Linked AUDIO with equivalent CONTENT [WCAG 01.1]	71
3	WCAG 1.0 checkpoint 1.2	73
3.1	Links are needed for server-side image map [WCAG 01.2]	73
4	WCAG 1.0 checkpoint 1.3	75
4.1	Multimedia with equivalent audio description [WCAG 01.3]	75
5	WCAG 1.0 checkpoint 1.4	77
5.1	Multimedia with synchronized alternative [WCAG 01.4]	77
6	WCAG 1.0 checkpoint 2.1	79
6.1	Color is not essential [WCAG 02.1]	79
7	WCAG 1.0 checkpoint 2.2	81
7.1	Colors are visible [WCAG 02.2]	81
8	WCAG 1.0 checkpoint 4.1	83
8.1	Clarify natural language usage [WCAG 04.1]	83
9	WCAG 1.0 checkpoint 5.1	85
9.1	Data table should have headers [WCAG 05.1]	85
9.2	Data table with valid THs [WCAG 05.1]	87
9.3	Cell of data table should refer to headers [WCAG 05.1]	89
9.4	Data tables should be defined by TABLE tag [WCAG 05.1]	91
10	WCAG 1.0 checkpoint 5.2	93
10.1	Multiple headers should be marked in data tables [WCAG 05.2]	93

11 WCAG 1.0 checkpoint 6.1	95
11.1 Style sheets should not be necessary [WCAG 06.1]	95
12 WCAG 1.0 checkpoint 6.2	97
12.1 Text only equivalent page may be needed [WCAG 06.2]	97
13 WCAG 1.0 checkpoints 6.3, 8.1, 9.2, 9.3	99
13.1 Scripts are accessible [WCAG 06.3]	99
14 WCAG 1.0 checkpoint 6.5	102
14.1 No javascript links are used [WCAG 06.5]	102
15 WCAG 1.0 checkpoint 7.1	103
15.1 GIFs do not cause the screen to flicker [WCAG 07.1]	103
15.2 Avoid causing the screen to flicker [WCAG 07.1]	104
16 WCAG 1.0 checkpoint 7.4	106
16.1 No auto refresh is used [WCAG 07.4]	106
17 WCAG 1.0 checkpoint 7.5	108
17.1 No auto redirect is used [WCAG 07.5]	108
18 WCAG 1.0 checkpoint 9.1	110
18.1 No server-side image maps should be used [WCAG 09.1]	110
19 WCAG 1.0 checkpoint 9.2	112
19.1 Jump Menu is Device Independent [WCAG 09.2]	112
20 WCAG 1.0 checkpoint 12.1	115
20.1 FRAME with valid TITLE [WCAG 12.1]	115
20.2 IFRAME with valid TITLE [WCAG 12.1]	117
21 WCAG 1.0 checkpoint 12.4	119
21.1 Form Control with Label [WCAG 12.4]	119
22 WCAG 1.0 checkpoint 13.6	122
22.1 Skip repetitive links [WCAG 13.6]	122
23 WCAG 1.0 checkpoint 14.1	125
23.1 Use clear language for site's content [WCAG 14.1]	125
A Test Categories	127

1 Introduction

This package contains tests that identify accessibility issues regarding W3C/WAI Web Content Accessibility Guidelines (WCAG) 1.0 priority 1 guidelines.

2 WCAG 1.0 checkpoint 1.1

2.1 Image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic
- **Test Categories:**
 - W3C WCAG Accessibility
 - Images
 - ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This rule is a **UsableNet extension** that does not consider spacers or other images that may play only a decorative role in the page. It applies only to images that are believed to carry some information or have some effect on visitors of the pages.

The test checks if images embedded in the page that are not used for decorative purposes only have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The image included in this document does not have a **valid** textual description. A **valid description** is a string value of the ALT attribute that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT is empty:** ALT of image is the empty string ""
- **ALT is blank:** ALT of image is the blank string " "
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file

- **ALT with filename:** ALT of image describes only the filename of the image
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- if the image is used as the content of a link and you provide link text too, use a space as the ALT attribute value of the IMG element. In such a case link text should be the alternative description for the image;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- As a rough guideline, for buttons use the same text that is shown by the image;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **BEWARE:** the ALT attribute should be the empty string ("") in cases where the image is already described by surrounding text;
- **BEWARE:** all images included in A links (including transparent GIFs) need to have valid ALTs describing the link destination.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- the W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.2 Composition image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#).

This test is a **UsableNet extension** that considers only images that are slices of bigger images in the page.

The test checks if such composition images embedded in the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The image included in this document does not have a **valid** textual description. A **valid description** for a composition image is a value of the ALT attribute that:

- may be the empty string (“”);
- may be a string with 1 or more spaces (“ ”);
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file
- **ALT with filename:** ALT of image describes only the filename of the image
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services that presents a page as would be displayed by the lynx textual browser;
- For composition images (i.e. slices of bigger images) the ALT can be also empty or blank. This is in fact appropriate when you have provided a non empty/non blank ALT for **another slice** of that image. In such a case users of speaking browsers, for example, would have to listen to the textual description of the image just once.
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string (“”) so that they are not considered by screen-readers. Similarly for images like bullets;

- **ATTENTION:** the ALT attribute should be the empty string ("") in cases where the image is already described by surrounding text;
- **ATTENTION:** all images included in A links (including transparent GIFs) need to have valid ALTs describing the link destination.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

For composition images (i.e. slices of bigger images) the ALT can be also empty or blank. This is in fact appropriate when you have provided a non empty/non blank ALT for another slice of that image. In such a case users of speaking browsers, for example, would have to listen to the textual description of the image just once.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.3 Banner image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images

– ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This rule is a **UsableNet extension** that considers only images that are banners.

The test checks if banners embedded in the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The banner included in this page does not have a **valid** textual description. A **valid description** is a string value of the ALT attribute that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for banner
- **ALT is empty:** ALT of banner is the empty string "". (This may be OK for banners linking to pages that are known to be not accessible.)
- **ALT is blank:** ALT of image is the blank string " ". (This may be OK for banners linking to pages that are known to be not accessible.)
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file
- **ALT with filename:** ALT of image describes only the filename of the image
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;
- for a banner, the ALT should be an invitation for the visitor to follow the link.
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- if the image is used as the content of a link and you provide link text too, use a space as the ALT attribute value of the IMG element. In such a case, link text should be the alternative description for the image;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- As a rough guideline, for buttons use the same text that is shown by the image;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less that 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **ATTENTION:** the ALT attribute should be the empty string ("") in cases where the image is already described by surrounding text;
- **ATTENTION:** all images included in A links (including transparent GIFs) need to have valid ALTs describing the link destination.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);

- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.4 Spacer image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#).

More specifically it is a UsableNet extension to those standards. It checks if images embedded in the page that are used for decorative purposes only (**spacers**) have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

Images included in this document that seem to be spacers (that is they serve a decorative purpose only) don't have corresponding **valid** textual descriptions. A **valid description** is a string value of the ALT attribute that is **blank**, i.e. made of one or more spaces or **empty**, i.e. "".

- **Specific Issues:**

- **ALT not defined:** No ALT defined for spacer image
- **ALT not empty:** ALT of spacer image should be empty or blank and isn't

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that for a spacer the description should be blank (" ") or empty (""), but it should exist anyway. Similarly for images like bullets.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. Images that have only a decorative role don't need to have a textual description since users unable to see the image will not care about it.

In addition, reading browsers or screen readers would not bother their users if the ALT text is the blank or empty string. However the ALT attribute should be defined, otherwise some screen-reader and reading browser would insert the filename for example, disturbing the user while listening/reading the page.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#)
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.5 Bullet image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#).

More specifically it is a UsableNet extension to those standards. It checks if images embedded in the page that are used as bullets have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The image included in this document seems to be a bullet and it does not have a corresponding **valid** textual description. A **valid description** is a string value of the ALT attribute that is **blank** (i.e. made of one or more spaces) or **empty**, i.e. "".

- **Specific Issues:**

- **ALT not defined:** No ALT defined for bullet image
- **ALT not empty:** ALT of bullet image should be empty or blank and isn't
- **ALT is '*' or '-':** ALT of bullet image is a '*' or a '-'; it should be empty or blank

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that for a bullet the description should be blank (" ") or empty (""), but it should exist anyway. Similarly for images like spacers.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. Images that have only a decorative role don't need to have a textual description since users unable to see the image will not care about it.

For bullets, that are repeated many times in a page, the user would be bothered with the same description over and over.

If the description is a single '*' or '-' reading browser might keep repeating the words 'asterisk' or 'dash' over and over.

Reading browsers or screen readers would not bother their users if the ALT text is the blank or empty string. However the ALT attribute should be defined, otherwise some screen-reader and reading browser would insert the filename for example, disturbing the user while listening/reading the page.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#)
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.6 Decorative image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#).

More specifically it is a UsableNet extension to those standards. It checks if images embedded in the page that are used for decorative purposes only have a **valid** alternative description as an ALT attribute.

A valid ALT attribute for a decorative image is either the empty string ("") or the blank string (" ").

- **Issue Description:**

The image seems to play only a decorative role within the page. It does not have a valid alternative description (i.e. ALT attribute).

Either the ALT attribute is not defined or it is neither empty nor blank.

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT not empty:** ALT of image should be empty or blank and it isn't

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that for a decorative image the description should be blank (" ") or empty (""), but it should exist anyway.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. Images that have only a decorative role don't need to have a textual description since users unable to see the image will not care about it.

In addition, reading browsers or screen readers would not bother their users if the ALT text is the blank or empty string. However the ALT attribute should be defined, otherwise some screen-reader and reading browser would insert the filename for example, disturbing the user while listening/reading the page.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- tutorial on the [use of ALT text in IMGs](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- another tutorial on [alternative content for graphics](#) rich of examples;
- discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#)
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.7 Repeated image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This rule is a **UsableNet extension** that considers only images that play a decorative role in the page and that are repeated more than once.

The test checks if **symbol images** embedded in the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The symbol image that is included in this page and that is repeated several times does not have a **valid** textual description. A **valid description** is a string value of the ALT attribute that:

- is not the name of the file containing the image;
- does not say only which is the size of the image file;

- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file
- **ALT with filename:** ALT of image describes only the filename of the image
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;
- this is especially important if you plan to define the ALT as a "*" , "-" or "|": a screen reader would read the words "asterisk", "dash" or "bar" several times; double check if these characters are really needed. If not, put an empty string (i.e. "").
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less that 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **ATTENTION:** all images included in A links (including transparent GIFs) need to have valid ALTs describing the link destination.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.8 Repeated image with consistent ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically this test is a **UsableNet extension** that considers only images that play a decorative role in the page and that are repeated more than once (symbol images).

The test checks if **symbol images** embedded in the page that have a valid ALT, are always associated to the **same** ALT text.

- **Issue Description:**

The image included in this page is repeated more than once in the page. One of its previous occurrences has a different ALT than the currently selected one.

- **How to fix:**

Change the ALT attribute to the IMG tag.

Remember that:

- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;
- this is especially important if you plan to define the ALT as a "*" , "-" or "|": a screen reader would read the words "asterisk", "dash" or "bar" several times; double check if these characters are really needed. If not, put an empty string (i.e. "").
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less that 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **ATTENTION:** all images included in A links (including transparent GIFs) need to have valid ALTs describing the link destination.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);

- the W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.9 Image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Images
 - Manual
 - ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It checks, for images embedded in the page that are not used for decorative purposes only and whose alternative description is valid, if the description is equivalent to the image itself.

This is manual test.

- **Issue Description:**

The image appears to have more than a decorative role. It should have an equivalent textual description in the ALT attribute of the IMG tag.

- **How to fix:**

Please check if the current ALT description conveys the same meaning of the image.

It should explain the role the image plays in the page: why it's there, what does it represent, how does it present the information. (Imagine hearing the description over the phone.)

However if the image has only decorative purposes (like bullets), its ALT string should be the **empty string** ("") or the **blank** one (" ").

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what does it represent, how does it present information.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not least, users of cellular phones, PDAs, browsers installed in cars, etc.).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen's Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.10 Composition image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Manual
- ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It checks, for images embedded in the page that are slices of bigger images, and whose alternative description is valid, if the description is equivalent to the image itself.

This is manual test.

- **Issue Description:**

The image is a slice of bigger image. It should have an **equivalent textual description** in the ALT attribute of the IMG tag.

- **How to fix:**

Please check if the current ALT description conveys the same meaning of the image.

It should explain the role the image plays in the page: why it's there, what does it represent, how does it present the information. (Imagine hearing the description over the phone.)

However since the image is a slice of a bigger image, at least one slice needs to have a valid equivalent description. The remaining ones should have the ALT set to the empty string ("") or the blank string (" ") to avoid annoying users of screen readers or reading browsers.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what does it represent, how does it present information.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not least, users of cellular phones, PDAs, browsers installed in cars, etc.).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen's Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.11 Banner image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility

- Images
- Manual
- ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It asks the user to check, for banner images embedded in the page whose alternative description is valid, if the description is equivalent to the image itself.

This is manual test.

- **Issue Description:**

The image appears to be a banner. It should have an equivalent textual description in the ALT attribute of the IMG tag.

- **How to fix:**

Please check if the current ALT description conveys the same meaning of the banner image.

Its ALT description should contain the same message conveyed by the image and motivate the user to follow the link. (Imagine hearing the description over the phone.)

However if the banner points to a page that is not accessible, defining an empty ALT ("") is OK. In this way the visitor would not be able to perceive the banner.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what does it represent, how does it present information.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not least, users of cellular phones, PDAs, browsers installed in cars, etc.).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);

- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen’s Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.12 Image with valid LONGDESC/D-LINK [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The rule implements WAI guideline 1.1 ”Provide a text equivalent for every non-text element” corresponding to rule 1194.22(a) of the [508 standard](#). More specifically it implements and extends [technique 1.1.2](#) of the WAI [AERT document](#).

The test is a UsableNet extension to those standards because it does not apply to images that are believed to have a decorative role only nor to images that are used as links or buttons.

It checks if the image has a **LONGDESC** attribute or if the image is followed by a **D-link** element (that is `D</>`).

The test also checks if either the **LONGDESC** attribute or **D-link** href attribute contain a valid URL pointing to an HTML file.

Finally, in case both attributes are defined, the test checks if they are the same.

- **Issue Description:**

The image has no valid **LONGDESC** or **D-link** element or, if both are present, they point to different files.

- **Specific Issues:**

- **LONGDESC attribute not defined:** Missing both the **LONGDESC** attribute and the **D-link** element
- **LONGDESC file does not exist:** Invalid **LONGDESC** attribute: mentioned file does not exist
- **LONGDESC file is not HTML:** Invalid **LONGDESC** attribute: mentioned file is not an HTML file
- **LONGDESC file is empty:** Invalid **LONGDESC** attribute: no file specified
- **LONGDESC URL is dead:** Invalid **LONGDESC** attribute: its URL is dead
- **LONGDESC URL generated bad response:** Invalid **LONGDESC** attribute: got an error response from the server of its URL

- **LONGDESC URL has a bad protocol:** Invalid LONGDESC attribute: it is not a local file nor an HTTP URL
 - **D-link file does not exist:** Invalid D-link: mentioned file does not exist
 - **D-link file is not HTML:** Invalid D-link: mentioned file is not an HTML file
 - **D-link file is empty:** Invalid D-link: no file specified
 - **D-link url is dead:** Invalid D-link: its URL is dead
 - **D-link bad response:** Invalid D-link: got an error response from the server of its URL
 - **D-link bad protocol:** Invalid D-link: it is not a local file nor an HTTP URL
 - **D-link and LONGDESC are different:** LONGDESC and D-link point to different files
- **How to fix:**

Please define a proper **LONGDESC** attribute for the image (by specifying the address – URL – of an HTML file).

Since not many browsers at the moment support the **LONGDESC** attribute, you may also want to place, near the image, a so called D-link, a normal textual link with label "D" pointing to an HTML page providing a full image description. Example:

```
<IMG src="chart.gif" alt="chart of beverages distribution"
longdesc="chart.html"><A href="chart.html">D</A>
```

Alternatively, placing a rich textual description close to the image (like a caption) is a viable solution. In such a case you don't need the **LONGDESC** attribute or the D-link.

- **Why to fix:**

The ALT attribute cannot be used to provide a full length description of the content of an image. Think of a diagram, a histogram or a chart. Or even a product picture in an online catalogue. To describe its content you need more flexibility.

The **LONGDESC** attribute can be used to provide a long description of the associated image that would not fit in the ALT attribute. By including **LONGDESC="any_HTML_file"** in your IMG tag you can link the image to the HTML file containing a formatted description of the image. The long description (unlike the ALT attribute) can contain HTML code, with links to other resources, formatting instructions, etc.

