Norwegian Digitalisation Agency
Norwegian Authority for Universal Design of ICT

ANNEX I

WCAG Interpretation and Test Rule Documentation in the WAI-Tools Project

Authority for Universal Design of ICT

Norwegian Digitalisation Agency

Table of Content

Preface

As a part of the WAI-Tools project, the Norwegian Digitalisation Agency (previously the Agency for Public Management and eGovernment – Difi) constitute the documentation of their internal interpretation of the Web Content Accessibility Guidelines (WCAG) and Accessibility Conformance Testing (ACT) Rules. This documentation is the part of deliverable "D2.4 Working Instance of Digdir Observatory" under Work Package 2[1] in the WAI-Tools project.

The purpose is to document the Agency's interpretation[2] of the WCAG success criteria and coverage of these success criteria by ACT Rules. In addition, the gap analysis between WCAG success criteria and ACT Rules has also been documented, such as aspects of the success criteria that are not covered by ACT Rules in WAI-Tools and aspects of the ACT Rules that go beyond the scope of the success criteria. This documentation is subdivided into one chapter per WCAG success criteria. Each chapter defines:

- specific details relevant to WCAG success criteria,
- purpose of success criteria,
- user accessibility needs (functional performance statements (fps)) (based on EN 301 549[3]),
- interpretation and specification of the success criteria (Norwegian Digitalisation Agency),
- coverage of Success Criteria by ACT rules developed in WAI-Tools,
- gap analysis, and ACT Rules related to each WCAG success criterion.

The Norwegian Digitalisation Agency's existing interpretation and specification of WCAG success criteria are limited to WCAG 2.0 with a brief overview of associated test procedures. The Norwegian Digitalisation Agency also has a fundamental interpretation and specification of the new WCAG 2.1 success criteria but does not have associated test procedures. Our contributions to the test rules development were based on our existing interpretation of the WCAG 2.0 and 2.1 success criteria.

Each chapter of this documentation has been written using a template specially designed for this documentation. This template contains two modules, one for WCAG success criteria and the other one for ACT Rule. To understand how these two modules have been filled out by the authors, the interested readers are referred to the template which can be found in Chapter 55 at the end of this documentation.

[1] The objective of Work Package 2 is "to apply the test rules and their implementations in exemplary demonstrators of monitoring observatories on national levels".
[2] Interpretation and overview of test procedures for WCAG 2.0 A and AA, https://uu.difi.no/om-oss/english/interpretation-and-overview-test-procedures-wcag-20-and-aa
[3] ETSI EN 301 549 V1.1.2 (2018-08), Accessibility requirements suitable for public procurement of ICT products and services in Europe

Background

This documentation has been written between 01 July 2020 and 31 October 2020 as one of the main results of deliverables in a European Commission (EC) co-funded project named Web Accessibility Initiative - Advanced Decision Support Tools for Scalable Web Accessibility Assessments (WAI-Tools) Project under Horizon 2020 Program[4] (780057).

WAI-Tools Project

WAI-Tools is an Innovation Action project that has been active for 39 months between 1 November 2017 and 31 January 2021.

In WAI-Tools Work Package 1 the objective is "to build, implement, and validate a set of open test rules, developed through the international consensus process of W3C". These test rules are developed by project staff through the W3C consensus process[5], implemented in open source engines developed and maintained by Deque, FCID (University of Lisbon), and Siteimprove.

The Norwegian Digitalisation Agency (previously the Agency for Public Management and eGovernment – Difi) is responsible for deliverable "Documentation of Digdir's WCAG interpretation and ACT Rules" under deliverable "D2. Deployment of Test Rules" under Work Package 2[6]. In this documentation, interpretation, and mapping of test rules developed under this project to WCAG 2.0/2.1 have been documented.

Web Accessibility Directive (WAD)

The Web Accessibility Directive (WAD), Directive (EU) 2016/2102[7], in force since 22 December 2016, aims to provide people with disabilities with better access to the public services websites and mobile applications. The Directive covers websites and applications of the public sector with a limited number of exceptions e.g. broadcasters, live streaming.

It is also required in WAD and its corresponding implementing act[8], that the member states are required to monitor the conformity of websites and mobile applications of public sector bodies with the accessibility requirements provided for in Article 4 of Directive (EU) 2016/2102 using a simplified monitoring method to detect non-compliance and an in-depth monitoring method to verify compliance.

Article 6 of Directive (EU) 2016/2102 refers to European standard EN 301 549. The content of websites and mobile applications meets harmonised standards and shall ensure at least a level of accessibility equivalent to that ensured by European standard EN 301 549, which in turn refers to WCAG 2.1 success criteria. In other words, it has been statutory throughout Europe for public sector bodies websites to be designed in accordance with WCAG 2.1.

---

[4] Web Accessibility Initiative - Advanced Decision Support Tools for Scalable Web Accessibility Assessments (WAI-Tools) Project, https://www.w3.org/WAI/about/projects/wai-tools/
[5] ACT Rules Community Group, https://act-rules.github.io/pages/about
[6] The objective of Work Package 2 is "to apply the test rules and their implementations in exemplary demonstrators of monitoring observatories on national levels".
[7] Directive (EU) 2016/2102 of the European Parliament and of the Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies. https://eur-lex.europa.eu/eli/dir/2016/2102/oj
[8] Commission Implementing Decision (EU) 2018/1524 of 11 October 2018 establishing a monitoring methodology and the arrangements for reporting by Member States in accordance with Directive (EU) 2016/2102, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.256.01.0108.01.ENG&toc=OJ:L:2018:256:TOC

In addition, we know from our experience and with reference to paragraph 56 of Directive (EU) 2016/2102 that there are different interpretations of WCAG success criteria across member states which requires harmonisation to monitor conformity. WAI-Tools is an initiative of a harmonized interpretation of WCAG, with associated test rules. This interpretation of WCAG is implicit in the test rules developed in WAI-Tools.

The terms compliance and non-compliance are not defined in Directive (EU) 2016/2102. The directive refers to European standard EN 301 549, where the determination of compliance is defined in the following way[9]:

> *"Compliance is achieved either when the pre-condition is true and the corresponding test (in Annex C in EN 301 549) is passed, or when the pre-condition is false (i.e. the pre-condition is not met or not valid)."*

For each of the requirements regarding web pages, the pre-conditions and tests are stated in the following way:

> *X.Y.Z [Name of requirement]*

Type of assessment	*Inspection*
Pre-conditions	*1. The ICT is a web page.*
Procedure	*1. Check that the web page does not fail WCAG 2.1 Success Criteria X.Y.Z [Name of Success Criteria)*
Result	*Pass: Check 1 is true* *Fail: Check 1 is false*

Web Content Accessibility Guidelines (WCAG) 2.0/2.1)

The Web Content Accessibility Guidelines (WCAG) standard is developed through a W3C process[10] in cooperation with individuals and organisations around the world. WCAG defines guidelines to make web content more accessible for people with disabilities. These accessibility guidelines cover a wide range of disabilities such as visual, auditory, physical, auditory, speech, cognitive, language, learning, and neurological disabilities.

WCAG 2.0 was published on 11 December 2008. WCAG 2.1 was published on 5 June 2018. There are 13 guidelines organised under 4 principles: perceivable, operable, understandable, and robust. For each guideline, there are testable success criteria, which are at three levels of conformance: A, AA, and AAA. All requirements ("success criteria") from 2.0 are included in 2.1. The 2.0 success criteria are exactly the same (verbatim, word-for-word) in 2.1.

The accessibility requirements in WCAG 2.0/2.1 are mostly written in a technology-neutral language which requires interpretation of what exactly is required to comply with these requirements especially when it comes to specific technology.

Accessibility Conformance Testing (ACT) Rules

[9] ETSI EN 301 549 V1.1.2, Accessibility requirements suitable for public procurement of ICT products and services in Europe, Annex C

[10] How WAI Develops Accessibility Standards through the W3C Process: Milestones and Opportunities to Contribute, https://www.w3.org/WAI/standards-guidelines/w3c-process/

An ACT Rule is a plain language explanation of how to test a specific type of content for specific aspects of WCAG accessibility requirements ("success criteria")[11]. An ACT Rule describes the expected results of accessibility testing tools and/or methodologies when running a conformance test against a specific part of the applicable test subject. In the context of WCAG, ACT Rules test for failures in satisfying Success Criteria. ACT rules are written for the testing process and usually more specific and limited to a single aspect of a WCAG accessibility requirement. Each ACT Rule contains a description of the type of content under test, the test to perform, and the expected results.

The ACT Rules were written and developed by the ACT Rules Community Group (ACT-R)[12] using W3C ACT Rules Format[13]. One of the objectives of this project was to establish an open community by bringing together key partners from industry, government, and research, who can develop authoritative resources on accessibility conformance testing (ACT).

The ACT Rules Community Group (ACT-R) is an open forum with the aim to document and harmonise the interpretation of W3C accessibility standards such as WCAG and WAI-ARIA. ACT-R is an initiative to keep working on the development of ACT Rules and harmonising the interpretation of W3C accessibility standards even though the WAI-Tools project has been ended. This long-term process of development of ACT Rules will continuously increase the coverage of WCAG success criteria requirements which later can be used not only in WAD simplified but in in-depth monitoring as well.

According to W3C ACT Rules Format, there are two types of ACT Rules:

- **Atomic rules** describe how to test a specific type of solution such as website content. It contains a precise definition of what elements, nodes or other "parts" of a test subject are to be tested, and when those test targets are considered to pass or fail the rule. Atomic rules are to be kept small and specific. These rules test a single "condition" and do so without using the outcomes from other rules.
- **Composite rules** describe how the outcomes of multiple atomic rules can be combined into a single outcome for each test target. In short, a composite rule can have multiple "conditions", each of these tested in separate atomic rules.

The implementation of ACT Rules is based on test mode. The Test Mode[14] of a rule describes how a test was carried out, in the case of ACT Rule, implementation can be automated, semi-automated, or manual. In an automated implementation, the test is carried out automatically means such as software tools and without any human intervention. In a manual implementation, the test is carried out by human evaluators, it not just includes actual test procedure carried out by the evaluators but also where the evaluators helped by instructions or guidance provided by software tools. An implementation is considered as semi-automated, where the test is carried out with the help of software tools, but human evaluation or judgment is still required to decide the outcome of the test.

The outcome of evaluation using ACT Rule on a test subject or one of its constituent test targets can be one of the three following types:

- **Inapplicable:** No part of the test subject matches the applicability of the rule
- **Passed:** A test target meets all expectations of the rule
- **Failed:** A test target does not meet all expectations of the rule

[11]Accessibility Conformance Testing (ACT) Rules Format 1.0, https://www.w3.org/TR/act-rules-format/#intro
[12] ACT RULES COMMUNITY GROUP, https://www.w3.org/community/act-r/
[13] Accessibility Conformance Testing (ACT) Rules Format 1.0, https://www.w3.org/TR/act-rules-format/
[14] Evaluation and Report Language (EARL) 1.0 Schema – TestMode Class, https://www.w3.org/TR/EARL10-Schema/#TestMode

The outcome of an ACT Rule “Passed” does not mean that a test subject is compliance with all the applicable requirements of WCAG success criteria but only compliance with a specific aspect of WCAG success criteria covered by that ACT Rule. There are a few manual ACT Rules that possibly fully satisfy some of the WCAG success criteria. However, in general, “Passed” and “Inapplicable” outcomes of ACT Rules require further testing or manual checks.