However, since not many browsers at the moment support the **LONGDESC** attribute, you may also want to place, near the image, a so called D-link, a normal textual link with label "D" pointing to an HTML page providing a full image description. Example:

```
<IMG src="chart.gif" alt="chart of beverages distribution"
longdesc="chart.html"><A href="chart.html">D</A>
```

Alternatively, placing a rich textual description close to the image (like a caption) is a viable solution. In such a case you don't need the **LONGDESC** attribute or the D-link.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not the least, users of cellular phones, PDAs, browsers installed in cars, etc.)

- **Learn More:**

- HTML 4.0 standard on the [IMG tag](#);
- HTML 4.0 standard on the [LONGDESC attribute](#);
- W3C on [how to define the LONGDESC attribute](#);
- [checkpoint 3](#) from IBM Web accessibility checklist - version 3.1
- some examples about [LONGDESC and D-links](#);
- W3C on [how to define the ALT attribute](#).

2.13 No LONGDESC for spacer image [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element" corresponding to rule 1194.22(a) of the [508 standard](#).

More specifically it implements and extends [technique 1.1.2](#) of the WAI [AERT document](#).

The test is a UsableNet extension to those standards because it applies only to images that are **spacers**.

It checks that spacers do not have **LONGDESC** attributes.

- **Issue Description:**

Image has a **LONGDESC** attribute even though it appears that the image role in the page is **decorative only**.

Such images should not have any long description.

- **How to fix:**

Remove the LONGDESC attribute from the IMG element.

- **Why to fix:**

The **LONGDESC** attribute can be used to provide a long description of the associated image that would not fit in the ALT attribute. This may be needed, for example, if the image represents a chart, a diagram, whose explanation may be rather long.

However, for images that are used only for **decorative** purposes, such a long description is useless, and people using browsers that take advantage of that feature will find it distracting.

- **Learn More:**

- HTML 4.0 standard on the [IMG tag](#);
- HTML 4.0 standard on the [LONGDESC attribute](#);
- W3C on [how to define the LONGDESC attribute](#);
- [checkpoint 3](#) from IBM Web accessibility checklist - version 3.1
- some examples about [LONGDESC and D-links](#);
- W3C on [how to define the ALT attribute](#).

2.14 Link image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This test is a **UsableNet extension** that considers only images used as link labels.

The test checks if link images embedded in the page have a **valid** alternative description as an ALT attribute or if they are surrounded by informative text.

- **Issue Description:**

The image included in the label of the link does not have a **valid** textual descriptions. The **textual description** is the value of the ALT attribute or the text surrounding the image. A description is **valid** if it:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT is empty:** ALT of image is the empty string "" and no text surrounds it
- **ALT is blank:** ALT of image is the blank string " " and no text surrounds it
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file and no text surrounds it
- **ALT with filename:** ALT of image describes only the filename of the image and no text surrounds it
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text and no text surrounds it

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the ALT attribute (and the link text, if any) should describe the link destination, i.e. what the user would find after clicking on the link;
- if you provide link text too, use a space as the ALT attribute value of the IMG element. In such a case the link text is the alternative description for the image;
- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;

- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- As a rough guideline, for buttons use the same text that is shown by the image;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **ATTENTION:** the ALT attribute should be the empty string ("") in cases where the image is already described by surrounding text;

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.15 Link image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Images
 - Links
 - Manual
 - ALL TESTS
- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It checks, for images embedded within a link, if their description is equivalent to the image itself.

This is manual test.

- **Issue Description:**

The image is used as label of a link. It should have a textual description (i.e. ALT attribute of the IMG tag and/or text surrounding the image itself) that is equivalent to the image.

- **How to fix:**

Please check if the current ALT description conveys the same meaning of the image in the context of the text surrounding the image.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what is the effect of following the link, what is the link destination. (Imagine hearing the link description over the phone.)

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what is the effect of following the link, what is the link destination.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not least, users of cellular phones, PDAs, browsers installed in cars, etc.).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen's Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.16 Map image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This test is a **UsableNet extension** that considers only images used as image maps (aka hotspots).

The test checks if image maps embedded in the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The image used as an image map (aka hotspots) does not have a **valid** textual description. The **textual description** is the value of the ALT attribute. A description is **valid** if it:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;

- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT is empty:** ALT of image is the empty string "" and no text surrounds it
- **ALT is blank:** ALT of image is the blank string " " and no text surrounds it
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file and no text surrounds it
- **ALT with filename:** ALT of image describes only the filename of the image and no text surrounds it
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text and no text surrounds it

- **How to fix:**

Add the ALT attribute to the image map (i.e. to the IMG tag).

Remember that:

- the ALT attribute should describe the purpose of the image map as a group of links. Each single hotspot inside the image should have its own ALT description. The ALT of the image map is usually shown (for example by lynx) as a label for the entire group of links. Imagine to listen over the phone the page content and the image description;
- you can check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- Be brief. Consider that for image maps that are repeated on every page of the site people would have to listen to the same description over and over;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).;

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

For an image map, ALT descriptions are important because when the page is accessed with a non-graphical browser (reading browser, text-only browser, cell phone, etc) what the user gets is the ALT description, if it is there. Otherwise s/he may get only the file name of the image.

The ALT description is therefore a powerful means to support navigation.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

ALT descriptions are also displayed before the associated image, which is helpful when image download and display takes several seconds.

Finally, adding keywords to the ALT attribute can also improve the page listing in some search engine.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 2](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.17 Thumbnail image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This test is a **UsableNet extension** that considers only images used as link labels.

The test checks if thumbnail images (i.e. images that are links to enlarged versions of the same images) embedded in the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The image included in the label of the link does not have a **valid** textual descriptions. The **textual description** is the value of the ALT attribute or the text surrounding the image. A description is **valid** if it:

- is not the empty string (“”);
- is not a string with 1 or more spaces (“ ”);
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT is empty:** ALT of image is the empty string ”” and no text surrounds it
- **ALT is blank:** ALT of image is the blank string ” ” and no text surrounds it
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file and no text surrounds it
- **ALT with filename:** ALT of image describes only the filename of the image and no text surrounds it
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text and no text surrounds it

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the ALT attribute should describe the link destination, i.e. the full-scale image
- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;

- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).;
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **ATTENTION:** the ALT attribute should be the empty string ("") in cases where the image is already described by surrounding text;

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds. It is also often displayed as a tooltip by the browser. This is particularly useful for thumbnail images so users can get a description of the full image before downloading it.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.18 Thumbnail image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Images
 - Links
 - Manual
 - ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It checks, for thumbnail images (i.e. images that are links to enlarged versions of the same images), if their description is equivalent to the image itself.

This is manual test.

- **Issue Description:**

The image is used as label of a link. It should have a textual description (i.e. ALT attribute of the IMG tag and/or text surrounding the image itself) that is equivalent to the image.

- **How to fix:**

Please check if the current ALT description conveys the same meaning of the image in the context of the text surrounding the image.

The image description should convey some information about the full-scale image. (Imagine hearing the link description over the phone.)

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what is the effect of following the link, what is the link destination.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not least, users of cellular phones, PDAs, browsers installed in cars, etc.).

- **Learn More:**

- W3C on [how to include objects and images](#);

- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen's Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.19 Thumbnail with valid LONGDESC/D-LINK [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The rule implements WAI guideline 1.1 "Provide a text equivalent for every non-text element" corresponding to rule 1194.22(a) of the [508 standard](#). More specifically it implements and extends [technique 1.1.2](#) of the WAI [AERT document](#).

The test is a UsableNet extension to those standards because it applies only to thumbnail images (i.e. small images that are linked to full-scale images).

It checks if the small image has a **LONGDESC** attribute or if it is followed by a **D-link** element (that is `D</>`).

The test also checks if either the LONGDESC attribute or D-link href attribute contain a valid URL pointing to an HTML file.

Finally, in case both attributes are defined, the test checks if they are the same.

- **Issue Description:**

The thumbnail image has no valid **LONGDESC** or **D-link** element or, if both are present, they point to different files.

- **Specific Issues:**

- **LONGDESC attribute not defined:** Missing both the LONGDESC attribute and the D-link element
- **LONGDESC file does not exist:** Invalid LONGDESC attribute: mentioned file does not exist

- **LONGDESC file is not HTML:** Invalid LONGDESC attribute: mentioned file is not an HTML file
- **LONGDESC file is empty:** Invalid LONGDESC attribute: mentioned file is empty
- **LONGDESC URL is dead:** Invalid LONGDESC attribute: its URL is dead
- **LONGDESC URL generated bad response:** Invalid LONGDESC attribute: got an error response from the server of its URL
- **LONGDESC URL has a bad protocol:** Invalid LONGDESC attribute: it is not a local file nor an HTTP URL
- **D-link file does not exist:** Invalid D-link: mentioned file does not exist
- **D-link file is not HTML:** Invalid D-link: mentioned file is not an HTML file
- **D-link file is empty:** Invalid D-link: mentioned file is empty
- **D-link url is dead:** Invalid D-link: its URL is dead
- **D-link bad response:** Invalid D-link: got an error response from the server of its URL
- **D-link bad protocol:** Invalid D-link: it is not a local file nor an HTTP URL
- **D-link and LONGDESC are different:** LONGDESC and D-link point to different files

- **How to fix:**

Please define a proper **LONGDESC** attribute for the image (by specifying the address – URL – of an HTML file).

Since not many browsers at the moment support the **LONGDESC** attribute, you may also want to place, near the image, a so called D-link, a normal textual link with label "D" pointing to an HTML page providing a full image description. Example:

```
<A href="large_picture.jpg">
  <IMG src="chart.gif" alt="chart of beverages distribution"
  longdesc="chart.html">
</A>
<A href="chart.html">D</A>
```

Alternatively, placing a rich textual description close to the image (like a caption) is a viable solution. In such a case you don't need the **LONGDESC** attribute or the D-link.

- **Why to fix:**

The ALT attribute cannot be used to provide a full length description of the content of an image. Think of a diagram, a histogram or a chart. Or even a product picture in an online catalogue. To describe its content you need more flexibility.

This is especially needed for thumbnails since a user might want to get a description of a large image before trying to download it.

The **LONGDESC** attribute can be used to provide a long description of the associated image that would not fit in the ALT attribute. By including **LONGDESC="any_HTML_file"** in your IMG tag you can link the image to the HTML file containing a formatted description of the image. The long description (unlike the ALT attribute) can contain HTML code, with links to other resources, formatting instructions, etc.

However, since not many browsers at the moment support the **LONGDESC** attribute, you may also want to place, near the small image, a so called D-link, a normal textual link with label "D" pointing to an HTML page providing a full image description. Example:

```
<A href="large_picture.jpg">
  <IMG src="chart.gif" alt="chart of beverages distribution"
  longdesc="chart.html">
</A>
<A href="chart.html">D</A>
```

Alternatively, placing a rich textual description close to the image (like a caption) is a viable solution. In such a case you don't need the **LONGDESC** attribute or the D-link.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not the least, users of cellular phones, PDAs, browsers installed in cars, etc.)

- **Learn More:**

- HTML 4.0 standard on the [IMG tag](#);
- HTML 4.0 standard on the [LONGDESC attribute](#);
- W3C on [how to define the LONGDESC attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- some examples about [LONGDESC and D-links](#);
- W3C on [how to define the ALT attribute](#).

2.20 Button image with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images

- Links
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This test is a **UsableNet extension** that considers only images used as link labels inside navigation bars.

The test checks if button images embedded in navigation bars within the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The image included in the label of the link does not have a **valid** textual descriptions. The **textual description** is the value of the ALT attribute. It is **valid** if it:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT is empty:** ALT of image is the empty string ""
- **ALT is blank:** ALT of image is the blank string " "
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file
- **ALT with filename:** ALT of image describes only the filename of the image
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the ALT attribute (and the link text, if any) should describe the link destination, i.e. what the user would find after clicking on the link;
- the description should explain the role of the image in the page. Imagine to listen over the phone the page content and the image description;
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- Be brief. Consider that images like logos are repeated on every page of the site, and people would have to listen to the same description over and over;
- As a rough guideline, for buttons use the same text that is shown by the image;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).
- Images used solely for decorative purposes (called spacers) should include an empty ALT string ("") so that they are not considered by screen-readers. Similarly for images like bullets;
- **ATTENTION:** the ALT attribute should be the empty string ("") in cases where the image is already described by surrounding text;

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

ALT descriptions are displayed before the associated image, which is helpful when image download and display takes several seconds.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);

- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.21 Button image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- Manual
- ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It checks, for images embedded within navigation bars, if their description is equivalent to the image itself.

This is manual test.

- **Issue Description:**

The image is used as label of a link within a navigation bar. It should have a textual description (i.e. ALT attribute of the IMG tag and/or text surrounding the image itself) that is equivalent to the image. An **equivalent** textual description conveys the same meaning as the image.

- **How to fix:**

Please check that the current ALT description conveys the same meaning of the image.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what is the effect of following the link, what is the link destination. (Imagine hearing the link description over the phone.)

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The image description should convey some information about the image and explain the role the image plays in the page: why it's there, what is the effect of following the link, what is the link destination.

Notice that adding a good alternative description to images is very important nowadays as many users, for one reason or the other, are not able to take advantage of graphics (not least, users of cellular phones, PDAs, browsers installed in cars, etc.).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen's Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.22 Hidden link with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- ALL TESTS

- **Test Description:**

The test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)" ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [evaluation and repair document](#). This test is a **UsableNet extension** that considers only images that are invisible and that are used for providing a link when images are not shown or cannot be seen. Such a link can be used only when images are not shown (i.e. hidden links).

The test checks if images embedded in hidden links within the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The invisible image included in the hidden link does not have a **valid** textual descriptions. The **textual description** is the value of the ALT attribute. It is **valid** if it:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for image
- **ALT is empty:** ALT of image is the empty string ""
- **ALT is blank:** ALT of image is the blank string " "
- **ALT with html tags:** ALT of image contains HTML tags
- **ALT too long:** ALT of image is too long
- **ALT with size:** ALT of image describes only the size of the image file
- **ALT with filename:** ALT of image describes only the filename of the image
- **ALT with placeholder text:** ALT of image appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the IMG tag.

Remember that:

- the ALT attribute (and the link text, if any) should describe the link destination, i.e. what the user would find after clicking on the link;
- Imagine to listen over the phone the page content and the image description;
- you can also check the text-only version of a page by using the **LIFT Online** service at www.usablenet.com/products_services service that presents a page as would be displayed by the lynx textual browser;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- Too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less that 10 words and 64 characters).

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

For hidden links, the ALT should be a description of the destination of the link; in other words, it should describe the effects of following that link.

Adding keywords to the ALT attribute can also improve the page listing in some search engine.

You can check the text-only version of a page by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 8](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.23 Hidden link image with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- Manual
- ALL TESTS

- **Test Description:**

The test implements WAI guideline 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

It prompts the user to check, for invisible images embedded within hidden links, if their description is a proper description of the link.

This is manual test.

- **Issue Description:**

The image is used within a hidden link. It should have a textual description (i.e. ALT attribute of the IMG tag) that describes properly the link destination.

- **How to fix:**

Please check if the current ALT description is a proper description of the link destination.

Imagine hearing the description over the phone. It should be like hearing the label of a textual link.

- **Why to fix:**

The ALT attribute describes the hidden link so that users without graphics-enabled browsers can still effectively navigate the page and take full advantage of the page content.

The description of the hidden link should convey some information about the link destination: it is as if it were a text label of a link.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on the [IMG tag](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 8](#) from IBM Web accessibility checklist - version 3.1
- a tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- J. Nielsen's Alertbox on [ALT descriptions](#);
- the [508 standard](#).

2.24 INPUT with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Forms
- ALL TESTS

- **Test Description:**

The rule implements WAI guideline 1.1 "Provide a text equivalent for every non-text element" corresponding to rule 1194.22(a) of the [508 standard](#). More specifically it implements [technique 1.1.3](#) of the WAI [AERT document](#).

The rule checks if a form has a button associated to an image (that is an element INPUT with type="image"). In such a case the rule checks if the ALT attribute is **valid**.

- **Issue Description:**

Image used as a button in the form does not have a **valid** textual descriptions.

A **valid description** is a string value of the ALT attribute that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for button image
- **ALT not empty:** ALT of button image should be empty or blank and isn't
- **ALT is empty:** ALT of button image is the empty string ""
- **ALT is blank:** ALT of button image is the blank string " "
- **ALT too long:** ALT of button image is too long
- **ALT with html tags:** ALT of button contains HTML tags
- **ALT with size:** ALT of button image describes only the size of the image file
- **ALT with filename:** ALT of button image describes only the filename of the image
- **ALT with placeholder text:** ALT of button image appears to contain only placeholder text

- **How to fix:**

Please add the ALT attribute to the INPUT element. Remember that:

- the description should explain the effect of clicking on the button;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).

If the server takes different actions depending on the location being clicked consider the following alternative approaches:

- use multiple submit buttons (each with its own image) in place of a single graphical submit button;
- use a client-side image map together with scripting.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

If the button image does not have a textual description, users of non-graphical browsers will have no idea what is the effect of clicking on that button.

In addition, when the mouse is used to click on the image, the form is submitted and the point coordinates are passed to the server. If the server takes different actions depending on the location being clicked, users of non-graphical browsers cannot take advantage of it.

- **Learn More:**

- the HTML 4.0 standard on the [INPUT element](#);
- the HTML 4.0 standard on [forms](#);
- the HTML 4.0 standard on [possible types of INPUTs](#);
- the W3C on [how to define the ALT attribute](#);
- a W3C discussion on [blank vs empty ALT strings](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- the HTML 4.0 standard on the [IMG tag](#).

2.25 INPUT with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Manual
- Forms
- ALL TESTS

- **Test Description:**

The rule implements WAI guideline 1.1 "Provide a text equivalent for every non-text element" corresponding to rule 1194.22(a) of the [508 standard](#). More specifically it implements [technique 1.1.3](#) of the WAI [AERT document](#).

The rule checks if a form has a button associated to an image (that is an element INPUT with type="image"). In such a case the rule checks if the ALT attribute is **equivalent** to the image.

- **Issue Description:**

Image used as a button in the form has a **valid** description. It should have an **equivalent** textual description.

- **How to fix:**

Please test the content of the ALT attribute to the INPUT element. Remember that the description should explain the effect of clicking on the button.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

If the button image does not have an **equivalent** textual description, users of non-graphical browsers will have no idea what is the effect of clicking on that button.

- **Learn More:**

- the HTML 4.0 standard on the [INPUT element](#);
- the HTML 4.0 standard on [forms](#);
- the HTML 4.0 standard on [possible types of INPUTs](#);
- the W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a W3C discussion on [blank vs empty ALT strings](#);
- the HTML 4.0 standard on the [IMG tag](#).

2.26 Form Button with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Forms

– ALL TESTS

- **Test Description:**

The rule implements WAI guideline 1.1 "Provide a text equivalent for every non-text element" corresponding to rule 1194.22(a) of the [508 standard](#). More specifically it implements [technique 1.1.3](#) of the WAI [AERT document](#).

The rule checks if a form has a `BUTTON` element with an image as content. In such a case the rule checks if the `ALT` attribute is **valid**.

- **Issue Description:**

Image used as a button in the form does not have a **valid** textual descriptions.

A **valid description** is a string value of the `ALT` attribute that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No `ALT` defined for button image
- **ALT not empty:** `ALT` of button image should be empty or blank and isn't
- **ALT is empty:** `ALT` of button image is the empty string ""
- **ALT is blank:** `ALT` of button image is the blank string " "
- **ALT too long:** `ALT` of button image is too long
- **ALT with html tags:** `ALT` of button contains `HTML` tags
- **ALT with size:** `ALT` of button image describes only the size of the image file
- **ALT with filename:** `ALT` of button image describes only the filename of the image
- **ALT with placeholder text:** `ALT` of button image appears to contain only placeholder text

- **How to fix:**

Please add the `ALT` attribute to the `IMG` element. Remember that:

- the description should explain the effect of clicking on the button;
- `ALT` descriptions are not interpreted by browsers and should not include `HTML` tags. Embedded tags can only confuse users and maybe search engines as well;

- too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

If the button image does not have a textual description, users of non-graphical browsers will have no idea what is the effect of clicking on that button.

- **Learn More:**

- the HTML 4.0 standard on the [BUTTON element](#);
- the HTML 4.0 standard on [forms](#);
- the W3C on [how to define the ALT attribute](#);
- a W3C discussion on [blank vs empty ALT strings](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- the HTML 4.0 standard on the [IMG tag](#).

2.27 Form Button with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Manual
- Forms
- ALL TESTS

- **Test Description:**

The rule implements WAI guideline 1.1 "Provide a text equivalent for every non-text element" corresponding to rule 1194.22(a) of the [508 standard](#). More specifically it implements [technique 1.1.3](#) of the WAI [AERT document](#).

The rule checks if a form has a BUTTON element with an image as content. In such a case the rule suggests the user to check if the ALT attribute is **equivalent** to the image.

- **Issue Description:**

Image used as a button in the form has a **valid** description. It should have an **equivalent** textual description.

- **How to fix:**

Please test the content of the ALT attribute to the IMG element.

Make sure that the description should explain the effect of clicking on the button.

- **Why to fix:**

The ALT attribute describes the associated image so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

If the button image does not have an **equivalent** textual description, users of non-graphical browsers will have no idea what is the effect of clicking on that button.

- **Learn More:**

- the HTML 4.0 standard on the [BUTTON element](#);
- the HTML 4.0 standard on [forms](#);
- the HTML 4.0 standard on [possible types of INPUTs](#);
- the W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- a W3C discussion on [blank vs empty ALT strings](#);
- the HTML 4.0 standard on the [IMG tag](#).

2.28 SCRIPT with valid NOSCRIPT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- ALL TESTS

- **Test Description:**

The rule implements WAI guideline 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.

This rule also corresponds to rule 1194.22(a) of the [508 standard](#). More specifically it implements [technique 1.1.10](#) of the WAI [AERT document](#).

The rule checks if all SCRIPTS element have a corresponding NOSCRIPT element with valid content. The exceptions are SCRIPT elements embedded within HEAD (NOSCRIPT

is not allowed within HEAD).

If such scripts are found then the rule prompts for manually checking if the script produces an interaction that is not accessible.

- **Issue Description:**

The page contains a SCRIPT element that does not have an associated valid **NOSCRIPT**. A NOSCRIPT is a script alternative that contains HTML and is enabled when a script cannot be executed.

Its is valid if:

- the **NOSCRIPT** directly follows the **SCRIPT** element;
- the content of the **NOSCRIPT** element should not be empty. It can include any HTML instruction.

Notice that the 508 standard **does not** require that all SCRIPTS are followed by corresponding NOSCRIPTS (though being a suggested technique according to WAI WCAG 1.0).

The 508 standard requires that when pages utilize scripting languages to display content, or to create interface elements, the information provided by the script is identified with functional text that can be read by assistive technology.

There are two kinds of scripts that require special care (and using the NOSCRIPT tag may help):

- **rollovers**: if a script changes an image on the screen when the user moves the cursor over the image and it does not indicate (through text readable by a screen reader) that it has changed the content of the page, the script cannot be made accessible.
- **keyboard inoperable scripts**: if the event triggering the script cannot be activated through keyboard, then the script cannot be made accessible.

- **Specific Issues:**

- **missing NOSCRIPT**: Missing NOSCRIPT: there is no NOSCRIPT following the SCRIPT
- **NOSCRIPT is empty**: Invalid NOSCRIPT: the NOSCRIPT is empty

- **How to fix:**

Please check that the information provided by the script can be also provided by text that can be read by assistive technology.

The easiest way to check if a script is accessible is to try to view the page with a browser where scripts and image loading are temporarily disabled (please consult the browser's documentation to learn how to disable scripts and image loading).

Secondarily, using the page without the mouse (by tabbing through elements and using exclusively the keyboard), gives you an idea of the role actually played by the script in the interaction.

If not, consider adding a NOSCRIPT element to the SCRIPT. It can contain any HTML tag.

The content of a NOSCRIPT should enable users that for whatever reason cannot execute the script to receive the same information and effects as users who are able to execute the script. In particular they should be able to access the same content and same interaction opportunities, including links.

- **Why to fix:**

One way to provide a **text equivalent** to a SCRIPT element is with **NOSCRIPT**. The content of this element is rendered when scripts are not enabled.

The NOSCRIPT element allows authors to provide alternate content when a script is not executed. The content of a NOSCRIPT element should only be rendered by a script-aware user agent in the following cases:

- The browser is configured not to evaluate scripts;
- The browser doesn't support a scripting language invoked by the script.

Users of screen-readers and reading browsers would not be able to take advantage of scripts that affect the graphical interface of a webpage.

Users not using a mouse or joystick would not be able to take advantage of scripts that open windows, dialogs, menus.

Notice that there are also many other cases where scripts cannot be run by browsers, and these scenarios are going to be more and more frequent in the near future:

- PDAs and cellular phones are not able to execute scripts;
- Not all browsers in diverse operating systems (Windows, MacOS, Linux) are able to execute scripts written in a given language (for example Visual Basic scripts are not run on Netscape on Linux or PDA's);
- Some users of graphical browsers might disable Javascript for security reasons.

- **Learn More:**

- W3C on how to use the [NOSCRIPT element](#);
- W3C on how to use [scripts](#);
- W3C/WAI techniques for making [accessible SCRIPTS](#) elements;
- W3C/WAI [guideline 1](#);
- W3C/WAI [checkpoint 1.1](#).
- [checkpoint 5](#) from IBM Web accessibility checklist - version 3.1

2.29 SCRIPT with equivalent NOSCRIPT [WCAG 01.1]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Scripts
 - Manual
 - ALL TESTS

- **Test Description:**

The rule implements WAI guideline 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.

It corresponds to rule 1194.22(a) of the [508 standard](#).

More specifically it implements [technique 1.1.10](#) of the WAI [AERT document](#).

The rule checks if all SCRIPTS elements with a corresponding valid NOSCRIPT element, have a textual content that is equivalent to the script itself.

This is manual test. (A NOSCRIPT is **valid** if it exists, it is close to the SCRIPT it refers to, and it is not empty.)

- **Issue Description:**

Script included in the page has a **valid** NOSCRIPT element whose content has to be **equivalent** to the script.

(A NOSCRIPT is **valid** if it exists, it is close to the SCRIPT it refers to, and it is not empty.)

- **How to fix:**

Please check that the content of the current NOSCRIPT conveys the same meaning of the associated SCRIPT.

The content of a NOSCRIPT should enable users that for whatever reason cannot execute the script to receive the same information and achieve the same effects as users who are able to execute the script. In particular they should be able to access the **same content** and same interaction opportunities, including **links**.

NOSCRIPT can contain any HTML tag.

- **Why to fix:**

One way to provide a **text equivalent** to a SCRIPT element is with **NOSCRIPT**. The content of this element is rendered when scripts are not enabled.

The NOSCRIPT element allows authors to provide alternate content when a script is not executed. The content of a NOSCRIPT element should only be rendered by a script-aware user agent in the following cases:

- The browser is configured not to evaluate scripts;
- The browser doesn't support a scripting language invoked by the script.

Users of screen-readers and reading browsers would not be able to take advantage of scripts that affect the graphical interface of a webpage.

Users not using a mouse or joystick would not be able to take advantage of scripts that open windows, dialogs, menus.

Notice that there are also many other cases where scripts cannot be run by browsers, and these scenarios are going to be more and more frequent in the near future:

- PDAs and cellular phones are not able to execute scripts;
- Not all browsers in diverse operating systems (Windows, MacOS, Linux) are able to execute scripts written in a given language (for example Visual Basic scripts are not run on Netscape on Linux);
- Some users of graphical browsers disable Javascript for security reasons.

- **Learn More:**

- W3C on how to use the [NOSCRIPT element](#);
- W3C on how to use [scripts](#);
- W3C/WAI techniques for making [accessible SCRIPTS](#) elements;
- W3C/WAI [guideline 1](#);
- W3C/WAI [checkpoint 1.1](#);
- [checkpoint 5](#) from IBM Web accessibility checklist - version 3.1
- J. Nielsen on [textual equivalents](#).

2.30 Hotspot with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images

- Links
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)). More specifically it implements [technique 1.1.9](#) of the W3C/WAI "Evaluation and Repair guidelines".

It checks if hotspots (i.e. AREA elements included in client-side image maps) have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

The imagemap included in this document has a hotspot (i.e. AREA elements) for which there is no **valid** textual descriptions. A **valid description** is a string value of the ALT attribute that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the image;
- does not say only which is the size of the image file;
- is not longer than 150 characters (this is actually only a suggestion: it is not required by WCAG 1.0 nor 508 standard).

- **Specific Issues:**

- **ALT not defined:** No ALT defined for hotspot
- **ALT is empty:** ALT of hotspot is the empty string ""
- **ALT is blank:** ALT of hotspot is the blank string " "
- **ALT too long:** ALT of hotspot is too long
- **ALT with html tags:** ALT of hotspot contains HTML tags
- **ALT with size:** ALT of hotspot describes only the size of the image file
- **ALT with filename:** ALT of hotspot describes only the filename of the image
- **ALT with placeholder text:** ALT of hotspot appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the hotspot (i.e. AREA element). Remember that:

- the description should explain the link destination or why the user would want to follow it;

- don't mention the mechanics, like "click here", but describe the destination of the link and its role for the user;
- ALT descriptions are not interpreted by browsers and should not include HTML tags. Embedded tags can only confuse users and maybe search engines as well;
- too long ALT descriptions may be truncated by browsers and increase the time required to download the page (J. Nielsen in his book www.useit.com/jakob/webusability suggests to use less than 10 words and 64 characters).

- **Why to fix:**

ALT attributes are the means to provide text equivalents for image maps. Doing so is important since image maps convey visual information that is crucial for navigation. Failing to do that may lead to sites that are not navigable if images cannot be shown.

The ALT attribute should describe the associated area of the image so that users without graphics-enabled browsers can still effectively navigate the page. Without ALT descriptions such users would not be able to navigate through the image map.

Each hotspot (i.e. AREA element) of the image map should have its own ALT attribute. For example (taken and revised from [a tutorial](#)):

```
<IMG src="map.gif" alt="navigation" usemap="#navigation">
<MAP name="navigation">
<AREA coords="1,1,40,100" alt="home" href="../index.html">
<AREA coords="1,100,40,180" alt="products" href="products.html">
</MAP>
```

As with other links, the link text should make sense when read out of context. Good link text should not be overly general; don't use "click here." Not only is this phrase device-dependent (it implies a pointing device), but it says nothing about what is to be found if the link is followed. Instead of "click here", link text should indicate the nature of the link target, as in "more information about company products" or "text-only version of this page".

The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever. Consider that there are many users that by choice or necessity use text-only browsers with screen-readers or reading browsers: visually impaired persons, people using the web over a phone, people using a reading browser in the car, etc.

Until user agents are able to render text equivalents for client-side image map links, by providing redundant textual links for each active region of a client-side image map you can make your page accessible to users that cannot see the graphics. However, by providing the ALT description for AREA elements you make your page already compliant with respect to these new browsers.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on [image maps](#);
- W3C/WAI technique 1.1.9 on [valid ALT for AREA](#);
- W3C Accessibility guidelines on [accessibility of image maps](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- the [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.31 Hotspot with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Links
- Manual
- ALL TESTS

- **Test Description:**

This test implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)). More specifically it implements [technique 1.1.9](#) of the W3C/WAI "Evaluation and Repair guidelines".

It checks if hotspots (i.e. AREAs elements included in client-side image maps) have an **equivalent** alternative description as an ALT attribute.

- **Issue Description:**

The hotspot of an image map included in this page has a **valid** textual descriptions (i.e. ALT attribute).

These descriptions should also be **equivalent** to the information shown by the image corresponding to the hotspot.

- **How to fix:**

Please check that the ALT attribute of the hotspot (i.e. AREA element) is meaningful. Remember that:

- it should explain the link destination or why the user would want to follow it;
- don't mention the mechanics, like "click here", but describe the destination of the link and its role for the user.

- **Why to fix:**

ALT attributes are the means to provide text equivalents for image maps. Doing so is important since image maps convey visual information that is crucial for navigation. Failing to do that may lead to sites that are not navigable if images cannot be shown.

The ALT attribute should describe the associated area of the image so that users without graphics-enabled browsers can still effectively navigate the page. Without **equivalent** ALT descriptions such users would not be able to navigate through the imagemap.

As with other links, the link text should make sense when read out of context. Good link text should not be overly general; don't use "click here." Not only is this phrase device-dependent (it implies a pointing device), but it says nothing about what is to be found if the link is followed. Instead of "click here", link text should indicate the nature of the link target, as in "more information about sea lions" or "text-only version of this page".