ACT Rules are open source and can be used by member states and other organisations to implement these rules in the development of manual testing procedures, semi-automatic and/or automatic testing tools.

Summary of our findings

The WCAG success criteria are generic, complex, and require a wide range of various aspects and/or situations to be compliant. It is hard to cover all the aspects and/or situations required by WCAG success criteria, ACT Rules developed in the WAI-Tools project are very specific and atomic to a specific aspect and/situation that cover a small portion of WCAG success criteria. There can be several ACT Rules to cover all applicable aspects of WCAG success criteria accessibility requirements, which means that there could still be a gap in compliance with all aspects of a WCAG success criterion.

Currently, the ACT Rules developed in WAI-Tools can be used in and limited to testing contents created using web technologies, such as Hypertext Markup Language (HTML), Cascading Style Sheets (CSS-2018), Accessible Rich Internet Applications (WAI-ARIA), Scalable Vector Graphics (SVG2).

There have been 07 Composite ACT Rules (uses outcomes from 26 Input Rules[15] that are atomic ACT Rules) and 49 Atomic ACT Rules developed until October 2020 in the WAI-Tools project. In total including both composite and atomic, there are 56 ACT Rules that have been documented in this documentation and covers compliance and/or non-compliance of certain accessibility requirements of 30 WCAG success criteria (SC). The documentation includes 4 atomic ACT Rules that are not related to any WCAG success criteria conformance.

The WAI-Tools project is moving forward, and the goal is to develop 70 ACT Rules by the end of January 2021. The overview of ACT Rules that have been documented in this documentation along with WCAG success criteria covered by these rules are shown in Table 1 below where N/A stands for not applicable:

NOTE: We documented each ACT Rule in this documentation when the rule has been completed after the intensive development process established by the project. There have been many changes along the way until the end of the project period. We may not update all the changes that were made after the ACT Rules are documented in this documentation. Therefore, the dates when the ACT Rule was last updated (completed) and documented in this documentation are also added in each ACT Rule module.

Table 1. Overview of ACT Rules and WCAG success criteria

#	ACT-R ID	ACT Rule Name	WCAG SC	Type
1	5c01ea	ARIA state or property is permitted	N/A	Atomic
2	73f2c2	autocomplete attribute has valid value	1.3.5	Atomic
3	97a4e1	Button has non-empty accessible name	4.1.2	Atomic
4	e086e5	Form field has non-empty accessible name	4.1.2	Atomic
5	2779a5	HTML page has non-empty title	2.4.2	Atomic
6	b5c3f8	HTML has lang attribute	3.1.1	Atomic
7	5b7ae0	HTML page lang and xml:lang attributes have matching values	3.1.1	Atomic
8	23a2a8	Image has non-empty accessible name	1.1.1	Atomic

[15] W3C Accessibility Conformance Testing (ACT) Rules Format 1.0 – Input Rules, https://www.w3.org/TR/act-rules-format/#input-rules

#	ACT-R ID	ACT Rule Name	WCAG SC	Type
9	c487ae	Link has non-empty accessible name	4.1.2, 2.4.4, 2.4.9	Atomic
10	bc659a	meta element has no refresh delay	2.2.1, 2.2.4, 3.2.5	Atomic
11	674b10	Role attribute has valid value	4.1.2	Atomic
12	4e8ab6	Element with role attribute has required states and properties	4.1.2	Atomic
13	de46e4	Element with lang attribute has valid language tag	3.1.2	Atomic
14	bf051a	HTML page lang attribute has valid language tag	3.1.1	Atomic
15	6cfa84	Element with aria-hidden has no focusable content	1.3.1, 4.1.2	Atomic
16	cae760	iframe element has non-empty accessible name	4.1.2	Atomic
18	3ea0c8	id attribute value is unique	4.1.1	Atomic
19	5f99a7	aria-* attribute is defined in WAI-ARIA	N/A	Atomic
20	6a7281	ARIA state or property has valid value	N/A	Atomic
21	c3232f	video element visual-only content has accessible alternative	1.2.1	Composite
	fd26cf	video element visual-only content is media alternative for text	N/A	Atomic
	ee13b5	video element visual-only content has transcript	N/A	Atomic
	ac7dc6	video element visual-only content has description track	N/A	Atomic
	d7ba54	video element visual-only content has audio track alternative	N/A	Atomic
22	e7aa44	audio element content has text alternative	1.2.1	Composite
	2eb176	audio element content has transcript	N/A	Atomic
	afb423	audio element content is media alternative for text	N/A	Atomic
23	eac66b	video element auditory content has accessible alternative	1.2.2	Composite
	f51b46	Video element auditory content has captions	N/A	Atomic
	ab4d13	Video element content is media alternative for text	N/A	Atomic
24	c5a4ea	video element visual content has accessible alternative	1.2.3	Composite
	1a02b0	Video element visual content has transcript	1.2.8	Atomic
	1ea59c	Video element visual content has audio description	N/A	Atomic
	ab4d13	Video element content is media alternative for text	N/A	Atomic
	f196ce	Video element visual content has description track	N/A	Atomic
25	1ec09b	Video element visual content has strict accessible alternative	1.2.5	Composite
	1ea59c	Video element visual content has audio description	N/A	Atomic
	ab4d13	Video element content is media alternative for text	N/A	Atomic
	f196ce	Video element visual content has description track	N/A	Atomic

#	ACT-R ID	ACT Rule Name	WCAG SC	Type
26	80af7b	Focusable element has no keyboard trap	2.1.2	Composite
	ebe86a	Focusable element has no keyboard trap via non-standard navigation	N/A	Atomic
	a1b64e	Focusable element has no keyboard trap via standard navigation	N/A	Atomic
27	e6952f	Attribute is not duplicated	4.1.1	Atomic
28	c4a8a4	HTML page title is descriptive	2.4.2	Atomic
29	cc0f0a	Form control label is descriptive	2.4.6	Atomic
31	59796f	Image button has non-empty accessible name	1.1.1, 4.1.2	Atomic
32	ff89c9	Aria required context role	1.3.1	Atomic
33	bc4a75	Aria required owned element	1.3.1	Atomic
34	ffd0e9	Heading has non-empty accessible name	1.3.1, 2.4.6	Atomic
35	7d6734	svg element with explicit role has non-empty accessible name	1.1.1	Atomic
36	9eb3f6	Image filename is accessible name for image	1.1.1	Atomic
38	b20e66	Links with identical accessible names have equivalent purpose	2.4.9	Atomic
39	4b1c6c	iframe elements with identical accessible names have equivalent purpose	4.1.2	Atomic
40	e88epe	Image not in the accessibility tree is decorative	1.1.1	Atomic
41	80f0bf	audio or video avoids automatically playing audio	1.4.2	Composite
	4c31df	audio or video that plays automatically has a control mechanism	N/A	Atomic
	aaa1bf	audio or video that plays automatically has no audio that lasts more than 3 seconds	N/A	Atomic
42	b33eff	Orientation of the page is not restricted using CSS transform property	1.3.4	Atomic
44	a25f45	headers attribute specified on a cell refers to cells in the same table element	1.3.1	Atomic
45	d0f69e	Table header cell has assigned data cells	1.3.1	Atomic
47	b4f0c3	meta viewport allows for zoom	1.4.4	Atomic
48	36b590	Error message describes invalid form field value	3.3.1	Atomic
49	fd3a94	Links with identical accessible names and context serve equivalent purpose	2.4.4, 2.4.9	Atomic
50	59br37	Zoomed text node is not clipped with CSS overflow	1.4.4	Atomic
51	5effbb	Link in context is descriptive	2.4.4, 2.4.9	Atomic
53	ffbc54	No keyboard shortcut uses only printable characters	2.1.4	Atomic
54	c249d5	Device motion based changes to the content can be disabled	2.5.4	Atomic

#	ACT-R ID	ACT Rule Name	WCAG SC	Type
55	efbfc7	Text content that changes automatically can be paused, stopped or hidden	2.2.2	Atomic
57	46ca7f	Element marked as decorative is not exposed	N/A	Atomic
58	0ssw9k	Scrollable element is keyboard accessible	2.1.1	Atomic
62	8fc3b6	Object element rendering non-text content has non-empty accessible name	1.1.1	Atomic
65	ucwvc8	HTML page language subtag matches default language	3.1.1	Atomic
67	7677a9	Device motion based changes to the content can also be created from the user interface	2.5.4	Atomic
68	qt1vmo	Image accessible name is descriptive	1.1.1	Atomic

Further use of this documentation

This section describes the value of this documentation and suggests how it can be used and further developed. After WAD was introduced and adopted by the member states, the accessibility of websites, applications, and ICT services is becoming one of the legal requirements around the member states in Europe.

According to WAD and the corresponding implementing act, all member states are required to perform monitoring of websites and mobile applications. This monitoring requirement of WAD made it more important for the member states to have a harmonized testing methodology to perform consistent monitoring.

ACT Rules can be used to monitor the degree of compliance with WAD accessibility requirements, which will make it possible for the organisations to know how to fulfil accessibility requirements required by the directive in a consistent manner. Monitoring bodies in member states can use this documentation of ACT Rules and interpretation of accessibility requirements to provide guidelines to businesses and perform controls, audits, and monitoring in their countries.

The Norwegian Agency tasks also include to perform pilot for simplified and in-depth monitoring in line with Directive (EU) 2016/2102 on a limited set of websites under Work Package 2, to establish a “demonstrator”. The testing of websites was performed using testing tools and/or open-source engines developed by Deque, FCID (University of Lisbon), and Siteimprove that have adopted the test rules developed under the project. Note that the pilot monitoring documentation which includes a description of the sampling approach, testing, analysis, data model, and reporting results is delivered as a separate deliverable in the project.

These efforts provide valuable input and documentation on how a monitoring body enforces the Web Accessibility Directive through the use of the ACT Rules and deliverables from the WAI-Tools project.

1 1.1.1 Non-text Content (Level A)

1.1 About Success Criteria 1.1.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#non-text-content

Success Criteria	All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. • **Controls, Input:** If non-text content is a control or accepts user input, then it has a name that describes its purpose. (Refer to Guideline 4.1 for additional requirements for controls and content that accepts user input.) • **Time-Based Media:** If non-text content is time-based media, then text alternatives at least provide descriptive identification of the non-text content. (Refer to Guideline 1.2 for additional requirements for media.) • **Test:** If non-text content is a test or exercise that would be invalid if presented in text, then text alternatives at least provide descriptive identification of the non-text content. • **Sensory:** If non-text content is primarily intended to create a specific sensory experience, then text alternatives at least provide descriptive identification of the non-text content. • **CAPTCHA:** If the purpose of non-text content is to confirm that content is being accessed by a person rather than a computer, then text alternatives that identify and describe the purpose of the non-text content are provided, and alternative forms of CAPTCHA using output modes for different types of sensory perception are provided to accommodate different disabilities. • **Decoration, Formatting, Invisible:** If non-text content is pure decoration, is used only for visual formatting, or is not presented to users, then it is implemented in a way that it can be ignored by assistive technology.
Purpose	The intent of this Success Criteria is to make information conveyed by non-text content accessible through the use of a text alternative. Text alternatives are a primary way for making information accessible because they can be rendered through any sensory modality (for example, visual, auditory or tactile) to match the needs of the user.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without hearing Secondary relationship • Usage with limited hearing • Usage with limited cognition

Digdir interpretation and specification of the success criteria	This requirement means that there must be text alternatives for non-text content with the same purpose as the non-text content. The wording of the success criteria includes a list of a number of content types and situations where the requirement involves more detailed specification of the text alternative for each individual situation, e.g. CAPTCHA.
	