The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever. Consider that there are many users that by choice or necessity use text-only browsers with screen-readers or reading browsers: visually impaired persons, people using the web over a phone, people using a reading browser in the car, etc.

Until user agents are able render text equivalents for client-side image map links, by providing redundant textual links for each active region of a client-side image map you can make your page accessible to users that cannot see the graphics. However, by providing the ALT description for AREA elements you make your page already compliant with respect to these new browsers.

- **Learn More:**

- W3C on [how to include objects and images](#);
- HTML 4.0 standard on [image maps](#);
- W3C/WAI technique 1.1.9 on [valid ALT for AREA](#);
- W3C Accessibility guidelines on [accessibility of image maps](#);
- W3C on [how to define the ALT attribute](#);
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1
- tutorial on the [use of ALT text in IMGs](#);
- another tutorial on [alternative content for graphics](#) rich of examples;
- a discussion on [empty vs blank ALT descriptions](#);
- [508 standard](#);
- J. Nielsen's Alertbox on [ALT descriptions](#).

2.32 APPLET with valid ALT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [AERT document](#).

It checks if applets embedded in the page have a **valid** alternative description as an ALT attribute.

- **Issue Description:**

Applets included in this document don't have corresponding **valid** textual descriptions. A **valid description** is a string value of the ALT attribute that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the applet;
- does not say only which is the size of the applet file.

- **Specific Issues:**

- **ALT not defined:** No ALT defined for applet
- **ALT is empty:** ALT of applet is the empty string ""
- **ALT is blank:** ALT of applet is the blank string " "
- **ALT with size:** ALT of applet describes only the size of the applet file
- **ALT with image filename:** ALT of applet describes only the filename of an image
- **ALT with applet filename:** ALT of applet describes only the filename of the applet
- **ALT with placeholder text:** ALT of applet appears to contain only placeholder text

- **How to fix:**

Add the ALT attribute to the APPLET tag.

The ALT description should convey the same information as the applet and it should explain the role the applet plays in the page: why it's there, what does it represent, how does it present the information.

- **Why to fix:**

There are many cases where applets cannot be run by browsers:

- PDAs and cellular phones are not able to execute applets;
- some applets use libraries written for a specific operating system (Windows, MacOS, Linux);
- some users of graphical browsers disable applets for security reasons.

Provide a text equivalent describing the behavior of the applet using the "alt" attribute and the content in the APPLET element. This enables the content to transform gracefully for those user agents that only support one of the two mechanisms ("alt" or content).

- **Learn More:**

- the W3C on [including an applet](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the W3C on [how to define the ALT attribute](#);
- the [508 standard](#).
- [checkpoint 6](#) from IBM Web accessibility checklist - version 3.1

2.33 APPLET with equivalent ALT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- Manual
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [AERT document](#).

It checks if the applets embedded in the page that have a **valid** ALT attribute and a **valid** textual content, have an **equivalent** alternative description.

This is a manual test.

- **Issue Description:**

Applets included in this document should have corresponding **equivalent** textual descriptions (for the "alt" attribute and the applet's content).

- **How to fix:**

Please check the applet's textual description (the ALT attribute value and the applet's content): it should convey the same information as the applet, and it should explain the role the applet plays in the page: why it's there, what does it represent, how does it present the information.

- **Why to fix:**

There are many cases where applets cannot be run by browsers:

- PDAs and cellular phones are not able to execute applets;
- some applets use libraries written for a specific operating system (Windows, MacOS, Linux);
- some users of graphical browsers disable applets for security reasons.

Providing a text describing the behavior of the applet, enables the content to transform gracefully for those user agents that only support one of the two mechanisms ("alt" or content).

- **Learn More:**

- W3C on [including an applet](#);
- W3C Accessibility guidelines on [ALT descriptions](#);
- W3C on [how to define the ALT attribute](#);
- the [508 standard](#).
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1

2.34 APPLET with valid CONTENT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- ALL TESTS

- **Test Description:**

The rule implements WAI checkpoint 1.1 "Provide a text equivalent for every non-text element (e.g., via alt, longdesc, or in element content)." ([guideline 1](#) of W3C/WAI corresponding to rule 1194.22(a) of the [508 standard](#)).

More specifically it implements [checkpoint 1.1](#) of the W3C/WAI [AERT document](#).

It checks if applets embedded in the page have a **valid** textual content, that can be used by browsers that are unable to execute the applet.

- **Issue Description:**

Applets included in this document don't have corresponding **valid** textual descriptions defined in their content. A **valid description** is a string value for the CONTENT that:

- is not the empty string ("");
- is not a string with 1 or more spaces (" ");
- is not the name of the file containing the applet;
- does not say only which is the size of the applet file.

The 'content' of an APPLET element is the HTML that you can specify between <APPLET> and </APPLET>:

```
<APPLET ... >
  applet for login into our <b>superprotected area</b>
</APPLET>
```

This content is shown by the browser in those cases when the applet cannot be executed.

- **Specific Issues:**

- **Content not defined:** No CONTENT defined for applet
- **Content is empty:** CONTENT of applet is the empty string ""
- **Content is blank:** CONTENT of applet is the blank string " "
- **Content with size:** CONTENT of applet describes only the size of the applet file
- **Content with image filename:** CONTENT of applet describes only the filename of an image
- **Content with applet filename:** CONTENT of applet describes only the filename of the applet
- **Content with placeholder text:** CONTENT of applet appears to contain only placeholder text

- **How to fix:**

Add a content to the APPLET tag.

The textual description should convey the same information as the applet and it should explain the role the applet plays in the page: why it's there, what does it represent, how does it present the information.

- **Why to fix:**

There are many cases where applets cannot be run by browsers:

- PDAs and cellular phones are not able to execute applets;
- some applets use libraries written for a specific operating system (Windows, MacOS, Linux);
- some users of graphical browsers disable applets for security reasons.

Provide a text equivalent describing the behavior of the applet using the "alt" attribute and the content in the APPLET element. This enables the content to transform gracefully for those user agents that only support one of the two mechanisms ("alt" or content).

- **Learn More:**

- the W3C on [including an applet](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the W3C on [how to define the ALT attribute](#);
- the [508 standard](#).
- [checkpoint 6](#) from IBM Web accessibility checklist - version 3.1

2.35 Image OBJECT with valid CONTENT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.1.

The rule checks if the page contains OBJECT elements that embed images and don't have a valid description between <OBJECT> and </OBJECT> tags.

- **Issue Description:**

The page contains OBJECT elements embedding images that do not have a **valid description** between <OBJECT> and </OBJECT> tags.

A **valid description** is an html fragment between the tags <OBJECT> and </OBJECT> that:

- contains at least one word;

- contains another object;
- contains an IMG element with valid alt-text;
- contains a link to a valid alternative representation;
- doesn't contain [placeholder text](#).

- **Specific Issues:**

- **Content is empty:** Alternative content of object is the empty string ""
- **Content is blank:** Alternative content of object is the blank string " "
- **Content with size:** Alternative content of object describes only the size of the image file
- **Content with filename:** Alternative content of object describes only the filename of the image
- **Content with placeholder text:** Alternative content of object appears to contain only placeholder text

- **How to fix:**

Insert a **meaningful** description inside <OBJECT> section.

The alternative textual description should explain the content of the image and its role in the document.

- **Why to fix:**

The OBJECT element offers a mechanism for specifying alternate object renderings; each embedded OBJECT declaration may specify alternate content types. If a browser cannot render the outermost OBJECT, it tries to render its contents, which may be another OBJECT element, etc.

At least one object in such a chain should be accessible by providing a valid equivalent description so that users without graphics-enabled browsers can still effectively navigate the page. The advent of hand-held, text-only browsers makes the use of the ALT descriptions more important now than ever.

Defining a textual description for an object can also improve the page listing in some search engine.

- **Learn More:**

- the W3C on [Objects, Images, and Applets](#);
- the W3C on [Generic inclusion: the OBJECT element](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#).
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1

2.36 Image OBJECT with equivalent CONTENT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Images
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.1.

The rule checks if the OBJECT elements contained in the page that embed images, have an **equivalent** description between <OBJECT> and </OBJECT> tags.

This is a manual test.

- **Issue Description:**

The OBJECT elements embedding images in this document should have corresponding **equivalent** description between <OBJECT> and </OBJECT> tags.

- **How to fix:**

Insert an **equivalent meaningful** textual description inside <OBJECT> section, or insert an image.

The alternative textual description should explain the content of the image and its role in the document.

- **Why to fix:**

The OBJECT element offers a mechanism for specifying alternate object renderings; each embedded OBJECT declaration may specify alternate content types. If a browser cannot render the outermost OBJECT, it tries to render its contents, which may be another OBJECT element, etc.

Objects that embed images should have corresponding **equivalent** descriptions or images to enable users without graphics enabled browsers to still effectively navigate the page.

The image description should convey the same information as the image and it should explain the role the image plays in the page: why it's there, what does it represent, how does it present the information.

- **Learn More:**

- the W3C on [Objects, Images, and Applets](#);

- the W3C on [Generic inclusion: the OBJECT element](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#);
- a tutorial on the [use of ALT text in IMGs](#).
- [checkpoint 1](#) from IBM Web accessibility checklist - version 3.1

2.37 Audio/video OBJECT with valid CONTENT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.1.

The rule checks if the page contains OBJECT elements embedding audio or video files without a **valid** description between <OBJECT> and </OBJECT> tags.

- **Issue Description:**

The page contains OBJECT elements embedding audio or video files without a **valid description** between <OBJECT> and </OBJECT> tags.

- **Specific Issues:**

- **Content is empty:** Alternative content of object is the empty string ""
- **Content is blank:** Alternative content of object is the blank string " "
- **Content with size:** Alternative content of object describes only the size of the audio/video file
- **Content with filename:** Alternative content of object describes only the name of the audio or video file
- **Content with placeholder text:** Alternative content of object appears to contain only placeholder text

- **How to fix:**

Insert a **meaningful** description inside <OBJECT> section.

The alternative textual description should be a transcription of the audio/video file.

- **Why to fix:**

The OBJECT element's design offers a mechanism for specifying alternate object renderings; each embedded OBJECT declaration may specify alternate content types. If a browser cannot render the outermost OBJECT, it tries to render the contents, which may be another OBJECT element, etc.

Objects included in this document that embed audio or video files, don't have corresponding **valid** descriptions. A **valid description** is an html fragment between the tags <OBJECT> and </OBJECT> that:

- contains at least one word;
- contains another object;
- contains a link to a valid alternative representation;
- doesn't contain [placeholder text](#).

- **Learn More:**

- the W3C on [Objects, Images, and Applets](#);
- the W3C on [Generic inclusion: the OBJECT element](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

2.38 Audio/video OBJECT with equivalent CONTENT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.1.

The rule checks if the OBJECT elements contained in the page embed audio or video files with an **equivalent** description between <OBJECT> and </OBJECT> tags.

This is a manual test.

- **Issue Description:**

The OBJECT elements with embedded audio or video files included in this document should have corresponding **equivalent** description between <OBJECT> and </OBJECT> tags.

- **How to fix:**

Insert a **meaningful** description inside <OBJECT> section.

In particular, check the content inside the <OBJECT> and </OBJECT> tags:

- if it is a textual description, it should be a transcription of the audio/video file;
- if it is a link, it should point to a transcription;
- if it is an <OBJECT>, it should be an audio file or a transcription.

- **Why to fix:**

Equivalents for sounds can be provided in the form of a text phrase on the page that links to a text transcript or description of the sound file. The link to the transcript should appear in a highly visible location such as at the top of the page. However, if a script is automatically loading a sound, it should also be able to automatically load a visual indication that the sound is currently being played and provide a description or transcript of the sound.

The OBJECT element's design offers a mechanism for specifying alternate object renderings; each embedded OBJECT declaration may specify alternate content types. If a browser cannot render the outermost OBJECT, it tries to render the contents, which may be another OBJECT element, etc.

- **Learn More:**

- the W3C on [Objects, Images, and Applets](#);
- the W3C on [Generic inclusion: the OBJECT element](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

2.39 OBJECT with valid CONTENT [WCAG 01.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.1.

The rule checks if the page contains OBJECT elements without a valid description between <OBJECT> and </OBJECT> tags.

- **Issue Description:**

The page contains OBJECT elements without a **valid** description between <OBJECT> and </OBJECT> tags.

- **Specific Issues:**

- **Content is empty:** Alternative content of object is the empty string ""
- **Content is blank:** Alternative content of object is the blank string " "
- **Content with size:** Alternative content of object describes only the size of the embedded file
- **Content with filename:** Alternative content of object describes only the name of the embedded file
- **Content with placeholder text:** Alternative content of object appears to contain only placeholder text

- **How to fix:**

Insert a **meaningful** description inside <OBJECT> section.

The ALT description should convey the same information as the object embedded and it should explain the role the object plays in the page: why it's there, what does it represent, how does it present the information.

- **Why to fix:**

The OBJECT element's design offers a mechanism for specifying alternate object renderings; each embedded OBJECT declaration may specify alternate content types. If a browser cannot render the outermost OBJECT, it tries to render the contents, which may be another OBJECT element, etc.

Objects included in this document don't have corresponding **valid** descriptions. A **valid** description is an html fragment between the tags <OBJECT> and </OBJECT> that:

- contains at least one word;
- contains another object;
- contains a link to a valid alternative representation;
- doesn't contain [placeholder text](#).

- **Learn More:**

- the W3C on [Objects, Images, and Applets](#);
- the W3C on [Generic inclusion: the OBJECT element](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

2.40 OBJECT with equivalent CONTENT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.1.

The rule checks if the **valid** descriptions of the OBJECT elements contained in the page are an **equivalent** description of the object themselves.

- **Issue Description:**

The **valid** description contained between <OBJECT> and </OBJECT> tags of the OBJECT elements in the page should be an **equivalent** description of the objects themselves.

- **How to fix:**

Insert a **meaningful** description inside <OBJECT> section.

The ALT description should convey the same information as the object embedded and it should explain the role the object plays in the page: why it's there, what does it represent, how does it present the information.

- **Why to fix:**

The OBJECT element's design offers a mechanism for specifying alternate object renderings; each embedded OBJECT declaration may specify alternate content types. If a browser cannot render the outermost OBJECT, it tries to render the contents, which may be another OBJECT element, etc.

- **Learn More:**

- the W3C on [Objects, Images, and Applets](#);
- the W3C on [Generic inclusion: the OBJECT element](#);
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

2.41 Linked AUDIO with equivalent CONTENT [WCAG 01.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(a) "Text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content)." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content" and more specifically checkpoint 1.1.

The rule checks if the page contains A or AREA elements pointing to audio files. If so, the rule prompts the user to check if the audio file is described within the document or if document contains a link to a text equivalent file.

This is a manual test.

- **Issue Description:**

The page contains A or AREA elements pointing to audio files.

Please check if the audio file is described within the document or if document contains a link to a transcript file or a file containing the same information.

- **How to fix:**

Insert a **meaningful** description of the audio file.

In particular, check the page content:

- it should contain a transcription of the audio/video file or
- it should contain a link pointing to a transcription.

- **Why to fix:**

Text equivalents for sounds can be provided as textual links on the page to a text transcript or a description of the sound file. The link to the transcript should appear in a highly visible location such as at the top of the page.

Unless transcripts are provided, users that cannot hear the audio's output would not be able to take advantage of its contents. Consider the following cases:

- deaf or poorly hearing users;
- users accessing a audio device on a noisy environment;
- users that do not have the appropriate plug-in for hearing sounds;
- users that access the site with a computer that has no audio devices;
- users accessing the site in a place where sounds cannot be played (like a public library).

- **Learn More:**

- the HTML 4.0 standard on [A](#) and [AREA](#) elements;
- the W3C Accessibility guidelines on [ALT descriptions](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

3 WCAG 1.0 checkpoint 1.2

3.1 Links are needed for server-side image map [WCAG 01.2]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Imagemaps
- Manual
- ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(e) "Redundant links shall be provided for each active region of a server-side image map." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content.", and more specifically Checkpoint 1.2 "Provide redundant text links for each active region of a server-side image map".

The test checks if the page includes a server-side image map. If so, it prompts the user to verify that each hotspot (i.e. active region of the image map) has a redundant textual link.

- **Issue Description:**

The page includes a server-side image map. Please verify that each hotspot (i.e. active region of the image map) has a redundant textual link.

- **Specific Issues:**

- **IMG element:** An IMG element has been used as a server-side imagemap
- **INPUT element:** An INPUT element has been used as a server-side imagemap

- **How to fix:**

Please check that each active region of the image map is duplicated as a redundant link in the page. You must inspect the implementation of the image map within the server to find out which are the URLs that are activated when the user clicks on any of the active regions.