	How to achieve compliance with the requirement:
	
	• Images that are a sensory experience or test have a short text alternative that provides a descriptive identification
	• Images that are meaningful have a short text alternative that provides the same information as the image
	• Images that are complex have both a short and a supplementary text alternative that provides the same information as the image
	• For linked images, information on the link target is provided by one of the following alternatives:
	  ○ The link text.
	  ○ The text alternative for the image.
	  ○ The link text and the text alternative for the image together.
	• For clickable areas in image maps, information on the link target is provided by the text alternative
	• If CAPTCHAs with non-text content are used, the following requirements are met:
	  ○ There are at least two types of CAPTCHA, e.g. in the form of an image, sound or text.
	  ○ Non-text content has a text alternative that identifies and describes the purpose.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **23a2a8:** HTML img elements or any HTML element with the semantic role of img except if the element has a hidden state of “true”, has an accessible name that is not empty ("") or has a semantic role of “none” or presentation. • **59796f:** HTML input elements where state of the type attribute is image, that is included in the accessibility tree, has an accessible name that is not empty (""). • **7d6734:** HTML elements in the SVG namespace with an explicit semantic role of either img, graphics-document, graphics-symbol, that is included in the accessibility tree has an accessible name that is not empty (""). • **9eb3f6:** HTML element that is included in the accessibility tree and has non-empty ("") accessible name where element is img with an accessible name that is equivalent to the filename of at least one of the image sources in its source set; or input element with a type of image that is input element in the image button state with an accessible name that is equivalent to the filename specified in its src attribute, the accessible name serves an equivalent purpose to the non-text content. If there are several image sources, then the accessible name must accurately describe all of them. In cases where the accessible name contains the file extension and the text content in the link does not accurately describe the image, the presence of the file extension in the accessible name is redundant and results in the accessible name not accurately describing the image. • **e88epe:** HTML img, canvas or svg elements that are visible which are excluded – that is not included in the accessibility tree; or ignored svg – that is an svg with an empty ("") accessibility name and a semantic role of graphics-document;

	or ignore canvas – that is a canvas with an empty (“”) accessible name and no explicit semantic role; are purely decorative. There is an exception that this rule never applies to elements that have an ancestor in the flat tree that is named from the author; or the element is an img element where the current request’s state is not completely available. For example, an image ignored because of an ancestor with a name from the author is when the image is a descendant of a button element that uses aria-label for its accessible name. • **qt1vmo:** HTML img, canvas or svg element that is visible and has non-empty accessible name that servers an equivalent purpose to the non-text content of the test target, except if the element has an ancestor in the flat tree that is named from author; or the element is an img element where the current request’s state is not completely available. • **8fc3b6:** HTML object element that is included in the accessibility tree and embeds a resource with an image MIME type or an audio or video MIME type has a non-empty ("") accessible name.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Images that are a sensory experience or test have a short text alternative that provides a descriptive identification. • Images that are complex have both a short and a supplementary text alternative that provides the same information as the image. • Linked images have text alternatives that, together with any link text, describe the purpose of the link (also see SC 2.4.4). • For clickable areas in image maps, information on the link target is provided by the text alternative (also see SC 2.4.4) • CAPTCHA, where non-text content is used has at least two types of CAPTCHA (e.g. in the form of an image, sound, or text) with text alternatives that identify and describe the purpose of the non-text content. This is not a comprehensive list of all aspects that may be covered by this success criteria. Aspects of the ACT rules that go beyond the scope of the success criteria: • None

1.2 Test Rule: Image has non-empty accessible name

- **WAI-Tools rule number:** 8
- **ACT Rules id:** 23a2a8
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 30.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/23a2a8

Description	This rule checks that each image either has a non-empty accessible name or is marked up as decorative.

Accessibility Requirements	1.1.1 Non-text Content (Level A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling
Applicability	The rule applies to HTML img elements and HTML elements with the semantic role of img, except if the element has a hidden state of "true".
Expectation	Each target element has an accessible name that is not empty ("") or has a semantic role of none or presentation.
Assumptions	There are currently no assumptions.
Accessibility Support	There is a known combination of a popular browser and assistive technology that does not by default support title as an accessible name. There are several popular browsers that do not treat images with empty alt attribute as having a role of presentation but instead add the img element to the accessibility tree with a semantic role of either img or graphic. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of img and fail this rule with some technology but users of other technologies would not experience any accessibility issue. Images can have their role set to presentation through an empty alt attribute. Presentational Roles Conflict Resolution does not specify what to do if such an image is focusable (it only specifies what to do in case of explicit role="none" or role="presentation"). Some browsers expose these images and some don't. Thus, this rule may fail for technologies that expose these without creating an accessibility issue for users of other technologies.

1.3 Test Rule: Image button has non-empty accessible name

- **WAI-Tools rule number:** 31
- **ACT Rules id:** 59796f
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 29.06.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/59796f

Description	This rule checks that each image button element has a non-empty accessible name.

Accessibility Requirements	1.1.1 Non-text Content (Level A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing

	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any input element where the state of the type attribute is image, and that is included in the accessibility tree.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	This rule assumes that all image buttons are user interface components as defined by WCAG 2.
Accessibility Support	There is a known combination of a popular browser and assistive technology that does not by default support title as an accessible name.

1.4 Test Rule: svg element with explicit role has non-empty accessible name

- **WAI-Tools rule number:** 35
- **ACT Rules id:** 7d6734
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 17.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/7d6734

Description	This rule checks that each SVG image element that is explicitly included in the accessibility tree has a non-empty accessible name.

Accessibility Requirements	1.1.1 Non-text Content (Level A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Three, CSS Styling
Applicability	The rule applies to any element in the SVG namespace with an explicit semantic role of either img, graphics-document, graphics-symbol, that is included in the accessibility tree.
Expectation	Each target element has an accessible name that is not empty.
Assumptions	This rule assumes that the presence of one of the roles outlined in the applicability indicates the authors intent to include the element in the accessibility tree and thus convey information to the user about that element.

Accessibility Support	The HTML Accessibility API Mappings specify that the <svg> element has an implicit role of graphics-document. However, browser support for the graphics-document role and the SVG Accessibility API Mappings is inconsistent. This rule is limited to the explicit use of roles, as a clear indication that content should convey meaning, until the SVG Accessibility API Mappings is more stable and browser support is more consistent. Browser and assistive technology support for SVG <title> and <desc> elements are currently inconsistent. Using WAI ARIA in combination with the img role for non-decorative <svg> elements significantly improves accessibility browser support. Until browser support for the SVG Accessibility API Mappings is more consistent it is recommended to explicitly remove decorative <svg> elements from the accessibility tree.

1.5 Test Rule: Image filename is accessible name for image

- **WAI-Tools rule number:** 36
- **ACT Rules id:** 9eb3f6
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 03.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/9eb3f6

Description	This rule checks that image elements that use their source filename as their accessible name do so without loss of information to the user.

Accessibility Requirements	1.1.1 Non-text Content (Level: A)
	
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	The rule applies to any HTML element that is included in the accessibility tree and has a non-empty ("") accessible name, for which one of the following is true: • img: the element is an img with an accessible name that is equivalent to the filename of at least one of the image sources in its source set; or • input image: the element is an input element in the Image Button state with an accessible name that is equivalent to the filename specified in its src attribute. When comparing accessible name and filename, difference in letter casing, leading and trailing whitespace should be ignored.
Expectation	Each test target has an accessible name that serves an equivalent purpose to the non-text content. If there are several image sources, then the accessible name must accurately describe all of them. **Note:** It is fairly common for CMS or other tools to default the alt-text of an image to its filename if no alt-text is provided. However, these names are usually not

	descriptive (often due to the presence of the file extension). This rule uses this heuristic to pinpoints cases where the accessible name should be looked at by human testers. This rule does not automatically decide in which case a filename is correct (notably, it does not automatically decide whether adding the file extension is acceptable).
Assumptions	This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content).
Accessibility Support	There are no major accessibility support issues known for this rule.

1.6 Test Rule: Image not in the accessibility tree is decorative

- **WAI-Tools rule number:** 40
- **ACT Rules id:** e88epe
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 27.05.2020
- **Last updated in GitHub:** 14.05.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/e88epe

Description	This rule checks that visible img, svg and canvas elements that are ignored by assistive technologies are decorative

Accessibility Requirements	1.1.1 Non-text Content (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, Accessibility Tree, CSS Styling
Applicability	Any img, canvas or svg element that is visible and for which one of the following is true: • **excluded**: The element is not included in the accessibility tree; or • **ignored svg**: The element is a svg with an empty ("") accessible name and a semantic role of graphics-document; or • **ignored canvas**: The element is a canvas with an empty ("") accessible name and no explicit semantic role; or **Exception**: This rule never applies to elements for which one of the following is true: • The element has an ancestor in the flat tree that is named from author; or • The element is an img element where the current request's state is not completely available. **Note**: An example of an image ignored because of an ancestor with named from author is when the image is a descendant of a button element that uses aria-label for its accessible name.

Expectation	Each test target is purely decorative. **Note:** It is relatively common for an informative image such as an icon to be marked up as decorative if the text alternative is adjacent to the image. This is a conforming alternative version for the image. This fails the rule but meets conformance requirement 1 of WCAG 2.1.
Assumptions	svg elements with a semantic role of graphics-document and with an empty ("") accessible name are ignored by assistive technologies tested for this rule. If some assistive technology does not ignore these elements, and that assistive technology is required for conformance, passing this rule does not ensure all decorative svg elements can be ignored, and the success criterion 1.1.1 Non-text content may still not be satisfied. The same is true for canvas elements with no semantic role and an empty ("") accessible name. A web page with informative images without an accessible name may conform to WCAG 2.1 Level A when the information provided by that image is available elsewhere on the web page itself. For example, if an equivalent text is adjacent to the image, or if the text alternative is included in the accessible name of a parent element.
Accessibility Support	No accessibility support issues known.

1.7 Test Rule: Object element rendering non-text content has non-empty accessible name

- **WAI-Tools rule number:** 62
- **ACT Rules id:** 8fc3b6
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 06.10.2020
- **Last updated in GitHub:** 05.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/8fc3b6

Description	This rule checks that each object element rendering non-text content has a non-empty accessible name.

Accessibility Requirements	1.1.1 Non-text Content (Level A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling, Accessibility Tree
Applicability	This rule applies to any object element that is included in the accessibility tree and embeds a resource with an image MIME type or an audio or video MIME type.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	The object element is not rendered for presentational purposes. If the object is decorative and not marked as decorative then the rule might fail but the success criterion might still be satisfied.

Accessibility Support	Non-supported media formats make screen readers render the text content of the element instead of other attributes.

1.8 Test Rule: Image accessible name is descriptive

- **WAI-Tools rule number:** 68
- **ACT Rules id:** qt1vmo
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 12.08.2020
- **Last updated in GitHub:** 17.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/qt1vmo

Description	This rule checks that the accessible names of images serve an equivalent purpose to the image.

Accessibility Requirements	1.1.1 Non-text Content (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	Any img, canvas or svg element that is visible and has a non-empty accessible name, except if one of the following is true: • The element has an ancestor in the flat tree that is named from author; or • The element is an img element where the current request's state is not completely available.
Expectation	Each test target has an accessible name that serves an equivalent purpose to the non-text content of that test target.
Assumptions	This rule assumes that the language of the accessible name of each test target can be correctly determined (either programmatically or by analyzing the content).
Accessibility Support	There are no major accessibility support issues known for this rule.

2 1.2.1 Audio-only and Video-only (Prerecorded) (Level A)

2.1 About Success Criteria 1.2.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#audio-only-and-video-only-prerecorded

Success Criteria	For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: • Prerecorded Audio-only: An alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content. • Prerecorded Video-only: Either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content.
Purpose	The intent of this Success Criteria is to make information conveyed by prerecorded audio-only and prerecorded video-only content available to all users. Alternatives for time-based media that are text based make information accessible because text can be rendered through any sensory modality (for example, visual, auditory or tactile) to match the needs of the user.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without hearing • Usage with limited hearing Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	This requirement relates to all audio-only video-only content, unless these are media alternatives to text and clearly marked as such. There is no requirement for the audio and/or video to be meaningful. The success criteria requires that an alternative for audio-only and video-only is provided. The alternative is in the form of text (for audio) or text or audio (for video). The alternative must be visually positioned close to the media, or it is possible to access via a mechanism. It must not be necessary to scroll to access the alternative/mechanism. The text alternative conveys the same information as the audio. A precise reproduction of the content is not required, but all essential content must be included in the correct order.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that
	• **c3232f:** For any non-streaming video element that is visible, where the video doesn't contain audio, at least one of the following is true:
	  ○ **fd26cf:** information contained in the video is available as text (directly or via text alternatives) that is visible and included in the accessibility tree. The video is labelled as a video alternative for text and this label is visible and included in the accessibility tree.
	  ○ **ac7dc6:** For video that is visible and does not contain audio and contains a track element with a kind="descriptions" attribute, the visual information in the video is described with a description (track element) in the same language as the video or the same language as the page.
	  ○ **ee13b5:** The visual information in the video is available through a text transcript that is available either on the

	page or through a link. The text transcript needs to be visible and included in the accessibility tree. ○ **d7ba54:** The visual information in the video is available through an audio track. • **e7a44:** For any non-streaming audio element that is either playing or has a play button that is visible and included in the accessibility tree, at least one of the following is true: ○ **2eb176:** The auditory information is available through a text transcript. The text transcript is visible and included in the accessibility tree, either on the page or through a link. ○ **afb423:** The auditory is available as text (directly or via text alternatives) that is visible and included in the accessibility tree. The audio is labelled as an audio alternative for text on the page and the label is visible and included in the accessibility tree.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • The text (for audio) or text or audio (for video) alternative (or a mechanism to access said alternative) conveys equivalent information for prerecorded audio-only and video-only content. • The text or audio alternative (or a mechanism to access said alternative) is positioned close to the media. It must not be necessary to scroll to access the alternative or mechanism. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

2.2 Test Rule: video element visual-only content has accessible alternative

- **WAI-Tools rule number:** 21
- **ACT Rules id:** c3232f
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Composite
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/c3232f

Description	This rule checks that video elements without audio have an alternative available.
Accessibility Requirements	1.2.1 Audio-only and Video-only (Prerecorded) (Level: A) • Required for conformance to WCAG 2.0 and above on level A and higher • Outcome mapping: ○ Any failed outcomes: success criterion is not satisfied. ○ All passed outcomes: success criterion needs further testing. ○ An inapplicable outcome: success criterion needs further testing.
Input rules	• Video Element Visual-Only Content Is Media Alternative for Text • Video Element Visual-Only Content Has Description Track • Video Element Visual-Only Content Has Transcript

	• Video Element Visual-Only Content Has Audio Track Alternative
Applicability	The rule applies to any non-streaming video element visible, where the video doesn't contain audio.
Expectation	For each test target, the outcome of at least one of the following rules is passed: • Video Element Visual-Only Content Is Media Alternative for Text • Video Element Visual-Only Content Has Description Track • Video Element Visual-Only Content Has Transcript • Video Element Visual-Only Content Has Audio Track Alternative
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	See Video Only Element Has Description Track: accessibility support.

2.2.1 Atomic rule: video element visual-only content is media alternative for text

- **ACT Rules id:** fd26cf
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 11.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/fd26cf

Description	This rule checks non-streaming silent video is a media alternative for text on the page.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual-only content has accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Language
Applicability	The rule applies to every non-streaming video element that is visible, where the video doesn't contain audio.
Expectation	**Expectation 1** All the information contained in each target element is available as text (directly or via text alternatives) that is visible and included in the accessibility tree. **Expectation 2** Each target element is labeled as a video alternative for text on the page. **Expectation 3** The label (from expectation 2) is visible and included in the accessibility tree. **Note:** The term label does not refer to the label element.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster.

	• This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

2.2.2 Atomic rule: video element visual-only content has description track

- **ACT Rules id:** ac7dc6
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/ac7dc6

Description	This rule checks that description tracks that come with non-streaming video elements, without audio, are descriptive.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual-only content has accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to every non-streaming video element that is visible where the video does not contain audio and contains a track element with a kind attribute value of descriptions.
Expectation	The visual information of each test target is described with a description track element that has the same language as the video or the same language as the page. **Note:** Multiple description track elements may be useful for different languages, but at least one must match the language of the video or the language of the page.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	Currently, the description track is not supported by most assistive technology. Video players may be able to work around the lack of support for the description track by using aria-live but few do this today.

2.2.3 Atomic rule: video element visual-only content has transcript

- **ACT Rules id:** ee13b5
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/ee13b5

Description	Non-streaming video elements without audio must have all visual information available in a transcript.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual-only content has accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to any non-streaming video element visible where the video doesn't contain audio.
Expectation	The visual information of each test target is available through a text transcript that is available either on the page or through a link. The text transcript needs to be visible and included in the accessibility tree. **Note:** A "text transcript" in the context of this rule is defined in WCAG 2 as an alternative for time based media.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

2.2.4 Atomic rule: video element visual-only content has audio track alternative

- **ACT Rules id:** d7ba54
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/d7ba54

Description	Non-streaming video elements without audio must have an audio alternative.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual-only content has accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to any non-streaming video element visible where the video does not contain audio.
Expectation	The visual information of each test target is available through an audio track.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.

Accessibility Support	There are no major accessibility support issues known for this rule.

2.3 Test Rule: audio element content has text alternative

- **WAI-Tools rule number:** 22
- **ACT Rules id:** e7aa44
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Composite
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/e7aa44

Description	This rule checks if audio only elements have a text alternative available.

Accessibility Requirements	1.2.1 Audio-only and Video-only (Prerecorded) (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input rules	• Audio element content has transcript • Audio element content is media alternative for text
Applicability	The rule applies to any non-streaming audio element that is: • playing; or, • has a "play button" that is visible and included in the accessibility tree. **Note:** A play button is an interactive element that when activated, plays the audio.
Expectation	For each test target, the outcome of at least one of the following rules is passed: • audio element content has transcript • audio element content is media alternative for text
Assumptions	This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

2.3.1 Atomic rule: audio element content has transcript

- **ACT Rules id:** 2eb176
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/2eb176

Description	Non-streaming audio elements must have a text alternative for all included auditory information.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Audio element content has text alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Language
Applicability	The rule applies to every non-streaming audio element that is: • playing; or, • has a "play button" that is visible and included in the accessibility tree. **Note:** A play button is an interactive element that when activated, plays the audio.
Expectation	The auditory information of each test target is available through a text transcript. That text transcript is visible and included in the accessibility tree, either on the page or through a link. **Note:** A "text transcript" in the context of this rule is defined in WCAG 2 as an alternative for time-based media.
Assumptions	This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

2.3.2 Atomic rule: audio element content is media alternative for text

- **ACT Rules id:** afb423
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/afb423

Description	This rule checks audio is a media alternative for text on the page.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Audio element content has text alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Language
Applicability	The rule applies to every non-streaming audio element that is: • playing; or, • has a "play button" that is visible and included in the accessibility tree. **Note:** A play button is an interactive element that when activated, plays the audio.
Expectation	**Expectation 1** The auditory information of each test target is available as text (directly or via text alternatives) that is visible and included in the accessibility tree.

	Expectation 2 Each target element is labeled as an audio alternative for text on the page. **Expectation 3** The label (from expectation 2) is visible and included in the accessibility tree. **Note:** The term label does not refer to the label element.
Assumptions	This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

3 1.2.2 Captions (Prerecorded) (Level A)

3.1 About Success Criteria 1.2.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#captions-prerecorded

Success Criteria	Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such.
Purpose	The intent of this Success Criteria is to enable people who are deaf or hard of hearing to watch synchronized media presentations. Captions provide the part of the content available via the audio track.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without hearing • Usage with limited hearing Secondary relationship • Usage with limited cognition

Digdir interpretation and specification of the success criteria	Prerecorded video with audio that is not a media alternative to text or video has an alternative in the form of captions or a text alternative
	Captions
	• not only include dialogue but identify who is speaking and include non-speech information conveyed through sound, including meaningful sound effects. Precise reproduction of the content is not required, but all essential content must be included in the correct order
	• are either permanent (open captions) or can be enabled (closed captions).
	• are visible, but do not obstruct important content in the video.
	Alternatively, the video can have a text alternative that
	• is visually positioned close to the video clip or can be accessed via a mechanism
	• is coded as text
	• conveys the content of audio and video
	The following are excepted from the requirement:
	• Prerecorded video with captions that is intended as a media alternative to text and is clearly marked as such.
	• For a media alternative to be clearly marked, the following requirements must be met:
	  ○ It is displayed in the direct vicinity of the text and there must be no doubt that it is a media alternative for the text in question.
	  ○ It comes before information that shows it is a media alternative, e.g. an icon indicating that it is a media

	player, or a link that says “listen to the text”, for example, and is linked to an adjacent text.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that
	• **eac66b:** For every non-streaming video element that is visible, where the video contains audio, at least one of the following is true:
	  ○ **ab4d13:** All the information contained in the video’s audio is available as text that is visible and included in the accessibility tree. Each video is labelled as a video alternative for text on the page. This label is visible and included in the accessibility tree.
	  ○ **f51b46:** Audio information that is not conveyed visually in the video, is available through captions. Captions can be either embedded in the video file itself or can be made available through a separate track. Captions convey not only the content of spoken dialogue, but also equivalents for non-dialogue audio information needed to understand the program content, including sound effects, music, laughter, speaker identification and location. Captions can be open or closed and should not obscure or obstruct relevant information in the video.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • The text, for which the video is an alternative, is visually positioned close to the video or can be accessed via a mechanism. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

3.2 Test Rule: video element auditory content has accessible alternative

- **WAI-Tools rule number:** 23
- **ACT Rules id:** eac66b
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Composite
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/eac66b

Description	This rule checks that video elements have an alternative for information conveyed through audio.