- **Why to fix:**

Server-side image maps pose big accessibility problems.

- Server-side image maps require specific input devices (like the mouse), which may not be available to certain users in certain contexts (for example, through a cellular phone, or via a voice controlled browser);
- certain users in certain situations could not accurately click on the map (eg. disabled persons, someone walking, driving, standing).

- the links specified by server-side imagemaps are hidden in the server and cannot be manipulated by browsers in any way, including any assistive technology. Therefore the browser cannot show any alternative links to the user;
- server-side image maps are less efficient than their client-side counterparts since an additional interaction with the server is needed for each user click.

For these reasons, server-side image maps are highly discouraged from being used. The only exception is for active regions whose shape is so irregular that client-side image maps cannot support.

• **Learn More:**

- HTML 4.0 standard on [image maps in general](#);
- HTML 4.0 standard on [server side image maps](#);
- HTML 4.0 standard on [possible shapes of client-side image maps](#);
- HTML 4.0 standard on the [ismap attribute](#);
- W3C/WAI on [how to make accessible image maps](#);
- W3C/WAI [server-side image maps](#);
- W3C/WAI [guideline 1](#) and checkpoint 1.2 **Provide redundant text links for each active region of a server-side image map**;
- W3C/WAI suggested techniques for [dealing with server-side image maps](#);
- a short tutorial on [accessible imagemaps](#).

4 WCAG 1.0 checkpoint 1.3

4.1 Multimedia with equivalent audio description [WCAG 01.3]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(b) "Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.3.

The rule checks if the page contains elements embedding multimedia files (OBJECT or EMBED) or elements referring directly to them (A, AREA).

In this case the rule prompts the user to check if auditory descriptions are provided for every visual track in the multimedia presentation.

- **Issue Description:**

The page contains elements embedding multimedia files (OBJECT or EMBED) or elements referring directly to them (A, AREA).

Please check if auditory descriptions are provided for every visual track in the multimedia presentation.

- **Specific Issues:**

- **link pointing to multimedia:** A direct link to a multimedia file is present
- **OBJECT embedding multimedia:** An OBJECT embedding a multimedia file is present
- **EMBED pointing to multimedia:** An EMBED element pointing to a multimedia file is present

- **How to fix:**

The page contains elements embedding multimedia files (OBJECT or EMBED) or elements referring directly to them (A, AREA).

Please check that audio descriptions are provided for all important visual information as:

- alternative tracks of the multimedia file;

- links to specific audio files.

- **Why to fix:**

Generally, multimedia presentations require users to rely on more than one sense, such as combining a reliance on hearing and vision to understand the content of a page. While these advances make web pages more interesting and "layered" for people without disabilities, those with disabilities face new barriers to understanding a page's content.

To ensure that multimedia is accessible to users with visual impairments, agencies should ensure that audio descriptions of video portions are provided in all cases, except where the video portion of a presentation is unimportant to understanding a presentation.

Auditory descriptions of the visual track provide narration of the key visual elements without interfering with the audio or dialogue of a movie. Key visual elements include actions, settings, body language, graphics, and displayed text.

Until the format you are using supports alternative tracks, you could add a link to an existing audio description file.

- **Learn More:**

- Netscape on the [EMBED](#) element;
- the W3C on [Generic inclusion: the OBJECT element](#);
- the HTML 4.0 standard on [A](#) and [AREA](#) elements;
- the W3C Accessibility guidelines on [multimedia equivalent alternatives](#);
- the W3C on [Synchronized Multimedia Integration Language \(SMIL\) 1.0 Specification](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

5 WCAG 1.0 checkpoint 1.4

5.1 Multimedia with synchronized alternative [WCAG 01.4]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Scripts
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(b) "Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation." according to W3C/WAI [Guideline 1](#) "Provide equivalent alternatives to auditory and visual content." and more specifically checkpoint 1.4.

The rule checks if the page contains elements embedding multimedia files (OBJECT or EMBED) or elements referring directly to them (A, AREA).

In this case the rule prompts the user to check if equivalent alternatives (e.g., captions or auditory descriptions of the visual track) are synchronized with the multimedia presentation.

- **Issue Description:**

The page contains elements embedding multimedia files (OBJECT or EMBED) or elements referring directly to them (A, AREA).

Please check if equivalent alternatives (e.g., captions or auditory descriptions of the visual track) are synchronized with the multimedia presentation.

- **Specific Issues:**

- **Link pointing to multimedia:** A direct link to a multimedia file is present
- **OBJECT embedding multimedia:** An OBJECT embedding a multimedia file is present
- **EMBED pointing to multimedia:** An EMBED element pointing to a multimedia file is present

- **How to fix:**

The page contains elements embedding multimedia files (OBJECT or EMBED) or elements referring directly to them (A, AREA).

Please check that:

- for any multimedia content, text captioning is provided for all audible output;

- all audio descriptions and text captions are synchronized with their associated dynamic content.

- **Why to fix:**

Auditory presentations must be accompanied by **text transcripts**, textual equivalents of auditory events. When these transcripts are presented synchronously with a video presentation they are called **captions** and are used by people who cannot hear the audio track of the video material.

Some media formats (e.g., QuickTime 3.0 and SMIL) allow captions and video descriptions to be added to the multimedia clip. SAMI allows captions to be added.

Until the format you are using supports alternative tracks, two versions of the movie could be made available, one with captions and descriptive video, and one without. Some technologies, such as SMIL and SAMI, allow separate audio/visual files to be combined with text files via a synchronization file to create captioned audio and movies.

Some technologies also allow the user to choose from multiple sets of captions to match their reading skills. For more information see the [SMIL 1.0](#) specification.

- **Learn More:**

- Netscape on the [EMBED](#) element;
- W3C on [Generic inclusion: the OBJECT element](#);
- HTML 4.0 standard on [A](#) and [AREA](#) elements;
- W3C Accessibility guidelines on [multimedia equivalent alternatives](#);
- W3C on [Synchronized Multimedia Integration Language \(SMIL\) 1.0 Specification](#);
- the [508 standard](#).
- [checkpoint 4](#) from IBM Web accessibility checklist - version 3.1

6 WCAG 1.0 checkpoint 2.1

6.1 Color is not essential [WCAG 02.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(c) "Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup." according to [WAI checkpoint 2.1](#) "Verify that information conveyed with color is available without color"

The rule checks if the page specifies a color for any possible item in the page or if it embeds an image, object, applet, or script. If so, the rule prompts the user to verify if any information is dependent on color.

- **Issue Description:**

The page contains colors and therefore checks must be made to ensure that the information conveyed within the page does not rely on these colors.

- **How to fix:**

Please check that the page can be understood and navigated even if users do not have the ability to identify specific colors or differentiate between colors.

Easy ways to test the page are:

- View the page on a black and white screen and go through each of its elements;
- Print the page on a black and white printer;
- Scan the page and find sentences like **press the red button**.

- **Why to fix:**

There are many situations where a person cannot distinguish colors. These include:

- Poor choice of background/foreground colors;
- Users of a textual browser;
- Users listening to a reading browser;
- Users with a black and white display, for example a PDA or cell phone;
- Color-blind people;

- Persons with impaired sight, or blindness.

Colors on a page should be used only as decorative, or graphic devices. Not for relaying information.

- **Learn More:**

- HTML 4.0 standard on [color names and color usage](#);
- HTML 4.0 (deprecated) [attribute to specify background color](#);
- [WAI guideline](#) on accessible color usage;
- Specific [techniques](#) suggested by WAI to handle colors in accessible pages.
- [checkpoint 12](#) from IBM Web accessibility checklist - version 3.1
- A set of guidelines for achieving effective [color contrast](#).
- How color blind view the world: [a short tutorial](#)

7 WCAG 1.0 checkpoint 2.2

7.1 Colors are visible [WCAG 02.2]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements [WAI checkpoint 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."

The rule checks if the page specifies a color for any possible item in the page or if it embeds an image, object, applet, or script. If so, the rule prompts the user to verify if colored items can be differentiated.

- **Issue Description:**

The page uses some colors and there might be cases where the contrast between foreground and background colors is not sufficient to differentiate them.

- **How to fix:**

Please check that colors and colored items in the page can be clearly differentiated in all the possible contexts in which users may be using the page.

Make sure that the contrast between foreground and background items is conveyed by other means in addition to colors, like different font styles, font sizes, font faces.

And if you use colors, consider exaggerating the differences between foreground and background colors by making colors differ in all the following three parameters:

- **hue**
- **saturation**
- **lightness**

Easy ways to test the page are:

- view the page on a black and white screen and go through each of its elements;
- print the page on a black and white printer;
- take the printout and copy it two or three times to see how it degrades. This will show you where you need to add redundant cues (like underlying links) or whether the cues are too small or indistinct to hold up well.

- **Why to fix:**

There are many situations where a bad choice of colors hinders perception and comprehension of some information item or image in a page.

The bad choice may depend on many factors including:

- bad choice of background/foreground colors;
- the user uses a screen incapable of rendering colors with the same quality as the one used by the page designer;
- the user is accessing the page via a black and white PDA or cell phone;
- the user needs to print the page on a black and white printer;
- the user is color-blind.

- **Learn More:**

- the HTML 4.0 standard on [color names and color usage](#);
- the HTML 4.0 (deprecated) [attribute to specify background color](#);
- A set of guidelines for achieving effective [color contrast](#).
- How color blind view the world: [a short tutorial](#)
- the [WAI guideline](#) on accessible color usage;
- specific [techniques](#) suggested by WAI to handle colors in accessible pages;
- [checkpoint 12](#) from IBM Web accessibility checklist - version 3.1

8 WCAG 1.0 checkpoint 4.1

8.1 Clarify natural language usage [WCAG 04.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements W3C/WAI [Guideline 4](#) "Clarify natural language usage." and more specifically checkpoint 4.1 "Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions)".

The rule prompts the user to check if the page contains chunks of text written using different natural languages. If so, then every chunk should be contained in a tag with a **lang** attribute.

- **Issue Description:**

Please check if the page contains chunks of text written using different natural languages (i.e. English, French, Spanish, etc.). If so, then every chunk should be contained in a tag with a **lang** attribute.

- **How to fix:**

Clearly identify changes in the natural language of a document's text and any text equivalents:

- identify chunks of text written using different natural languages;
- add a **lang** attribute to the innermost element containing the text for each chunk identified previously.

- **Why to fix:**

If you use a number of different languages on a page, make sure that any changes in language are clearly identified by using the **lang** attribute.

Identifying changes in language is important for a number of reasons:

- users who are reading the document in braille will be able to substitute the appropriate control codes (markup) where language changes occur to ensure that the braille translation software will generate the correct characters (accented characters, for instance);

- similarly, speech synthesizers that **speak** multiple languages will be able to generate the text in the appropriate accent with proper pronunciation. If changes are not marked, the synthesizer will pronounce the words in the language it works in, usually producing sounds that cannot be understood;
- users who are unable to translate between languages themselves, will be able to have unfamiliar languages translated by machine translators.

Consider that more and more sites will be accessed through audio only devices (for example by telephone).

The **lang** attribute may be helpful in many other situations:

- assisting search engines;
- helping a browser select glyph variants for high quality typography;
- helping a browser choose a set of quotation marks;
- helping a browser make decisions about hyphenation, ligatures, and spacing;
- assisting spell checkers and grammar checkers.

- **Learn More:**

- HTML 4.0 standard on the [lang](#) attribute;
- W3C Accessibility [Guideline 4](#);
- W3C HTML techniques on [identifying changes in language](#).

9 WCAG 1.0 checkpoint 5.1

9.1 Data table should have headers [WCAG 05.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Tables
- ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(g) "Row and column headers shall be identified for data tables" according to W3C/WAI [Guideline 5](#) "Create tables that transform gracefully" and more specifically checkpoint 5.1 "For data tables, identify row and column headers".

The test checks if the page includes a table **not used for layout** that does not have specific cells marked as headers (i.e. **TH** elements).

- **Issue Description:**

Table does not have cells in first row or first column marked as headers (i.e. using **TH** tags instead of TDs).

- **Specific Issues:**

- **no TH is defined:** No header cell (TH) has been defined for the table
- **Only some TH is defined:** Only some header cell (TH) has been defined for the table
- **Cell is not a header:** Cell is marked as data cell (TD) but it should be a header cell (TH)

- **How to fix:**

Make sure that each row and column has cells that are appropriately marked up as being headers. Use the TH tag instead of TD.

If you don't like the way in which a graphical browser displays TH elements, you can define an appropriate CSS rule to display them the way you like. In this way, the table is marked up using **structural HTML** but that does not affect the way in which it is graphically presented.

Consider also that:

- **TH** elements should set the **ID** attribute for identification;
- **TD** elements should set the **HEADERS** attribute to refer to the appropriate **TH** elements.

Alternatively the **TH** element may specify a **SCOPE='col'** or **SCOPE='row'** attribute meaning that the header refers to the entire column or row, respectively.

It might be a good idea to use the attribute **ABBR** for TH in order to provide a more concise description of the header that can be repeated over and over by reading browsers.

- **Why to fix:**

There are two flavors of tables that can be used in an HTML page:

Data tables are used to present data such as a bus schedule, a comparison of regional sales figures, or a listing of employee contact information. Cells in data tables are related to each other and usually must be perceived as a group.

Layout tables on the other hand are used to arrange images and text on a page according to set grid. Each cell in a layout table is normally independent and can be viewed on its own.

The table considered by this test appears to be a Data Table.

Data tables are used to convey information in a bidirectional medium, which might not be always available. Consider for example:

- a reading browser, that has to read sequentially the content of the table;
- a braille reader, that also scans the table sequentially;
- a browser with a very small display (like a PDA or a cellular phone) that shows only a limited portion of the table.
- a textual browser that may not align correctly rows and columns (for example because the content of a cell wraps);

In all these cases the user has to remember the context of each cell (which row and which column does it refer to?). This requirement may be too strong when the table is large, when the user is under stress, or is impatient to find the needed information.

Consider also that in these situations users might not be able to move directly from one cell of the table to any other. They are constrained to move sequentially, from one cell to its neighbors only (for example within the same row).

- **Learn More:**

- HTML 4.0 standard on [tables in general](#);
- HTML 4.0 standard on the [TH and TD elements](#);
- HTML 4.0 standard on [table rendering](#) by various types of browsers;
- W3C/WAI on [how to make accessible data tables](#);
- [checkpoint 10](#) from IBM Web accessibility checklist - version 3.1
- a [detailed discussion](#) on linearizing layout tables;

- you can also check how tables are linearized by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

9.2 Data table with valid THs [WCAG 05.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Tables
- ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(g) "Row and column headers shall be identified for data tables" according to W3C/WAI [Guideline 5](#) "Create tables that transform gracefully" and more specifically checkpoint 5.1 "For data tables, identify row and column headers".

The test checks if the page includes a table **not used for layout** containing a TH cell that does not specify an ID or SCOPE attribute. (Such a TH cannot be referred to.)

- **Issue Description:**

Table cell is a TH element that does not have the attribute SCOPE nor a valid ID attribute. Therefore the TH cell cannot be referred to by other cells.

- **Specific Issues:**

- **All TH cells are not valid:** All TH cells in the table are considered not valid because no data cells refer to them or ID and SCOPE attributes are missing or are not valid
- **Scope used inside a complex table:** TH cell has a SCOPE attribute defined but it is inside a complex table. Some screen reader might not be able to process it correctly
- **Header cell not referred by any cell:** TH cell has a valid ID attribute but there are not cells pointing to it
- **Undefined ID:** TH cell has no valid ID attribute defined
- **Undefined SCOPE and ID:** TH cell has no valid SCOPE or ID attributes defined
- **Duplicated or not valid ID:** TH cell has a duplicated or a not valid ID

- **How to fix:**

Make sure that each row and column has cells that are appropriately marked up so that they can be referred to.

- **TH** elements should set the **ID** attribute for identification;

- **TD** elements should set the **HEADERS** attribute to refer to the appropriate **TH** elements.

Alternatively the **TH** element may specify a **SCOPE='col'** or **SCOPE='row'** attribute meaning that the header refers to the entire column or row, respectively.

If a table contains more than one row or column of headers, or if there are cells spanning two or more columns or rows, the table is considered to be **complex**.

Some screen readers and speaking machines may fail to bind data cells to headers cells when **SCOPE** attribute is used with these tables. For complex tables we suggest to use **ID** and **HEADERS** instead.

It might be a good idea to use the attribute **ABBR** for **TH** in order to provide a more concise description of the header that can be repeated over and over by specialized browsers.