Accessibility Requirements	1.2.2 Captions (Prerecorded) (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.

	○ An inapplicable outcome: success criterion needs further testing.
Input rules	• Video element content is media alternative for text • Video element auditory content has captions
Applicability	The rule applies to every non-streaming video element that is visible, where the video contains audio.
Expectation	For each test target, the outcome of at least one of the following rules is passed: • Video Element Content Is Media Alternative For Text • Video Element Auditory Content Has Captions
Assumptions	• This rule assumes that the video element is used to play a video (for example, not only used to display an image), and that there is a mechanism to start the video. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

3.2.1 Atomic rule: Video element content is media alternative for text

- **ACT Rules id:** ab4d13
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 11.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/ab4d13

Description	This rule checks non-streaming video is a media alternative for text on the page.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element auditory content has accessible alternative • Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Language
Applicability	The rule applies to every non-streaming video element that is visible, where the video contains audio.
Expectation	**Expectation 1** All the information contained in each test target is available as text that is visible and included in the accessibility tree. **Expectation 2** Each test target is labelled as a video alternative for text on the page. This label is visible and included in the accessibility tree. **Note:** The term label does not refer to the label element. If several expectations, input all in same cell.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster.

	• This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

3.2.2 Atomic rule: Video element auditory content has captions

- **ACT Rules id:** f51b46
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/f51b46

Description	This rule checks that captions are available for audio information in non-streaming video elements.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element auditory content has accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output
Applicability	The rule applies to every non-streaming video element that is visible where the video contains audio.
Expectation	For each test target, audio information that is not conveyed visually in the video, is available through captions. **Note:** Captions can be either embedded in the video file itself or can be made available through a separate track.
Assumptions	This rule assumes that the video element is used to play a video (for example, not only used to display an image), and that there is a mechanism to start the video.
Accessibility Support	There are no major accessibility support issues known for this rule.

4 1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A)

4.1 About Success Criteria 1.2.3 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 15.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#audio-description-or-media-alternative-prerecorded

Success Criteria	An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such.
Purpose	The intent of this Success Criteria is to provide people who are blind or visually impaired access to the visual information in a synchronized media presentation. This Success Criteria describes two approaches, either of which can be used. One approach is to provide audio description of the video content. The audio description augments the audio portion of the presentation with the information needed when the video portion is not available. During existing pauses in dialogue, audio description provides information about actions, characters, scene changes, and on-screen text that are important and are not described or spoken in the main sound track. The second approach involves providing all of the information in the synchronized media (both visual and auditory) in text form. An alternative for time-based media provides a running description of all that is going on in the synchronized media content. The alternative for time-based media reads something like a screenplay or book. Unlike audio description, the description of the video portion is not constrained to just the pauses in the existing dialogue. Full descriptions are provided of all visual information, including visual context, actions and expressions of actors, and any other visual material. In addition, non-speech sounds (laughter, off-screen voices, etc.) are described, and transcripts of all dialogue are included. The sequence of description and dialogue transcripts are the same as the sequence in the synchronized media itself. As a result, the alternative for time-based media can provide a much more complete representation of the synchronized media content than audio description alone.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that
	• **c5a4ea:** For non-streaming video element that is visible, where the video contains audio, at least one of the following is true:
	  ○ **1ea59c:** The visual information in the video is available through its audio, or through an audio description track.
	  ○ **1a02b0:** A text transcript containing all the visual and auditory information in the video is available, either on the page or available through a link.
	  ○ **f196ce:** The visual information of the video, that is not available through its audio, is described with a description track element.
	  ○ **ab4d13:** All the information contained in the video's audio is available as text that is visible and included in the accessibility tree. Each video is labelled as a video alternative for text on the page. This label is visible and included in the accessibility tree.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

4.2 Test Rule: Video element visual content has accessible alternative

- **WAI-Tools rule number:** 24
- **ACT Rules id:** c5a4ea
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 26.08.2020
- **Rule Type:** Composite
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/c5a4ea

Description	This rule checks that video elements with audio have an alternative for the video content as audio or as text.

Accessibility Requirements	1.2.3 Audio Description or Media Alternative (Prerecorded) (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	1.2.5 Audio Description (Prerecorded) (Level AA)
	• Required for conformance to WCAG 2.0 and later on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

	1.2.8 Media Alternative (Prerecorded) (Level AAA)
	• Required for conformance to WCAG 2.0 and later on level AAA.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input rules	• Video element visual content has audio description • Video element visual content has transcript • Video element visual content has description track • Video element content is media alternative for text
Applicability	The rule applies to every non-streaming video element that is visible, where the video contains audio.
Expectation	For each test target, the outcome of at least one of the following rules is passed: • Video element visual content has audio description • Video element visual content has transcript • Video element visual content has description track • Video element content is media alternative for text
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	See Video element audio described: accessibility support. See Video element description track: accessibility support.

4.2.1 Atomic rule: Video element visual content has audio description

- **ACT Rules id:** 1ea59c
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/1ea59c

Description	This rule checks that non-streaming video elements have all visual information also contained in the audio.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to every non-streaming video element that is visible where the video contains audio.
Expectation	The visual information of each test target is available through its audio, or through an audio description track.

Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are only a few implementations of video players (without third party technologies) that support audio description tracks at the time of writing.

4.2.2 Atomic rule: Video element visual content has transcript

- **ACT Rules id:** 1a02b0
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/1a02b0

Description	This rule checks that non-streaming video elements have all audio and visual information available in a transcript.

Accessibility Requirements	1.2.8 Media Alternative (Prerecorded) (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: further testing is needed

Used in Rules	• Video element visual content has accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to every non-streaming video element that is visible where the video contains audio.
Expectation	A text transcript containing all the visual and auditory information of the test target is available, either on the page or available through a link. **Note:** A "text transcript" in the context of this rule is defined in WCAG 2 as an alternative for time-based media.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

4.2.3 Atomic rule: Video element visual content has description track

- **ACT Rules id:** f196ce
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 07.07.2020

- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/f196ce

Description	This rule checks that description tracks that come with non-streaming video elements are descriptive.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to every non-streaming video element that is visible where the video contains audio and a track element with a kind attribute value of descriptions.
Expectation	The visual information of each test target not available through its audio is described with a description track element. **Note:** Multiple description track elements may be useful for different languages, but at least one must match the language of the video.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	Currently, the description track is not supported by most assistive technology. Accessibility support for the description track attribute is relatively low to non-existent. Video players may be able to work around the lack of support for the description track by using aria-live, but few do this today. This means that the rule can only provide a pass for these success criteria if assistive technology supports the description track or if the video player that is used has implemented such a workaround.

4.2.4 Atomic rule: Video element content is media alternative for text

- **ACT Rules id:** ab4d13
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 11.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/ab4d13

Description	This rule checks non-streaming video is a media alternative for text on the page.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element auditory content has accessible alternative • Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Language

Applicability	The rule applies to every non-streaming video element that is visible, where the video contains audio.
Expectation	**Expectation 1** All the information contained in each test target is available as text that is visible and included in the accessibility tree. **Expectation 2** Each test target is labelled as a video alternative for text on the page. This label is visible and included in the accessibility tree. **Note:** The term label does not refer to the label element.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

5 1.2.4 Captions (Live) (Level AA)

5.1 About Success Criteria 1.2.4 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#captions-live

Success Criteria	Captions are provided for all live audio content in synchronized media.
Purpose	The intent of this Success Criteria is to enable people who are deaf or hard of hearing to watch real-time presentations. Captions provide the part of the content available via the audio track. Captions not only include dialogue, but also identify who is speaking and notate sound effects and other significant audio. This success criteria was intended to apply to broadcast of synchronized media and is not intended to require that two-way multimedia calls between two or more individuals through web apps must be captioned regardless of the needs of users. Responsibility for providing captions would fall to the content providers (the callers) or the "host" caller, and not the application.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without hearing • Usage with limited hearing Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

6 1.2.5 Audio Description (Prerecorded) (Level AA)

6.1 About Success Criteria 1.2.5 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 15.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#audio-description-prerecorded

Success Criteria	Audio description is provided for all prerecorded video content in synchronized media.
Purpose	The intent of this Success Criteria is to provide people who are blind or visually impaired access to the visual information in a synchronized media presentation. The audio description augments the audio portion of the presentation with the information needed when the video portion is not available. During existing pauses in dialogue, audio description provides information about actions, characters, scene changes, and on-screen text that are important and are not described or spoken in the main sound track.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that
	• **c5a4ea:** For every non-streaming video element that is visible, where the video contains audio, at least one of the following is true:
	  ○ **1ea59c:** The visual information in the video is available through its audio, or through an audio description track.
	  ○ **ab4d13:** All the information contained in the video's audio is available as text that is visible and included in the accessibility tree. Each video is labelled as a video alternative for text on the page. This label is visible and included in the accessibility tree.
	  ○ **f196ce:** The visual information of the video, that is not available through its audio, is described with a description track element.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None.

	Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

6.2 Test Rule: Video element visual content has strict accessible alternative

- **WAI-Tools rule number:** 25
- **ACT Rules id:** 1ec09b
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Composite
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/1ec09b

Description	This rule checks that video elements with audio have audio description.

Accessibility Requirements	1.2.5 Audio Description (Prerecorded) (Level: AA)
	• Required for conformance to WCAG 2.0 and later on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input rules	• Video element visual content has audio description • Video element content is media alternative for text • Video element visual content has description track
Applicability	The rule applies to every non-streaming video element that is visible, where the video contains audio.
Expectation	For each test target, the outcome of at least one of the following rules is passed: • Video element visual content has audio description • Video element content is media alternative for text • Video element visual content has description track
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	See Video element audio described: accessibility support. See Video element description track: accessibility support.

6.2.1 Atomic rule: Video element visual content has audio description

- **ACT Rules id:** 1ea59c
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 20.03.2020
- **Rule Type:** Atomic (belongs to composite rule)

- **URL:** https://act-rules.github.io/rules/1ea59c

Description	This rule checks that non-streaming video elements have all visual information also contained in the audio.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output, Language
Applicability	The rule applies to every non-streaming video element that is visible where the video contains audio.
Expectation	The visual information of each test target is available through its audio, or through an audio description track.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are only a few implementations of video players (without third party technologies) that support audio description tracks at the time of writing.