- **Why to fix:**

There are two flavors of tables that can be used in an HTML page:

Data tables are used to present data such as a bus schedule, a comparison of regional sales figures, or a listing of employee contact information. Cells in data tables are related to each other and usually must be perceived as a group.

Layout tables on the other hand are used to arrange images and text on a page according to set grid. Each cell in a layout table is normally independent and can be viewed on its own.

The table considered by this test appears to be a Data Table.

Data tables are used to convey information in a bidirectional medium, which might not be always available. Consider for example:

- a reading browser, that has to read sequentially the content of the table;
- a braille reader, that also scans the table sequentially;
- a browser with a very small display (like a PDA or a cellular phone) that shows only a limited portion of the table.
- a textual browser that may not align correctly rows and columns (for example because the content of a cell wraps);

In all these cases the user has to remember the context of the cell (which row and which column does it refer to?). This requirement may be too strong when the table is large, when the user is under stress, or is impatient to find the needed information.

Consider also that in these situations users might not be able to move directly from one cell of the table to any other. They might be constrained to move sequentially, from one cell to its neighbors only (for example within the same row).

- **Learn More:**

- HTML 4.0 standard on [tables in general](#);
- HTML 4.0 standard on the [TH and TD elements](#);
- HTML 4.0 standard on [table rendering](#) by various types of browsers;
- W3C/WAI on [how to make accessible data tables](#);
- [checkpoint 10](#) from IBM Web accessibility checklist - version 3.1
- a [detailed discussion](#) on linearizing layout tables;
- you can also check how tables are linearized by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

9.3 Cell of data table should refer to headers [WCAG 05.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Tables
- ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(g) "Row and column headers shall be identified for data tables" according to W3C/WAI [Guideline 5](#) "Create tables that transform gracefully" and more specifically [checkpoint 5.1](#) "For data tables, identify row and column headers".

The test checks if the page includes a table with TD elements that do not refer to any row's or column's THs.

- **Issue Description:**

The page includes a table that has TD elements that do not refer to the row's or column's headers (i.e. TH).

This happens because either the TD element has no HEADERS attribute defined, or because the row/column header TH has no proper SCOPE attribute defined.

- **Specific Issues:**

- **All TDs without valid reference to THs:** All data cells (TDs) miss a reference to row/column headers (THs) or referred headers are empty
- **Reference to empty row's TH:** TD cell refers only to empty row headers (TH). This reference is useless to assistive technology
- **Reference to empty column's TH:** TD cell refers only to empty column headers (TH). This reference is useless to assistive technology

- **Missing reference to row's TH:** TD cell does not refer to its row header (TH)
- **Missing reference to column's TH:** TD cell does not refer to its column header (TH)
- **Missing reference to row's and column's TH:** TD cell does not refer to its row/column headers (TH)

- **How to fix:**

Make sure that the table cell refers to row and column headers.

- **TH** elements should set the **ID** attribute for identification;
- **TD** elements should set the **HEADERS** attribute to refer to the appropriate **TH** elements.

Alternatively the **TH** element may specify a **SCOPE='col'** or **SCOPE='row'** attribute meaning that the header refers to the entire column or row, respectively.

- **Why to fix:**

There are two flavors of tables that can be used in an HTML page:

Data tables are used to present data such as a bus schedule, a comparison of regional sales figures, or a listing of employee contact information. Cells in data tables are related to each other and usually must be perceived as a group.

Layout tables on the other hand are used to arrange images and text on a page according to set grid. Each cell in a layout table is normally independent and can be viewed on its own.

Data tables are used to convey information in a bidirectional medium, which often is not available. Consider for example:

- a reading browser, that has to sequentially read the content of the table;
- a braille reader, that also sequentially scans the table;
- a browser with a very small display (like a PDA or a cellular phone) that shows only a limited portion of the table.
- a textual browser that may not align correctly rows and columns (for example because the content of a cell wraps);

In all these cases the user has to remember the context of the cell (which row and which column does it refer to?). This requirement may be too strong, for example, when the table is large, when the user is under stress, or is impatient to find the needed information.

Consider also that in these situations users might not be able to move directly from one cell of the table to any other. They might be constrained to move sequentially, from one cell to its neighbors only (for example within the same row).

- **Learn More:**

- HTML 4.0 standard on [tables in general](#);
- HTML 4.0 standard on the [TH and TD elements](#);
- HTML 4.0 standard on [table rendering](#) by various types of browsers;
- W3C/WAI on [how to make accessible data tables](#);
- [checkpoint 10](#) from IBM Web accessibility checklist - version 3.1
- you can also check how tables are linearized by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

9.4 Data tables should be defined by TABLE tag [WCAG 05.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- Tables
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(g) "Row and column headers shall be identified for data tables" according to W3C/WAI [Guideline 5](#) "Create tables that transform gracefully" and more specifically [checkpoint 5.1](#) "For data tables, identify row and column headers".

The rule checks if the page includes a PRE element with pre-formatted tabular data. The rule suggests the user to check if the PRE tag is used to present data and in such a case to convert it to a TABLE element.

- **Issue Description:**

The page includes a PRE element with pre-formatted tabular data. Please check if the PRE tag is used to present data and in such a case convert it to a TABLE element.

- **How to fix:**

Please verify if the PRE tag is used to arrange and present data in a tabular format. If not, skip this problem.

Otherwise, you should use the TABLE, TR, TH, TD tags with appropriate attributes (ID, HEADERS, SCOPE) to appropriately markup the tabular structure of the data.

In addition:

- **TH** elements should set the **ID** attribute for identification;

- **TD** elements should set the **HEADERS** attribute to refer to the appropriate **TH** elements.

Alternatively the **TH** element may specify a **SCOPE='col'** or **SCOPE='row'** attribute meaning that the header refers to the entire column or row, respectively.

It might be a good idea to use the attribute **ABBR** for **TH** in order to provide a more concise description of the header that can be repeated over and over by specialized browsers.

- **Why to fix:**

Data tables are used to convey information in a bidirectional medium, which often is not available. Consider for example:

- a textual browser that may not align correctly rows and columns (for example because the content of a cell wraps);
- a reading browser, that has to sequentially read the content of the table;
- a braille reader, that also sequentially scans the table;
- a browser with a very small display (like a PDA or a cellular phone) that shows only a limited portion of the table.

In all these cases the user has to remember the context of the cell (which row and which column does it refer to?). This requirement may be too strong, for example, when the table is large, when the user is under stress, or is impatient to find the needed information.

Consider also that in these situations users might not be able to move directly from one cell of the table to any other. They are constrained to move sequentially, from one cell to its neighbors only (for example within the same row).

- **Learn More:**

- the HTML 4.0 standard on [tables in general](#);
- the HTML 4.0 standard on the [TH and TD elements](#);
- the HTML 4.0 standard on [table rendering](#) by various types of browsers;
- the W3C/WAI on [how to make accessible data tables](#);
- you can also check how tables are linearized by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).

10 WCAG 1.0 checkpoint 5.2

10.1 Multiple headers should be marked in data tables [WCAG 05.2]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Manual
 - Tables
 - ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(h) "Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers" according to W3C/WAI [Guideline 5](#) "Create tables that transform gracefully" and more specifically checkpoint 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".

The test checks if the page includes a TABLE element with a TH element and does not have any THEAD, TFOOT, TBODY elements.

The test prompts the user to check if the table includes multiple levels of row and column headers. If so the test suggests the user to markup such levels.

- **Issue Description:**

The page includes a TABLE element with cells and/or headers spanning more than one row or column.

You need to inspect it and check if its headers are appropriately marked up and are referred to by internal cells.

- **How to fix:**

Please verify if the data is organized in different sections requiring different headers.

If not, skip this problem.

Otherwise, you should use the THEAD, TFOOT, TBODY tags (perhaps with COLGROUP and COL) to markup these different sections.

Make sure that the table headers are appropriately marked up with TH and that table cells refers to row and column headers.

- **TH** elements should set the **ID** attribute for identification;
- **TD** elements should set the **HEADERS** attribute to refer to the appropriate **TH** elements.

Alternatively the **TH** element may specify a **SCOPE='col'**, **SCOPE='row'**, **SCOPE='colgroup'** or **SCOPE='rowgroup'** attribute meaning that the header refers to the entire column or row, respectively.

- **Why to fix:**

Data tables are used to convey information in a bidirectional medium, which often is not available. Consider for example:

- a reading browser, that has to sequentially read the content of the table;
- a braille reader, that also sequentially scans the table;
- a browser with a very small display (like a PDA or a cellular phone) that shows only a limited portion of the table.
- a textual browser that may not align correctly rows and columns (for example because the content of a cell wraps);

In all these cases the user has to remember the context of the cell (which row and which column does it refer to?). This requirement may be too strong, for example, when the table is large, when the user is under stress, or is impatient to find the needed information.

Consider also that in these situations users might not be able to move directly from one cell of the table to any other. They might be constrained to move sequentially, from one cell to its neighbors only (for example within the same row).

If the table is structured in different sections, each with its own row or column headers, or if cells span more than one row or one column, then this complex structure should be clearly marked up so that specialized browsers can take advantage of it and articulate it (perhaps by repeating the appropriate headings when moving on a cell).

- **Learn More:**

- HTML 4.0 standard on [tables in general](#);
- HTML 4.0 standard on the [TH and TD elements](#);
- HTML 4.0 standard on the [THEAD, TFOOT, TBODY elements](#);
- HTML 4.0 standard on [table rendering](#) by various types of browsers;
- W3C/WAI on [how to make accessible data tables](#);
- you can also check how tables are linearized by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).
- [checkpoint 10](#) from IBM Web accessibility checklist - version 3.1

11 WCAG 1.0 checkpoint 6.1

11.1 Style sheets should not be necessary [WCAG 06.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(d) "Documents shall be organized so they are readable without requiring an associated style sheet." according to [WAI checkpoint 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."

The rule checks if the page requires the use of Cascade Style Sheets (CSS). If so, the rule prompts the user to verify if the page can be understood and navigated without applying style specifications.

- **Issue Description:**

The page requires the use of Cascade Style Sheets (CSS). Please verify if the page can be understood and navigated without applying style specifications.

- **How to fix:**

Please check that this document can be read without style sheets.

- **Why to fix:**

Cascade Style Sheets (CSS) is a great technology for separating content and its structure from presentation. Usually, when content is organized logically, it will be rendered in a meaningful order when style sheets are turned off or not supported. However there may be situations where CSS gets in the way, limiting accessibility to a web page.

- old browsers do not support CSS, and thus page authors should not rely on styles to convey information or enable navigation;
- user-defined style information (like font size, font color) should be unaffected by styles specified in the page.

- **Learn More:**

- the HTML 4.0 standard on [adding styles to documents](#);
- the HTML 4.0 standard on [the STYLE element](#);

- the HTML 4.0 standard on [the style attribute](#);
- what the W3C says about [browsers default styles](#);
- the W3C/WAI [Guideline 6](#) and Checkpoint 6.1;
- [checkpoint 11](#) from IBM Web accessibility checklist - version 3.1
- W3C/WAI suggested techniques for [appropriate use of content with CSS styles](#);
- W3C/WAI suggested techniques for [specifying horizontal rules](#).

12 WCAG 1.0 checkpoint 6.2

12.1 Text only equivalent page may be needed [WCAG 06.2]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Scripts
 - Manual
 - ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(k) "A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes" which is related to [W3C/WAI Guideline 6](#) "Ensure that pages featuring new technologies transform gracefully" and more specifically Checkpoint 6.2 "Ensure that equivalents for dynamic content are updated when the dynamic content changes".

The test checks if the page contains SCRIPT, OBJECT or APPLET tags. If so it prompts the user to verify if the page satisfies all the 508 accessibility conditions. If not, the user is asked to check if there is an accessible version of the same page and if it is updated when the current page changes.

- **Issue Description:**

The page contains programmatic objects that can make it not accessible. If it cannot be made accessible, there should be a text-only version of the page that is updated whenever this page changes.

- **How to fix:**

Please verify if this page satisfies all the 508 accessibility conditions. If some is not satisfied, and the page cannot be changed to satisfy them, then an **equivalent text-only** version of this page is needed.

Check if such a page is present and verify that it is updated whenever the current page is changed.

- **Why to fix:**

If a page cannot be made accessible, then an accessible **equivalent** page (a text-only version of it would suffice) is needed. Text-only alternatives to inaccessible pages allow people with disabilities to have full access to all information available to others. Such pages must contain equivalent information or functionality as the primary pages.

It is also important that these redundant copies of pages are updated as often as the primary pages from which they are derived, otherwise they would not be equivalent any more.

You should only resort to alternative pages when other solutions fail because alternative pages are generally updated less often than "primary" pages. An out-of-date page may be as frustrating as one that is inaccessible since, in both cases, the information presented on the original page is unavailable. Automatically generating alternative pages may lead to more frequent updates, but content developers must still be careful to ensure that generated pages always make sense, and that users are able to navigate a site by following links on primary pages, alternative pages, or both. Before resorting to an alternative page, reconsider the design of the original page; making it accessible is likely to improve it for all users.

- **Learn More:**

- HTML 4.0 standard on [objects and applets](#);
- HTML 4.0 standard on the [OBJECT tag](#);
- HTML 4.0 standard on the [APPLET tag](#);
- HTML 4.0 standard on [scripts](#) ;
- HTML 4.0 standard on the [SCRIPT tag](#);
- W3C/WAI [guideline 6](#);
- W3C/WAI [checkpoint 6.2](#).
- [checkpoint 15](#) from IBM Web accessibility checklist - version 3.1

13 WCAG 1.0 checkpoints 6.3, 8.1, 9.2, 9.3

13.1 Scripts are accessible [WCAG 06.3]

- **Test Type:** Manual
- **Test Categories:**
 - W3C WCAG Accessibility
 - Scripts
 - Manual
 - ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(1) "When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology" which is related to WAI/ WCAG 1.0

[Checkpoint 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. ", [Checkpoint 8.1](#) "Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies", [Checkpoint 9.2](#) "Ensure that any element that has its own interface can be operated in a device-independent manner" and [Checkpoint 9.3](#) "For scripts, specify logical event handlers rather than device-dependent event handlers."

The rule checks if the page contains SCRIPT, OBJECT, EMBED or APPLET tags. If so it prompts the user to verify if the programmatic object produces information (content or navigation options) that is otherwise not available.

- **Issue Description:**

The page contains a programmatic object (a script, a plug-in or an applet) that may be used to change the content of the page and/or its navigation options and perhaps to reduce accessibility of the page.

- **Specific Issues:**

- **SCRIPT is used:** A script is used in page body: check its accessibility
- **APPLET is used:** An applet is used: check its accessibility
- **OBJECT is used:** A plug-in is used: check its accessibility
- **EMBED is used:** A plug-in is used: check its accessibility
- **Javascript event hadler is used:** A javascript event handler is used: check its accessibility

- **How to fix:**

Not accessible programmatic objects may include:

- rollovers;
- scripts that present menu options;
- scripts that handle events triggered by devices other than keyboard;
- plug-ins (Flash, Shockwave, RealAudio, RealVideo);
- Java applets.

Examine the programmatic object included in the page and verify if it provides information or interaction options that are not available through other text in the page.

If the script cannot be made accessible one viable solution is to include a NOSCRIPT tag with alternative and equivalent content and interaction (via a form).

Consider also transforming the client-side script (executed by the user browser) into an equivalent server-side script. In this case you are required to write it so that it produces accessible pages.

- **Why to fix:**

Programmatic objects may change the page presentation (like color of a button), its content (like providing new specific information) and its navigation options (like presenting new links to follow). If some of these changes is not associated to text that can be read by assistive technology, then the page is not accessible.

For example, a script showing a menu of choices in a page that does not present those choices as textual links, would make the page not accessible.

A related concept is **device independence**. As [defined by W3C/WAI](#) it means that:

- users must be able to interact with a website using the supported input and output devices of their choice and according to their needs. Input devices may include pointing devices, keyboards, braille devices, head wands, microphones, and others. Output devices may include monitors, speech synthesizers, and braille devices.
- Please note that "device-independent support" does not mean that the browser must support every input or output device. It should offer redundant input and output mechanisms for those devices that are supported. For example, if a browser supports keyboard and mouse input, users should be able to interact with all features using either the keyboard or the mouse.

Device-independent access means that the user may interact with the user agent or document with a preferred input (or output) device – mouse, keyboard, voice, head wand, or other. If, for example, a form control can only be activated with a mouse or other pointing device, someone who is using the page without sight, with voice input, or with a keyboard or who is using some other non-pointing input device will not be able to use the form.

The script mentioned above is an example of **device dependence**, since its use would be possible only through a mouse.

Generally, pages that allow keyboard interaction are also accessible through speech input or a command line interface.

- **Learn More:**

- HTML 4.0 standard on [objects and applets](#);
- HTML 4.0 standard on the [OBJECT tag](#);
- HTML 4.0 standard on the [APPLET tag](#);
- HTML 4.0 standard on [scripts](#);
- HTML 4.0 standard on the [SCRIPT tag](#);
- WAI/ WCAG [guideline 8](#);
- [checkpoint 5](#) from IBM Web accessibility checklist - version 3.1
- Netscape on the [EMBED element](#);
- WAI/ WCAG [guideline 9](#).

14 WCAG 1.0 checkpoint 6.5

14.1 No javascript links are used [WCAG 06.5]

- **Test Type:** Automatic
- **Test Categories:**
 - W3C WCAG Accessibility
 - Scripts
 - ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(1) "When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology" and the W3C/WAI [Guideline 6](#) "Ensure that pages featuring new technologies transform gracefully"; more specifically [Checkpoint 6.5](#) "Ensure that dynamic content is accessible or provide an alternative presentation or page".

The rule checks if the page contains A links that specify "javascript:" as a protocol. If so, it suggests to use other means to activate scripts.

- **Issue Description:**

The page contains a link that can be followed only by browsers that support javascript.

- **How to fix:**

You should replace the link that directly starts the script with other ways to start it, like defining a button, separately defining the script (remember also the NOSCRIPT element) and binding an event (like onKeyPress) on the button to the script.

- **Why to fix:**

Links that activate scripts can only be followed by browsers capable of executing javascript. Not all browsers can do that. Consider for example textual browsers like [lynx](#), or browsers coupled with screen readers, or browsers for PDAs or cellular phones.

A user with such browsers would not be able to navigate the page. Even if other ways (links or buttons) to reach that destination are available, the user would face such a link, thinking that it would work as any other does. But the browser failure to follow it will increase user frustration and confusion.

- **Learn More:**

- the HTML 4.0 standard on [links](#) ;
- the W3C/WAI [guideline 6](#).
- [checkpoint 5](#) from IBM Web accessibility checklist - version 3.1

15 WCAG 1.0 checkpoint 7.1

15.1 GIFs do not cause the screen to flicker [WCAG 07.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(j) "Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz" according to W3C/WAI [Guideline 7](#) "Ensure user control of time-sensitive content changes" and more specifically checkpoint 7.1 "Until user agents allow users to control flickering, avoid causing the screen to flicker".

The rule checks if the page contains **GIF animations** that may cause the screen to flicker. It checks GIFs contained in IMG, INPUT and BODY elements. It skips images that are spacers, thumbnails or hidden links.

This is manual test.

- **Issue Description:**

The page contains at least **GIF image** that may cause the screen to flicker. Please check that images do not contain an animation with a refresh rate in the 2 to 55 flashes per second (Hertz) range.

- **How to fix:**

Please ensure that no image causes the display to flicker at a rate in the 2 to 55 flashes per second (Hertz) range. In other words, no image should repeatedly flash or change from dark to light between 2 and 55 times per second.

If so, you should increase the delay time among frames composing the animation or change its colors.

- **Why to fix:**

A flickering or flashing screen may cause seizures in users with photosensitive epilepsy. Content developers should thus avoid causing the screen to flicker.

Seizures can be triggered by flickering or flashing in the 2 to 55 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

Moreover consider that too much animation GIFs may distract some users from reading the page.

- **Learn More:**

- the W3C/WAI [guideline 7](#);
- the W3C/WAI [techniques for guideline 7](#);
- the [508 standard](#).
- Information on epilepsy at www.epilepsy.org.uk
- an authoritative book on the subject is: Harding, G. F. A. & Jeavons, P. M. Photosensitive Epilepsy. MacKeith Press, London, 1994 (available from Cambridge University Press).
- [checkpoint 13](#) from IBM Web accessibility checklist - version 3.1

15.2 Avoid causing the screen to flicker [WCAG 07.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(j) "Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz" according to W3C/WAI [Guideline 7](#) "Ensure user control of time-sensitive content changes" and more specifically checkpoint 7.1 "Until user agents allow users to control flickering, avoid causing the screen to flicker".

The rule checks if the page contains elements that may cause the screen to flicker. It checks SCRIPT, OBJECT, EMBED, APPLET elements.

This is manual test.

- **Issue Description:**

The page contains a graphic object that may cause the screen to flicker.

Please check that the object is not an animation with a refresh rate in the 2 to 55 flashes per second (Hertz) range.

- **Specific Issues:**

- **APPLET is used:** APPLET is used: is it causing the screen to flicker? If so, please modify or remove the applet.
- **OBJECT is used:** OBJECT is used: is it causing the screen to flicker? If so, please modify or remove the applet.

- **EMBED is used:** EMBED is used: is it causing the screen to flicker? If so, please modify or remove the applet.
- **SCRIPT is used:** SCRIPT is used: is it causing the screen to flicker? If so, please modify or remove the applet.

- **How to fix:**

Please ensure that no image causes the display to flicker at a rate in the 2 to 55 flashes per second (Hertz) range. In other words, no image should repeatedly flash or change from dark to light between 2 and 55 times per second.

If so, you should increase the delay time among frames composing the animation or change its colors.

- **Why to fix:**

A flickering or flashing screen may cause seizures in users with photosensitive epilepsy. Content developers should thus avoid causing the screen to flicker.

Seizures can be triggered by flickering or flashing in the 2 to 55 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

These are HTML elements that may cause flickering:

- **SCRIPT** : script language can be used to create animations;
- **OBJECT** : an OBJECT can contain movies, animated GIFs or java applets used to create animations;
- **EMBED** : usually used to include movies;
- **APPLET** : java applets can be used to create animations.

- **Learn More:**

- the W3C on [how to include objects, images and applets](#);
- the W3C on how to use [scripts](#);
- Netscape on the [EMBED](#) element;
- the W3C/WAI [guideline 7](#);
- the W3C/WAI [techniques for guideline 7](#);
- the [508 standard](#).
- Information on epilepsy at www.epilepsy.org.uk
- an authoritative book on the subject is: Harding, G. F. A. & Jeavons, P. M. Photosensitive Epilepsy. MacKeith Press, London, 1994 (available from Cambridge University Press).
- [checkpoint 13](#) from IBM Web accessibility checklist - version 3.1

16 WCAG 1.0 checkpoint 7.4

16.1 No auto refresh is used [WCAG 07.4]

- **Test Type:** Automatic
- **Test Categories:**
 - W3C WCAG Accessibility
 - ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(p) "When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required" and the W3C/WAI [Guideline 7](#) "Ensure user control of time-sensitive content changes" and more specifically Checkpoint 7.4 "Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages."

The rule checks if the page contains a META element with attributes HTTP-EQUIV="refresh" content=NUMBER. If so it suggests to the user to remove that feature.

- **Issue Description:**

The page is automatically updated after a given time. You should remove this behavior.

- **How to fix:**

Until browsers will allow users to turn off these automatically updating features, you should not use auto-refresh.

Try to achieve a similar effect by specifying caching properties and configure appropriately the webserver. This would not affect accessibility of the pages because the page would change only as an effect of the user posing a new request to the server.

- **Why to fix:**

Pages that are automatically updated may pose significant problems to people that are disabled or that use technology that hinders normal interaction patters.

For example, screen readers might not be working properly when the page updates. Or people with physical disabilities might not be able to move quickly or accurately through the page content and navigation items. Also abled people may face problems if, for some reasons, they are slow in reading the page, if they use a very slow internet connection, if they use a small screen that forces them to slow down the reading pace.

Until browsers will allow users to turn off these automatically updating features, you should not use auto-refresh.

- **Learn More:**

- HTML 4.0 standard on the [META tag](#) ;

- HTML 4.0 standard on the [HTTP-EQUIV attribute](#) ;
- W3C/WAI [Guideline 7](#): "Ensure user control of time-sensitive content changes".
- [checkpoint 14](#) from IBM Web accessibility checklist - version 3.1

17 WCAG 1.0 checkpoint 7.5

17.1 No auto redirect is used [WCAG 07.5]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(p) "When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required"

and the W3C/WAI [Guideline 7](#) "Ensure user control of time-sensitive content changes" and more specifically Checkpoint 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."

The rule checks if the page contains a META element with attributes HTTP-EQUIV="refresh" content="NUMBER; URL". If so it suggests to the user to remove that feature or at least to add links and content to the first and second page as well.

- **Issue Description:**

The page is automatically updated after a given time and a new page is automatically loaded after a given time.

You should remove this behavior.

- **How to fix:**

Until browsers will allow users to turn off these automatically updating features, you should not use auto-refresh or auto-redirect.

If auto redirect is needed, then implement it using the redirection capabilities offered by the server. This would not affect accessibility of the pages because the page would change only as an effect of the user posing a new request to the server.

If this is not possible, then you should provide means for the user to access and navigate the items that are present in the current page also in the new page that is loaded automatically. In such a way the user would still be able to access those items when the two pages switch.

- **Why to fix:**

Pages that are automatically updated may pose significant problems to people that are disabled or that use technology that hinders normal interaction patterns.

For example, screen readers might not be working properly when the page updates. Or people with physical disabilities might not be able to move quickly or accurately through the page content and navigation items. Also abled people may face problems if, for some

reasons, they are slow in reading the page, if they use a very slow internet connection, if they use a small screen that forces them to slow down the reading pace.

Until browsers will allow users to turn off these automatically updating features, you should not use auto-refresh.

- **Learn More:**

- the HTML 4.0 standard on the [META tag](#) ;
- the HTML 4.0 standard on the [HTTP-EQUIV attribute](#) ;
- the W3C/WAI [Guideline 7](#): "Ensure user control of time-sensitive content changes".
- [checkpoint 14](#) from IBM Web accessibility checklist - version 3.1

18 WCAG 1.0 checkpoint 9.1

18.1 No server-side image maps should be used [WCAG 09.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- Imagemaps
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(f) "Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape." according to W3C/WAI [Guideline 9](#) "Use features that enable activation of page elements via a variety of input devices" and more specifically checkpoint 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".

The rule checks if the page includes a server-side image map. If so, the rule prompts the user to verify if that image map can be replaced by a more accessible client-side image map.

- **Issue Description:**

The page includes a server-side image map. Please verify if that image map can be replaced by a more accessible client-side image map.

- **Specific Issues:**

- **IMG element:** An IMG element has been used as a server-side imagemap
- **INPUT element:** An INPUT element has been used as a server-side imagemap

- **How to fix:**

Please verify if the server side image map is **really** needed (i.e. if it defines active regions that cannot be specified via the available shapes). You should at least provide alternative ways (e.g. based on textual links) for reaching the same destination pages.

You must inspect the implementation of the image map within the server to find out how active regions are defined.

- **Why to fix:**

Server-side image maps pose big accessibility problems.

- Server-side image maps require specific input devices (like the mouse), which may not be available to certain users in certain contexts (for example, through a cellular phone, or via a voice controlled browser);
- certain users in certain situations could not accurately click on the map (eg. disabled persons, someone walking, driving, standing);
- the links specified by server-side imagemaps are hidden in the server and cannot be manipulated by browsers in any way, including any assistive technology. Therefore the browser cannot show any alternative links to the user;
- server-side image maps are less efficient than their client-side counterparts since an additional interaction with the server is needed for each user click.

For these reasons, server-side image maps are highly discouraged from being used. The only exception is for active regions whose shape is so irregular that client-side image maps cannot support.

You can recognize a server-side imagemap with the browser by pointing the mouse at the image: mouse coordinates are displayed by the browser as you move the mouse over the image.

- **Learn More:**

- the HTML 4.0 standard on [image maps in general](#);
- the HTML 4.0 standard on [server side image maps](#);
- the HTML 4.0 standard on [possible shapes of client-side image maps](#);
- the HTML 4.0 standard on the [ismap attribute](#);
- the W3C/WAI on [how to make accessible image maps](#);
- the W3C/WAI [guideline 1](#) and checkpoint 1.2 **Provide redundant text links for each active region of a server-side image map**;
- W3C/WAI suggested techniques for [Technique 9.1.1](#) "Check for use of server-side image maps";
- a short tutorial on [accessible imagemaps](#).

19 WCAG 1.0 checkpoint 9.2

19.1 Jump Menu is Device Independent [WCAG 09.2]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Forms
- ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(n) "When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues".

It also implements W3C WCAG checkpoint 9.2 "Ensure that any element that has its own interface can be operated in a device-independent manner."

The test checks if there is a **SELECT** element with an **ONCHANGE** event handler that loads another page.

- **Issue Description:**

The page contains a **jump menu** (i.e. a menu with a list of options each leading to a different page) based on a **SELECT** element with an **ONCHANGE** event handler that loads another page. This behavior prevents visitors using keyboards to scroll through the list to select an option.

- **Specific Issues:**

- **MM.jumpMenu() function is used:** MM.jumpMenu() is used to load pages in actual or parent window
- **window.location property is used:** The ONCHANGE event handler associated with SELECT element contains a javascript instruction to load pages in actual or parent window

- **How to fix:**

There are two steps to make device independent a jump menu :

1. remove the **ONCHANGE** attribute from the **SELECT** element;
2. add a button (**INPUT** of type **BUTTON**) after the menu;
3. assign to the button an attribute **ONCLICK** with the content of the **ONCHANGE** attribute previously removed. Put the same content inside the **ONKEYPRESS** attribute.

When adding a Dreamweaver **jump menu** object to the document, it is possible to automatically insert a button after the menu. Simply enable the checkbox '**Insert Go Button After Menu**'. Remember to remove the ONCHANGE attribute of SELECT and to add ONKEYPRESS to the button.

For example, the following code is **wrong**:

```
<form name="select_country">
Select a country:
  <select name="country" onChange="MM_jumpMenu('parent',this,0)">
    <option value="http://www.this_site.com/be" selected>Belgium</option>
    <option value="http://www.this_site.com/us">United States</option>
  </select>
</form>
```

An accessible version might be:

```
<form name="select_country" action="http://www.this_site.com/jump.cgi">
Select a country:
  <select name="country">
    <option value="http://www.this_site.com/be" selected>Belgium</option>
    <option value="http://www.this_site.com/us">United States</option>
  </select>
  <input type="submit" value="Go">
</form>
```

- **Why to fix:**

The javascript code associated with the SELECT element does not allow to scroll through the available options when using a keyboard. It is not **device independent**.

Javascript code is device independent if:

- users can interact with a website using input and output devices of their choice. Input devices may include pointing devices, keyboards, braille devices, microphones, and others. Output devices may include monitors, speech synthesizers, and braille devices.
- "Device-independent support" does not mean that the browser must support every input or output device. It should offer redundant input and output mechanisms for those devices that are supported. For example, if a browser supports keyboard and mouse, users should be able to interact with all features using either the keyboard or the mouse.

Generally, pages that allow keyboard interaction are also accessible through speech input or a command line interface.

Remember also that a **jump menu** works only if Javascript is enabled and available un users' browsers. There are browsers that do not support Javascript (e.g. for cell phones and PDAs) and there are organizations that turn off Javascript from normal browsers for security reasons.

Include a NOSCRIPT tag with alternative and equivalent content and interaction (i.e. links and forms).

Consider also writing a server-side script to process the URLs passed by the menu and to delive the appropriate page.

- **Learn More:**

- HTML 4.0 standard on [scripts](#);
- HTML 4.0 standard on the [SCRIPT tag](#);
- WAI/ WCAG [guideline 9](#).

20 WCAG 1.0 checkpoint 12.1

20.1 FRAME with valid TITLE [WCAG 12.1]

- **Test Type:** Automatic
- **Test Categories:**
 - W3C WCAG Accessibility
 - Frames
 - ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(i) "Frames shall be titled with text that facilitates frame identification and navigation" according to W3C/WAI [Guideline 12](#) "Provide context and orientation information to help users understand complex pages or elements" and more specifically checkpoint 12.1 "Title each frame to facilitate frame identification and navigation".

The rule checks if all FRAME elements have a **valid** TITLE attribute.

A TITLE attribute of a FRAME is **valid** if it exists, it is not the **empty string** ("") nor the **blank string** (" ") .

- **Issue Description:**

FRAME defined in the document does not have a **valid** TITLE attribute.

A TITLE attribute of a FRAME is **valid** if it exists, it is not the **empty string** ("") nor the **blank string** (" ") .