6.2.2 Atomic rule: Video element content is media alternative for text

- **ACT Rules id:** ab4d13
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 11.03.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/ab4d13

Description	This rule checks non-streaming video is a media alternative for text on the page.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element auditory content has accessible alternative • Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Language
Applicability	The rule applies to every non-streaming video element that is visible, where the video contains audio.
Expectation	**Expectation 1** All the information contained in each test target is available as text that is visible and included in the accessibility tree. **Expectation 2**

	Each test target is labelled as a video alternative for text on the page. This label is visible and included in the accessibility tree. **Note:** The term label does not refer to the label element.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	There are no major accessibility support issues known for this rule.

6.2.3 Atomic rule: Video element visual content has description track

- **ACT Rules id:** f196ce
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/f196ce

Description	This rule checks that description tracks that come with non-streaming video elements are descriptive.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Video element visual content has accessible alternative • Video element visual content has strict accessible alternative
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output
Applicability	The rule applies to every non-streaming video element that is visible where the video contains audio and a track element with a kind attribute value of descriptions.
Expectation	The visual information of each test target not available through its audio is described with a description track element. **Note:** Multiple description track elements may be useful for different languages, but at least one must match the language of the video.
Assumptions	• This rule assumes that a mechanism is available to start the video and that the video element is not simply used to display the poster. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	Currently the description track is not supported by most assistive technology. Accessibility support for the description track attribute is relatively low to non-existent. Video players may be able to work around the lack of support for the description track by using aria-live but few do this today. This means that the rule can only provide a pass for these success criteria if assistive technology support the description track or if the video player that is used has implemented such a work around.

7 1.3.1 Info and Relationships (Level A)

7.1 About Success Criteria 1.3.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#info-and-relationships

Success Criteria	Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.
Purpose	The intent of this Success Criteria is to ensure that information and relationships that are implied by visual or auditory formatting are preserved when the presentation format changes. For example, the presentation format changes when the content is read by a screen reader or when a user style sheet is substituted for the style sheet provided by the author.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	This success criteria applies to all content types, structural components and relationships. In the operationalisation of the requirement, the Authority has chosen to incorporate the following content in the test procedures: Headings, lists and bullet points, tables, form elements and groups of form elements. In some cases, it will not be technically possible to display information or relationships in other formats. In such instances, information, structures and relationships must be explained in the form of text. The understanding article for this success criteria nevertheless emphasises that if it is technically possible to determine the info and relationships programmatically, this is recommended rather than making the information available as text. For visual headings: • Visual headings are coded as headings • Headings that are included in a visual heading hierarchy are coded with the correct heading level For tables, the following content in tables is coded semantically correctly: • Table caption. • Header cells in tables with headings in just one row or column. • Header cells in tables with headings in both a column and a row. • Header cells in tables with headings on several rows and/or columns inside the table. Visual lists are coded as follows:

	• Ordered lists are coded as ordered lists. • Unordered lists are coded as unordered lists. • Description lists (lists that have two levels and provide supplementary explanations) are coded as description lists. Content types, structural components and relationships that are currently not covered in the test procedures are paragraphs, quotations and references, superscript and subscript, tables used for layout, and WAI-ARIA regions and landmarks.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **6cfa84:** Elements with an aria-hidden="true" attribute are not part of sequential focus navigation, nor do they have descendants in the flat tree that are part of sequential focus navigation. • **ff89c9:** HTML or SVG element that is included in the accessibility tree and has a WAI-ARIA 1.1 explicit semantic role with a WAI-ARIA required context role, except if the element has an implicit semantic role that is identical to its explicit semantic role. Each test target is owned by an element that has a semantic role that is one of the WAI-ARIA required context roles of the target element. • **bc4a75:** HTML or SVG element that is included in the accessibility tree and has a WAI-ARIA 1.1 explicit sematic role with required owned elements only owns elements with a sematic role from the required owned element list for the test target's semantic role except if it has: an implicit semantic role that is identical to its explicit semantic role; or a semantic role of combobox; or the aria-busy attribute value of true, or has an ancestor in the accessibility tree with this attribute value. • **ffd0e9:** HTML element with the semantic role of heading that is included in the accessibility tree, has a non-empty ("") accessible name. • **a25f45:** Headers attribute specified on a cell within a table element where the table element is visible and included in the accessibility tree: is a set of space separated IDs; each of which is the ID of an element, that is a cell of the same table and none of which is the ID of the element that is referring to its own ID. • **d0f69e:** HTML element that has the semantic role of rowheader or columnheader and; is visible and; is included in the accessibility tree and; has at least one ancestor in the flat tree with a semantic role of either table or grid; and whose closest ancestor in the flat tree with a semantic role of either table or grid is included in the accessibility tree: is assigned to at least one element with a semantic role of cell or gridcell.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Visual headings are marked up as headings. • Headings that are included in a visual heading hierarchy are marked up with the corresponding heading level. • Table caption and header cells are semantically marked up correctly. • Lists are semantically marked up as unordered lists, ordered lists and description lists. **Note:** This success criteria covers many possible situations. This is not a complete list of possible ways to comply with the requirement.

	Note: There are several test rules currently in development. This will increase coverage for this success criteria. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

7.2 Test Rule: Element with aria-hidden has no focusable content

- **WAI-Tools rule number:** 15
- **ACT Rules id:** 6cfa84
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 21.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/6cfa84

Description	This rule checks that elements with an aria-hidden attribute do not contain focusable elements.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	
	4.1.2 Name, Role, Value (Level: A)
	
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any element with an aria-hidden attribute value of true. **Note:** Using aria-hidden="false" on a descendant of an element with aria-hidden="true" does not expose that element. aria-hidden="true" hides itself and all its content from assistive technologies.
Expectation	None of the target elements are part of sequential focus navigation, nor do they have descendants in the flat tree that are part of sequential focus navigation.
Assumptions	There are currently no assumptions
Accessibility Support	Some user agents treat the value of aria-hidden attribute as case-sensitive.

7.3 Test Rule: ARIA required context role

- **WAI-Tools rule number:** 32
- **ACT Rules id:** ff89c9

- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 10.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/ff89c9

Description	This rule checks that an element with an explicit semantic role exists inside its required context.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, CSS styling, DOM Tree
Applicability	The rule applies to any HTML or SVG element that is included in the accessibility tree and has a WAI-ARIA 1.1 explicit semantic role with a required context role, except if the element has an implicit semantic role that is identical to its explicit semantic role.
Expectation	Each test target is owned by an element that has a semantic role that is one of the required context roles of the target element.
Assumptions	This rule assumes that the role attribute is used to give a semantic role to the element according to ARIA specifications. If it is used for other purposes, and relationships between elements are already programmatically determinable by other means, it is possible to fail this rule but still satisfy WCAG success criterion 1.3.1 Info and Relationships.
Accessibility Support	• User agents do not all have the same accessibility tree. Particularly the method of deriving which element owns which other elements varies between browsers. This can lead to different results for this rule, depending on which accessibility tree is used as input. • aria-owns has limited support in some user agents. • There exist some combination of popular browsers and assistive technologies who do not announce correctly relationships based on a mix of implicit and explicit roles.

7.4 Test Rule: ARIA required owned elements

- **WAI-Tools rule number:** 33
- **ACT Rules id:** bc4a75
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 21.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/bc4a75

Description	This rule checks that an element with an explicit semantic role has at least one of its required owned elements.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, CSS Styling, DOM Tree
Applicability	The rule applies to any HTML or SVG element that is included in the accessibility tree and has a WAI-ARIA 1.1 explicit semantic role with required owned elements, except if • the element has an implicit semantic role that is identical to its explicit semantic role; or • the element has a semantic role of combobox; or • the element has the aria-busy attribute value of true or has an ancestor in the accessibility tree with this attribute value.
Expectation	Each test target only owns elements with a semantic role from the required owned element list for the test target's semantic role. **Note:** The definition of owned by used in this rule is different than the definition of "owned element" in WAI-ARIA. See more in the owned by definition.
Assumptions	If the explicit semantic role on the target element is incorrectly used, and any relationships between elements are already programmatically determinable, failing this rule may not result in accessibility issues for users of assistive technologies, and it should then not be considered a failure under WCAG success criterion 1.3.1 Info and Relationships.
Accessibility Support	User agents do not all have the same accessibility tree. Particularly the method of deriving which element owns which other elements varies between browsers. This can lead to different results for this rule, depending on which accessibility tree is used as input. aria-owns has limited support in some user agents. Assistive technologies are not consistent in how they handle situations where a required owned element has a missing or incorrect role. This can lead to situations where inaccurate owned elements behave as expected in one assistive technology, but not in another. Some user agents treat the value of aria-busy as case-sensitive.

7.5 Test Rule: Heading has non-empty accessible name

- **WAI-Tools rule number:** 34
- **ACT Rules id:** ffd0e9
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 21.08.2020
- **Last updated in GitHub:** 29.06.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto

- **URL:** https://act-rules.github.io/rules/ffd0e9

Description	This rule checks that each heading has a non-empty accessible name.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.6 Headings and Labels (Level: AA)
	• Required for conformance to WCAG 2.0 and later on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility, DOM Tree, CSS Styling
Applicability	This rule applies to any HTML element with the semantic role of heading that is included in the accessibility tree.
Expectation	Each test target has a non-empty ("") accessible name.
Assumptions	There are currently no assumptions.
Accessibility Support	Some assistive technologies may hide headings with empty accessible name from the users. This depends both on the user agent and how the accessible name was computed (the accessible name and description computation is not clear concerning which characters should be trimmed) and of the assistive technology itself. Hence, there are cases where the outcome of this rule is failed, but users of certain assistive technology and browser combinations will not experience an issue. **Note:** Completely empty headings ( ) seem to be consistently ignored by assistive technologies. However, they fail Technique H42: Using h1-h6 to identify headings (by using heading markup for content which is not heading). Moreover, they may be rendered on screen (by breaking flow content, or because of custom styling), thus causing concerns for sighted users. Therefore, this rule also fails on these. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of heading and fail this rule with some technology but

	users of other technologies would not experience any accessibility issue. The accessible name and description computation suggest that if an aria-labelledby attribute refers to an existing but empty element, the computation should stop and return an empty name without defaulting to the next steps. Several user agents and assistive technologies chose to use the next step in the computation in this case.

7.6 Test Rule: Headers attribute specified on a cell refers to cells in the same table element

- **WAI-Tools rule number:** 44
- **ACT Rules id:** a25f45
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 29.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/a25f45

Description	This rule checks that the headers attribute on a cell refer to other cells in the same table element.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	This rule applies to any headers attribute specified on a cell within a table element, where the table element is visible and included in the accessibility tree.
Expectation	**Expectation 1** Each target attribute is a set of space separated IDs, each of which is the ID of an element, that is a cell of the same table. Note: headers attribute referencing elements that are non-existent or not in the table are ignored when assigning header cells (step 3, first case, point 2). **Expectation 2** Each target attribute is a set of space separated IDs, none of which is the ID of the element on which the test target is specified. **Note:** headers attribute referencing to the cell itself are ignored when assigning header cells (step 3, first case, point 2).
Assumptions	This rule assumes that the headers attribute is only used to identify table headers. If other information is included in the headers attribute, the rule may fail on issues

	that are not accessibility concerns. For example, if headers is used to include information for scripts, this rule may not be accurate. This rule assumes that the headers attribute is required to express the relationship between data and header cells in the table. If the browser computes an adequate fallback header, this rule may produce incorrect results.
Accessibility Support	There are no major accessibility support issues known for this rule.

7.7 Test Rule: Table header cell has assigned data cells

- **WAI-Tools rule number:** 45
- **ACT Rules id:** d0f69e
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 23.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/d0f69e

Description	This rule checks that each table header has assigned data cells in a table element.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, CSS styling, DOM Tree
Applicability	The rule applies to any HTML element with the semantic role of rowheader or columnheader for which all of the following is true: • the element is visible; and • the element is included in the accessibility tree; and • the element has at least one ancestor in the flat tree with a semantic role of either table or grid; and • the element's closest ancestor in the flat tree with a semantic role of either table or grid is included in the accessibility tree.
Expectation	Each target element is assigned to at least one element with a semantic role of cell or gridcell.
Assumptions	This rule assumes that table header cells have a relationship conveyed through presentation with other cells within the same table. This excludes edge cases such as a table definition where there is only one header cell, or a table definition where there are multiple headers and no other cells. In such scenarios the rule fails, but success criterion 1.3.1 Info and Relationships could still be satisfied.

Accessibility Support	Table markup and header cell association is not well supported by some popular assistive technologies. Passing this rule can still cause issues for users of those assistive technologies. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have one of the applicable semantic roles and fail this rule with some technology, but users of other technologies would not experience any accessibility issue.

8 1.3.2 Meaningful Sequence (Level A)

8.1 About Success Criteria 1.3.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#meaningful-sequence

Success Criteria	When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.
Purpose	The intent of this Success Criteria is to enable a user agent to provide an alternative presentation of content while preserving the reading order needed to understand the meaning. It is important that it be possible to programmatically determine at least one sequence of the content that makes sense. Content that does not meet this Success Criteria may confuse or disorient users when assistive technology reads the content in the wrong order, or when alternate style sheets or other formatting changes are applied. Screen readers and other assistive technology must be able to present content in the same meaningful sequence as the visual presentation on a website. Content that is not presented in a logical reading order may confuse the user and render the content inaccessible. A sequence is meaningful if the order of content in the sequence cannot be changed without affecting its meaning. For example, if a page contains two independent articles, the relative order of the articles may not affect their meaning, as long as they are not interleaved. In such a situation, the articles themselves may have meaningful sequence, but the container that contains the articles may not have a meaningful sequence. The semantics of some elements define whether or not their content is a meaningful sequence. For instance, in HTML, text is always a meaningful sequence. Tables and ordered lists are meaningful sequences, but unordered lists are not.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The same meaningful sequence in which the content is presented on a page must be observed when presented by a screen reader. The understanding article for the success criteria indicates that a sequence is meaningful if the order of the content in the sequence cannot be changed without affecting its meaning. The semantics of some elements define whether or not their content is a meaningful sequence. Examples of meaningful sequences include • content coded as text

	• content coded as a table • content coded as an ordered list Unordered lists are not deemed to be a meaningful sequence. The understanding article otherwise specifies the following: • Providing a particular linear order is only required where it affects meaning. • There may be more than one order that meets the requirement to present content in a meaningful order. • Only one correct order needs to be provided. The reading order of content in a display with CSS disabled, compared with standard display, is either the same or a reading order that otherwise presents the same meaning of the content.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

9 1.3.3 Sensory Characteristics (Level A)

9.1 About Success Criteria 1.3.3 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#sensory-characteristics

Success Criteria	Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, color, size, visual location, orientation, or sound.
Purpose	The intent of this Success Criteria is to ensure that all users can access instructions for using the content, even when they cannot perceive shape or size or use information about spatial location or orientation. Some content relies on knowledge of the shape or position of objects that are not available from the structure of the content (for example, "round button" or "button to the right"). Some users with disabilities are not able to perceive shape or position due to the nature of the assistive technologies they use. This Success Criteria requires that additional information be provided to clarify anything that is dependent on this kind of information. Some assistive technologies do not perceive the shape or position of content components, e.g. “round” or “on the left”. This success criteria requires that more information be given in order to clarify this type of information. Providing information by means of shape, position or similar is nevertheless an effective method for many users, including individuals with cognitive limitations. It is therefore possible to refer to sensory characteristics such as “the round button”, but more information must also be provided. In some languages, it is commonly understood that "above" refers to the content previous to that point in the content and "below" refers to the content after that point. In such languages, if the content being referenced is in the appropriate place in the reading order and the references are unambiguous, statements such as "choose one of the links below" or "all of the above" would conform to this Success Criteria.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without perception of color • Usage without hearing • Usage with limited hearing Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The success criteria applies to instructions for using content and requirements for the provision of more information than reference to sensory characteristics. The understanding article for the success criteria indicates that the instructions may contain words that refer to the reading order if the same reading order is observed in the code.

	The instruction is provided in the form of a symbol, icon or graphic representation, and this is identified in the code Clarifications: • Color: A note on the success criteria indicates that the use of color is linked to success criteria 1.4.1, which specifies that color must not be the only visual means of conveying information. • Image: Instructions that are images (e.g. form instructions) come under success criteria 1.1.1 and are verified under this success criteria. • Running text: We do not verify instances where instructions on use that refer to the components' sensory characteristics are specified in running text. This would be highly resource intensive, with a risk of random testing and uneven test result quality.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

10 1.3.4 Orientation (Level AA)

10.1 About Success Criteria 1.3.4 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 22.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#orientation

Success Criteria	Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless a specific display orientation is essential.
Purpose	The intent of this Success Criteria is to ensure that content displays in the orientation (portrait or landscape) preferred by the user. Some websites and applications automatically set and restrict the screen to a particular display orientation and expect that users will respond by rotating their device to match, but this can create problems. Some users have their devices mounted in a fixed orientation (e.g. on the arm of a power wheelchair). Therefore, websites and applications need to support both orientations by not restricting the orientation. Changes in content or functionality due to the size of display are not covered by this criteria which is focused on restrictions of orientation. The goal of this Success Criteria is that authors should never restrict content's orientation, thus ensuring that it always match the device display orientation. It is important to distinguish between an author's responsibility not to restrict content to a specific orientation, and device-specific settings governing the locking of display orientation. Many hand-held devices offer a mechanical switch or a system setting (or both) to allow the user to lock the device display to a specific orientation. Where a user decides to lock their entire device to an orientation, all applications are expected to pick up that setting and to display content accordingly. This Success Criteria complements device "lock orientation" settings. A web page that does not restrict its display orientation will always support the system-level orientation setting, since the system setting is picked up by the user agent. Alternatively, where a device-level orientation lock is not in place, the user agent should display the page according to the orientation of the device (either its default, or the current orientation determined by any device sensors). The exception for things considered essential is aimed at situations where the content would only be understood in a particular orientation, or where the technology restricts the possible orientations. If content is aimed at a specific environment which is only available in one orientation (such as a television) then the content can restrict the orientation. Technologies such as virtual reality use screens within goggles that cannot change orientation relative to the user's eyes.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited manipulation or strength Secondary relationship • Usage of limited cognition

Digdir interpretation and specification of the success criteria	None. Not a part of current legislation
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **b33eff:** Any HTML element that is visible and has a CSS transform property that is applied conditionally on the orientation media feature with a value of landscape or portrait, where the CSS transform property has any of the transformation functions that is rotate, rotate3d, rotate, matrix, matrix3d. Then the target element is neither rotated clockwise nor counter clockwise around the Z-axis at an corresponding to 90 degree relative from the position of the element in landscape orientation to the position of the element in portrait orientation, and vice versa.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None Aspects of the ACT rules that go beyond the scope of the success criteria: • None

10.2 Test Rule: Orientation of the page is not restricted using CSS transform property

- **WAI-Tools rule number:** 42
- **ACT Rules id:** b33eff
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/b33eff

Description	This rule checks that page content is not restricted to either landscape or portrait orientation using CSS transform property.

Accessibility Requirements	1.3.4 Orientation (Level: AA)
	• Required for conformance to WCAG 2.1 on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree - CSS Styling
Applicability	The rule applies to any HTML element that is visible and has a CSS transform property that is applied conditionally on the orientation media feature with a value of landscape or portrait, where the CSS transform property has any of the below transformation functions: • rotate

	• rotate3d • rotateZ • matrix • matrix3d **Note:** These specific transformation functions are of interest to this rule as they have the potential to affect the rotation of a given element. **Note:** The rotate3d, rotateZ and matrix3d are currently part of a W3C Editor's Draft.
Expectation	The target element is neither rotated clockwise nor counter clockwise around the Z-axis at an angle corresponding to 90 degrees relative from the position of the element in landscape orientation to the position of the element in portrait orientation, and vice versa. **Note:** Imagine the display of a smartphone with an upwards pointing arrow at its center. If a user turns the smartphone a quarter turn, that is a move from one orientation to the other, they would want the arrow to continue pointing upwards. The smartphone accomplishes this by rotating the contents of its display a quarter turn to counter the users change in orientation. In effect, the arrow has remained in place and its rotation relative from one orientation to the other is 0 degrees. Now imagine that a developer rotates the arrow by a quarter turn only when in one orientation and not the other; its rotation relative from one orientation to the other would then be 90 degrees and it would appear stuck, or locked, as the user moves between orientations. What the developer has done is effectively counter the smartphones attempt at countering the users change in orientation.
Assumptions	This rule does not consider and may produce incorrect results for: • Elements for which a particular display orientation is essential. • The existence of any control on the page that can change the orientation on demand.
Accessibility Support	There are no major accessibility support issues known for this rule.

11 1.3.5 Identify Input Purpose (Level AA)

11.1 About Success Criteria 1.3.5 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#identify-input-purpose

Success Criteria	The purpose of each input field collecting information about the user can be programmatically determined when: • The input field serves a purpose identified in the Input Purposes for User Interface Components section; and • The content is implemented using technologies with support for identifying the expected meaning for form input data.
Purpose	The intent of this Success Criteria is to ensure that the purpose of a form input collecting information about the user can be programmatically determined, so that user agents can extract and present this purpose to users using different modalities.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **73f2c2:** The HTML autocomplete attribute has a correct value. Input, select and textarea elements with an autocomplete attribute contain a single term or a list of terms, follow the HTML specification, and the autocomplete term is a proper term for the element.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

11.2 Test Rule: autocomplete attribute has valid value

- **WAI-Tools rule number:** 2
- **ACT Rules id:** 73f2c2
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 21.08.2020
- **Last updated in GitHub:** 07.07.2020

- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/73f2c2

Description	This rule checks that the HTML autocomplete attribute has a correct value

Accessibility Requirements	1.3.5 Identify Input Purpose (Level: AA)
	• Required for conformance to WCAG 2.1 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling
Applicability	The rule applies to any HTML input, select and textarea element with an autocomplete attribute that is a set of one or more space separated tokens, except if one of the following is true: • The element is not visible, and not included in the accessibility tree • The element is an input element with a type property of hidden, button, submit or reset • The element has an aria-disabled="true" attribute • The element is not part of sequential focus navigation and has a semantic role that is not a widget role.
Expectation	**Expectation 1** The autocomplete attribute is a single term, or a space separated list of terms. **Expectation 2** The autocomplete term(s) follow the HTML 5.2 specification, which requires that it/they match the following in the correct order: • Has a value that starts with "section-" (optional) • Has either "shipping" or "billing" (optional) • Has either "home", "work", "mobile", "fax" or "pager" (optional, only for "email", "impp", "tel" or "tel-*") • Has a correct autocomplete field (required) **Note:** Autocomplete terms are case insensitive. When multiple terms are used, they must be used in the correct order. **Expectation 3** The correct autocomplete field is an appropriate field for the form control.
Assumptions	For this rule, it is assumed that the autocomplete attribute is not used on form fields that do not correspond to a autocomplete field described in the HTML 5.2 specification. If the autocomplete field is used to describe "custom" taxonomy, rather than that described in the specification, this rule may produce incorrect results.
Accessibility Support	While autocomplete in a promising technique for supporting personalization in HTML, support for this is fairly limited. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of none and fail this rule with some technology, but users of other technologies would not experience any accessibility issue.