- **Specific Issues:**

- **TITLE not defined:** Invalid FRAME: missing TITLE attribute
- **TITLE is empty:** Invalid FRAME: TITLE attribute exists but is the empty string ""
- **TITLE is blank:** Invalid FRAME: TITLE attribute exists but is the blank string " "
- **TITLE with html tags:** Invalid FRAME: TITLE attribute exists but contains HTML tags

- **How to fix:**

Please define a **valid** TITLE attribute for the FRAME.

Title the frame to facilitate frame identification and navigation by describing concisely its content and role in the page.

(Since the TITLE attribute is not widely supported, consider that providing titles, as normal text, at the top of the contents of each frame is sufficient for the 508 rule 1194.22(i).)

As a viable alternative, define both the TITLE and NAME attributes to the same string (for example, the lynx textual browser uses the latter).

Remember that a TITLE is **valid** if

- it is a string with no HTML tags;
- it is not the **empty string** ("");
- nor the **blank string** (" ").

- **Why to fix:**

Frames are good for implementing complex navigational structures. However consider that if not implemented correctly they may become an obstacle for many users. You should make sure that your framed page can be accessed by any user using any possible technology in any possible context.

A general **user interface design** guideline is "to provide context and orientation information to help users understand complex pages or elements".

For frames this means that the TITLE attribute should be defined because its value is all that non-graphical browsers show. In fact, each frame will be shown independently for the other ones, making it hard for the user to figure out their relationship. Titles like "content area" or "navigationals" are much more informative than "left" or "top-frame".

Providing contextual information about the relationships between elements can be useful for all users. Complex relationships between parts of a page may be difficult for people with **cognitive disabilities** and people with **visual disabilities** to interpret.

Consider that:

- **cognitive disabilities** do not mean only people that are mentally retarded, but also persons that work under specific adverse contexts, like somebody using an e-ticketing service in an airport hall, or using a reading browser by phone in a noisy environment, somebody using a PDA where a quick decision is required, or during a meeting where changing the mental focus is difficult, or while driving a car and finding the place to go;
- similarly **visually disabled** are also those that forgot their glasses at home, those that use a poorly lit display in a dark environment, those that use a reading browser or screen reader over a phone.

- **Learn More:**

- the HTML 4.0 standard on [frames](#) ;
- the HTML 4.0 standard on the [title attribute](#);
- the W3C/WAI [guideline 12](#);
- the W3C/WAI [techniques for guideline 12](#);
- [checkpoint 9](#) from IBM Web accessibility checklist - version 3.1
- a discussion on [creating quality frames](#);
- J. Nielsen's [Alertbox on frames](#).

20.2 IFRAME with valid TITLE [WCAG 12.1]

- **Test Type:** Automatic

- **Test Categories:**

- W3C WCAG Accessibility
- Frames
- ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(i) "Frames shall be titled with text that facilitates frame identification and navigation" according to W3C/WAI [Guideline 12](#) "Provide context and orientation information to help users understand complex pages or elements" and more specifically checkpoint 12.1 "Title each frame to facilitate frame identification and navigation".

The rule checks if all IFRAME elements (inline frames) have a **valid** TITLE attribute.

A TITLE attribute of an IFRAME is **valid** if it exists, it is not the **empty string** ("") nor the **blank string** (" ").

- **Issue Description:**

IFRAME defined in the document does not have a **valid** TITLE attribute.

A TITLE attribute of a IFRAME is **valid** if it exists, it is not the **empty string** ("") nor the **blank string** (" ").

- **Specific Issues:**

- **TITLE not defined:** Invalid IFRAME: missing TITLE attribute
- **TITLE is empty:** Invalid IFRAME: TITLE attribute exists but is the empty string ""
- **TITLE is blank:** Invalid IFRAME: TITLE attribute exists but is the blank string " "
- **TITLE with html tags:** Invalid IFRAME: TITLE attribute exists but contains HTML tags

- **How to fix:**

Please define a **valid** TITLE attribute for the IFRAME.

Title the frame to facilitate frame identification and navigation by describing concisely its content and role in the page.

(Since the TITLE attribute is seldom supported, consider that providing titles, as normal text, at the top of the contents of each frame is sufficient for the 508 rule 1194.22(i).)

Remember that a TITLE is **valid** if

- it is a string with no HTML tags;
- it is not the **empty string** (“”);
- nor the **blank string** (“ ”).

- **Why to fix:**

Frames are good for implementing complex navigational structures. However consider that their drawback is that in many cases this added complexity may become an obstacle for many users. You should make sure that your framed page can be accessed by any user using any possible technology in any possible context.

A general **user interface design** guideline is “to provide context and orientation information to help users understand complex pages or elements”.

Providing contextual information about the relationships between elements can be useful for all users. Complex relationships between parts of a page may be difficult for people with **cognitive disabilities** and people with **visual disabilities** to interpret.

Consider that:

- **cognitive disabilities** refer also to people that work under specific technical constraints, like somebody using an e-ticketing service in an airport hall, or using a reading browser by phone in a noisy environment, somebody using a PDA where a quick decision is required, or during a meeting where changing the mental focus is difficult, or while driving a car and finding the place to go;
- similarly **visually disabled** are also those that forgot their glasses at home, those that use a poorly lit display in a dark environment, those that use a reading browser or screen reader over a phone.

- **Learn More:**

- HTML 4.0 standard on [frames](#);
- HTML 4.0 standard on [inline frames](#);
- HTML 4.0 standard on the [title attribute](#);
- W3C/WAI [guideline 12](#);
- W3C/WAI [techniques for guideline 12](#);
- J. Nielsen’s [Alertbox on frames](#).

21 WCAG 1.0 checkpoint 12.4

21.1 Form Control with Label [WCAG 12.4]

- **Test Type:** Automatic
- **Test Categories:**
 - W3C WCAG Accessibility
 - Forms
 - ALL TESTS

- **Test Description:**

The test implements 508 rule 1194.22(n) "When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues".

More specifically it implements W3C/WAI checkpoint 12.4 "Associate labels explicitly with their controls."

The test checks if the FORM controls (text fields, menus, radio buttons) are explicitly associated with a LABEL element.

- **Issue Description:**

The page contains a FORM control (INPUT, TEXTAREA or SELECT element) that is not explicitly associated with a LABEL element. Some screen readers need this association to correctly inform the user on which form control they are actually positioned.

- **Specific Issues:**

- **Text field without label:** The text field element (INPUT with type TEXT) is not associated to any label
- **Multi line text field without label:** The multi line text field element (TEXTAREA) is not associated to any label
- **Passowrd field without label:** The password field element (INPUT with type PASSWORD) is not associated to any label
- **Radio button without label:** The radio button element (INPUT with type RADIO) is not associated to any label
- **Checkbox without label:** The checkbox element (INPUT with type CHECKBOX) is not associated to any label
- **File field without label:** The file field element (INPUT with type FILE) is not associated to any label
- **List/menu without label:** The list/menu element (SELECT) is not associated to any label

- **Text field with invalid ID:** The text field element (INPUT with type TEXT) has an ID used by another element in the document
 - **Multi line text field with invalid ID:** The multi line text field element (TEXTAREA) has an ID used by another element in the document
 - **Password field with invalid ID:** The password field element (INPUT with type PASSWORD) has an ID used by another element in the document
 - **Radio button with invalid ID:** The radio button element (INPUT with type RADIO) has an ID used by another element in the document
 - **Checkbox with invalid ID:** The checkbox element (INPUT with type CHECKBOX) has an ID used by another element in the document
 - **File field with invalid ID:** The file field element (INPUT with type FILE) has an ID used by another element in the document
 - **List/menu with invalid ID:** The list/menu element (SELECT) has an ID used by another element in the document
 - **Text field with implicitly assigned label:** The text field element (INPUT with type TEXT) is implicitly associated to a label. Use the explicit method instead
 - **Multi line text field with implicitly assigned label:** The multi line text field element (TEXTAREA) is implicitly associated to a label. Use the explicit method instead.
 - **Password field with implicitly assigned label:** The password field element (INPUT with type PASSWORD) is implicitly associated to a label. Use the explicit method instead.
 - **Radio button with implicitly assigned label:** The radio button element (INPUT with type RADIO) is implicitly associated to a label. Use the explicit method instead.
 - **Checkbox with implicitly assigned label:** The checkbox element (INPUT with type CHECKBOX) is implicitly associated to a label. Use the explicit method instead.
 - **File field with implicitly assigned label:** The file field element (INPUT with type FILE) is implicitly associated to a label. Use the explicit method instead.
 - **List/menu with implicitly assigned label:** The list/menu element (SELECT) is implicitly associated to a label. Use the explicit method instead.
- **How to fix:**

There are three steps to follow to link a label to a control:

 1. assign a unique identifier to the control using the **ID** attribute;
 2. create a **LABEL** element containing the label to associate;
 3. add the **FOR** attribute to LABEL element with the control's ID as value.
 - **Why to fix:**

When navigating through a form using the tab key, a visual user can easily associate the controls with the labels that are placed near them. For a screen reader user, however, this information is not enough. You need to explicitly specify which label is linked to which form control.

The best way to do it is provided by the **LABEL** element. LABEL must contain the actual text labeling the form control and its attribute **FOR** must contain the **ID** of the control.

One LABEL can point only to one control but more that one LABELs can point to the same control. This feature is not implemented by all screen readers yet and so we suggest to assign only one LABEL to each control.

It is possible to associate a label with a control implicitly: just insert both the label and the control as contents of LABEL element. While this technique is suggested by HTML 4.01 Recommendation, it is not supported by all screen readers yet.

- **Learn More:**

- [HTML 4.0 standard on forms in general](#) ;
- [HTML 4.0 standard on the LABEL tag](#);
- [W3C on labeling form controls](#);
- [checkpoint 7](#) from IBM Web accessibility checklist - version 3.1

22 WCAG 1.0 checkpoint 13.6

22.1 Skip repetitive links [WCAG 13.6]

- **Test Type:** Automatic
- **Test Categories:**
 - W3C WCAG Accessibility
 - Links
 - ALL TESTS

- **Test Description:**

The rule implements 508 rule 1194.22(o) "A method shall be provided that permits users to skip repetitive navigation links" and WAI checkpoint 13.6.

The test scans all the navigation bars within the page. If some of them does not follow a **hidden link** that enables to skip the navigation bar, the test prompts the user to verify if such a link is needed.

This test assumes that the **hidden link** mechanism (see below) is used to implement this requirement. The test also assumes that the hidden link is located to the left and close to the navigation bar.

- **Issue Description:**

The page contains a navigation bar that does not follow a **hidden link**. A hidden link (in the context of this test) is a link that is normally not shown by the browser and is implemented like the following example:

```
<A href="#content">
  <IMG alt="jump over products navigation bar" src="spacer.gif"
    width="1" height="1">
</A>
```

and is used to jump over a navigation bar directly to a content section of the page.

NOTE: if there is some alternative mechanism for skipping the navigation bar that this test was not able to identify, you should not worry about this issue.

In addition, if this page is framed (i.e. it is the content of a FRAME) and frames have appropriate titles, then you **don't need** to implement what is suggested by this test, since a frame containing links is already a skippable group of links.

- **Specific Issues:**

- **Missing hidden link:** Couldn't find hidden links used to skip navigation bar
- **Missing named anchor:** Hidden link doesn't point to a fragment of the document

- **How to fix:**

The easiest way to satisfy this requirement is to place a normal textual link (labeled like "skip navigationals") that points to a named anchor in the same page just where the actual content begins.

A more effective way that does not affect the visual appearance of the page is to put, just before the navigation bar, a link pointing to the content of the page whose label is a transparent gif with an appropriate ALT defined. For example:

```
<A href="#content">
  <IMG alt="jump over products navigation bar" src="spacer.gif"
    width="1" height="1">
</A>
```

In this way, users of graphical browsers would not see that link. It can be used, however, by users of non-graphical browsers.

The ALT description should describe what is being skipped, to inform the user about its existence and purpose.

You can also place the **hidden link** at the beginning of the page if you want to provide a single link for skipping one or more groups of links.

Another way to implement the hidden link is to set the color of the text to be same as the background color. The link would not be perceived when colors are shown.

Another way to cope with the 'skip repetitive links' requirement is to implement the navbar as an image map (i.e. collection of 'hotspots'). In this way browsers will treat all the links associated to all the hotspots as a group, making it possible for the user to skip them all together.

For example:

```

<map name="map33">
  <area shape="rect" coords="8,174,253,242" href="index.html"
    alt="Home" title="Home">
  <area shape="rect" coords="6,4,343,86" href="forms.html"
    alt="Login" title="Login">
</map>
```

Another way to implement this requirement is to use frames. In fact frames can be used to group links and, if they contain informative titles, then they can support navigation very well.

- **Why to fix:**

Web designers often place a set of navigational links at a standard location - often across the top, bottom, or side of a page. While very useful in general, navigation bars have some disadvantage.

Consider a non-disabled visitor returning to a web page or site and knowing that s/he wants to view the contents of that particular page instead of selecting a navigation link to go to another page. S/he may simply look past the links and begin reading wherever the desired text is located.

The situation is very different for those who use screen readers or other types of assistive technologies. In fact, it is a tedious and time-consuming chore to wait for the assistive technology to work through and announce each of the standard navigational links before getting to the page content.

To reduce the consequences of this problem there should be a mechanism for users to skip repetitive navigational links.

- **Learn More:**

- HTML 4.0 standard on [links in general](#) ;
- HTML 4.0 standard on the [anchor names](#);
- W3C/WAI on [grouping and bypassing links](#);
- a [detailed discussion](#) on linearizing layout tables and its effects on navigation;
- you can check how links are shown after linearizing tables by using the [LIFT Online](#) service that presents a page as would be displayed by the [lynx textual browser](#).
- [checkpoint 8](#) from IBM Web accessibility checklist - version 3.1

23 WCAG 1.0 checkpoint 14.1

23.1 Use clear language for site's content [WCAG 14.1]

- **Test Type:** Manual

- **Test Categories:**

- W3C WCAG Accessibility
- Manual
- ALL TESTS

- **Test Description:**

The rule implements the W3C/WAI [Guideline 14](#) "Ensure that documents are clear and simple." and more specifically checkpoint 14.1.

The rule prompts the user to check if the text uses the clearest and the simplest language appropriate for a site's content.

This is a manual test.

- **Issue Description:**

Consistent page layout, recognizable graphics, and easy to understand language benefit all users. In particular, they help people with cognitive disabilities or who have difficulty reading.

Using clear and simple language promotes effective communication. Access to written information can be difficult for people who have cognitive or learning disabilities. Using clear and simple language also benefits people whose first language differs from your own, including those people who communicate primarily in sign language.

- **How to fix:**

The page contains text: please check if it is easy to read by everyone, especially by people with reading and/or cognitive disabilities.

- **Why to fix:**

The following are some of the writing style suggestions that should help make the content of your site easier to read.

- Strive for clear and accurate headings and link descriptions (some users browse by jumping from link to link and listening only to link text).
- State the topic of the sentence or paragraph at the beginning of the sentence or paragraph (this is called "front-loading"). This will help both people who are skimming visually, but also people who use speech synthesizers.
- Limit each paragraph to one main idea.

- Avoid slang, jargon, and specialized meanings of familiar words, unless defined within your document.
- Favor words that are commonly used. For example, use "begin" rather than "commence" or use "try" rather than "endeavor".

To help determine whether your document is easy to read, consider using the [Gunning-Fog](#) reading measure. This algorithm, when run on a text, generally produces a lower score when content is easier to read. If you find your Index soaring into the [teens](#) (or higher!), beware: you've lost most of your audience in the dense fog!

- **Learn More:**

- W3C Accessibility [Guideline 14](#);
- W3C on [Writing style](#);
- the [Gunning-Fog](#) reading measure, an example on [how it works](#).

A Test Categories

ALL TESTS This category contains all the available tests.

508 Accessibility This category contains tests implementing **Section 508 1194.22** guidelines.

Consult [Section 508](#) for additional material on web accessibility issues.

]

This category contains the **accessibility (priority 1)** rules specified by the W3C/WAI group ([Guidelines 1.0](#)). Consult [WAI references](#) for additional material on web accessibility issues.

Images This category contains tests dealing with images embedded in HTML documents.

Manual This category contains tests requiring manual inspection of the HTML documents.

Forms This category contains tests dealing with FORMs and related elements.

Frames This category contains tests dealing with FRAMEs and related elements.

Tables This category contains tests dealing with TABLEs and related elements.

Imagemaps This category contains tests dealing with image maps.

Links This category contains tests dealing with images within links and navigation bars.

Scripts This category contains tests dealing with programmatic objects.

Suggestions This category contains tests that suggest ways in which to use the LIFT Online service on www.usablenet.com.