	Some user agents treat the value of the aria-disabled attribute as case-sensitive.

12 1.4.1 Use of Color (Level A)

12.1 About Success Criteria 1.4.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 22.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#use-of-color

Success Criteria	Colour is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
Purpose	The intent of this Success Criteria is to ensure that all users can access information that is conveyed by colour differences, that is, by the use of colour where each colour has a meaning assigned to it. If the information is conveyed through colour differences in an image (or other non-text format), the colour may not be seen by users with colour deficiencies. In this case, providing the information conveyed with colour through another visual means ensures users who cannot see colour can still perceive the information. Colour is an important asset in design of Web content, enhancing its aesthetic appeal, its usability, and its accessibility. However, some users have difficulty perceiving colour. People with partial sight often experience limited colour vision, and many older users do not see colour well. In addition, people using text-only, limited-colour or monochrome displays and browsers will be unable to access information that is presented only in colour. Information that is conveyed using color or colour differences must be accessible to all. Colour must therefore not be the only visual means used to convey information, refer to an action, ask for a response or distinguish a visual component. Colour must be used in combination with other means such as text or a visual marker other than colour.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without perception of colour Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The Authority assumes that the success criteria applies only if colour is used as a means for conveying information, and not if colour is used for purely decorative or aesthetic purposes. Use of colours or colour coding is not advised against as long as other visual expressions are also used to convey the information. The success criteria applies to all content types. The Authority's operationalisation of the requirement in test procedures incorporates link text, information and graphic representations, form elements and error messages. The test procedure tests links in running text. Link testing does not cover links in menus and links that are grouped together and not visually positioned next to non-link content.

	Links that are marked with colour are distinguished from adjacent non-link text by means other than colour alone, e.g. font type, font size, underlining, bold type, frames or icons. Links that are marked with colour alone have a contrast of at least 3:1 to adjacent non-link text and have visible marking by means other than colour alone in the case of both mouseover and keyboard focus. Changing the cursor and standard focus indicator when navigating using a keyboard are not sufficient marking. If colour is used as a visual instrument for conveying information or distinguishing visual components in graphic representations, one of the following requirements is also met: • Explanatory text or other visual instruments that provide the same information conveyed by use of colour. Example: pattern, shape (e.g. star, diamond, square, etc.) or a direct link between part of the illustration where colour is used and explanatory information. • There is a mechanism that activates such information. Form elements, instructions and error messages that use colour to: • convey information • show an action • request a response • or distinguish a visual component are marked by means other than colour alone and the marking is visual.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

13 1.4.2 Audio Control (Level A)

13.1 About Success Criteria 1.4.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 22.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#audio-control

Success Criteria	If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level.
Purpose	Individuals who use screen reading software can find it hard to hear the speech output if there is other audio playing at the same time. This difficulty is exacerbated when the screen reader's speech output is software based (as most are today) and is controlled via the same volume control as the sound. Therefore, it is important that the user be able to turn off the background sound. Note: Having control of the volume includes being able to reduce its volume to zero. Playing audio automatically when landing on a page may affect a screen reader user's ability to find the mechanism to stop it because they navigate by listening and automatically started sounds might interfere with that navigation. Therefore, we discourage the practice of automatically starting sounds (especially if they last more than 3 seconds), and encourage that the sound be started by an action initiated by the user after they reach the page, rather than requiring that the sound be stopped by an action of the user after they land on the page.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited hearing Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The success criteria requires a mechanism for audio control when audio started automatically lasts for more than three seconds. The mechanism for audio control must be readily accessible, and it must be possible to control the audio on the website. The success criteria indicates that the general volume control in the operating system must not be used to control the audio. Technique G 170 linked to the success criteria indicates that a mechanism is visually positioned close to the start of the page when it is not necessary to scroll away from the start of the page in order to access it. Mechanism can be operated with a keyboard, is early in tab order sequence (maximum 5 tab steps), and has visual marker in the form of text or a symbol.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that: • **80f0bf:** HTML audio or video element that has an autoplay attribute value of true; and both paused and muted attributes

	value of false; and either a src attribute or a child source element that references content with a duration of more than 3 seconds that contains audio. For of these elements, the outcome of at least one of the following rules is passed: ○ **4c31df:** a mechanism is provided to pause or stop the audio or turn the audio volume off independently from the overall system volume control. The mechanism to pause or stop or turn the audio volume off is visible, has an accessible name that is not only whitespace, and is included in the accessibility tree. ○ **aaa1bf:** the total audio output does not last more than 3 seconds.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WA I-Tools: • The mechanism to pause or stop or turn the audio volume off is visually positioned near the start of the page. • The mechanism to pause or stop or turn the audio volume off can be operated with a keyboard. • The mechanism to pause or stop or turn the audio volume off is early in the tab order sequence (maximum 5-tab steps). This is not a comprehensive list of all aspects that may be covered by this success criteria. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

13.2 Test Rule: audio or video avoids automatically playing audio

- **WAI-Tools rule number:** 41
- **ACT Rules id:** 80f0bf
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 01.10.2020
- **Rule Type:** Composite
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/80f0bf

Description	This rule checks that audio or video that plays automatically does not have audio that lasts for more than 3 seconds or has an audio control mechanism to stop or mute it.
Accessibility Requirements	1.4.2 Audio Control (Level: A) • Required for conformance to WCAG 2.0 and later on level A and higher. • Outcome mapping: ○ Any failed outcomes: success criterion is not satisfied. ○ All passed outcomes: success criterion needs further testing. ○ An inapplicable outcome: success criterion needs further testing.

Input rules (composite only)	• audio or video that plays automatically has a control mechanism • audio or video that plays automatically has no audio that lasts more than 3 seconds
Applicability	This rule applies to any audio or video element for which all the following are true: • (**autoplay**) the element has an autoplay attribute value of true; and • (**not muted**) the element has a muted attribute value of false; and • (**not paused**) the element has a paused attribute value of false; and • (**duration**) the element has a media resource lasting more than 3 seconds and that contains audio.
Expectation	For each test target, the outcome of at least one of the following rules is passed: • audio or video that plays automatically has a control mechanism • audio or video that plays automatically has no audio that lasts more than 3
Assumptions	• This rule assumes that it is not possible to satisfy Success Criterion 1.4.2 Audio Control if the total length of the automatically playing audio is more than 3 seconds, even if there are pauses in the sound and no more than 3 seconds in a row with actual sound. • This rule assumes that the mechanism to control the sound must be located in the same web page. Note that mechanisms that are provided by user agents, assistive technologies, or operating systems are hardly testable as they depend on the user. Note that mechanisms that are located on other pages can still create accessibility issues for users relying on sound to navigate (e.g. screen readers users) since the autoplaying sound will interfere with their ability to find and activate the mechanism. If a mechanism external to the web page is provided, it is possible to fail this rule but still satisfy Success Criterion 1.4.2 Audio Control. • This rule assumes that the mechanism to control the sound must be visible and accessible in order to be effective and usable by all kinds of users. If the mechanism is hidden to some users, it is possible to fail this rule but still satisfy Success Criterion 1.4.2 Audio Control.
Accessibility Support	The native video and audio controls in several browser and assistive technology combinations are not keyboard accessible and the video or audio element itself may not be announced. Authors are recommended to use custom controls for keyboard navigation and cross browser accessibility support in general.

13.2.1 Atomic rule: audio or video that plays automatically has a control mechanism

- **ACT Rules id:** 4c31df
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **URL:** https://act-rules.github.io/rules/4c31df

Description	audio or video that plays automatically must have a control mechanism.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• audio or video avoids automatically playing audio

Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output
Applicability	This rule applies to any audio or video element that has: • an autoplay attribute value of true, and • a muted attribute value of false, and • a paused attribute whose value is false, and • either a src attribute or a child source element that references content with a duration of more than 3 seconds that contains audio. **Note:** autoplay and muted are both HTML attributes set on elements. On the other hand, paused is a DOM attribute which is not specified in HTML.
Expectation	**Expectation 1** For each test target a mechanism is provided to pause or stop the audio, or turn the audio volume off independently from the overall system volume control. **Expectation 2** The mechanism to pause or stop or turn the audio volume off is visible, has an accessible name that is not only whitespace , and is included in the accessibility tree.
Assumptions	There are currently no assumptions
Accessibility Support	The native <video> and <audio> controls in several browser and assistive technology combinations are not keyboard accessible and the <video> or <audio> element itself may not be announced. Authors are recommended to use custom controls for keyboard navigation and cross browser accessibility support in general.

13.2.2 Atomic rule: Audio or video that plays automatically has no audio that lasts more than 3 seconds

- **ACT Rules id:** aaa1bf
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **URL:** https://act-rules.github.io/rules/aaa1bf

Description	audio or video that plays automatically does not output audio for more than 3 seconds.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• audio or video avoids automatically playing audio
Input Aspect	DOM Tree, CSS Styling, Audio Output, Visual Output
Applicability	This rule applies to any audio or video element that has: • an autoplay attribute value of true, and • a muted attribute value of false, and • a paused attribute whose value is false, and • either a src attribute or a child source element that references content with a duration of more than 3 seconds that contains audio.

	Note: autoplay and muted are both HTML attributes set on elements. On the other hand, paused is a DOM attribute which is not specified in HTML.
Expectation	For each test target the total audio output does not last more than 3 seconds. **Note:** This rule does not cover single audio instances that play repeatedly for more than three seconds, or multiple audio instances for more than three seconds. The WCAG Understanding documentation for 1.4.2 Audio Controls is ambiguous about how to handle these scenarios.
Assumptions	There are currently no assumptions
Accessibility Support	There are no major accessibility support issues known for this rule.

14 1.4.3 Contrast (Minimum) (Level AA)

14.1 About Success Criteria 1.4.3 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 22.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#contrast-minimum

Success Criteria	The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following: • **Large Text:** Large-scale text and images of large-scale text have a contrast ratio of at least 3:1; • **Incidental:** Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement. • **Logotypes:** Text that is part of a logo or brand name has no contrast requirement.
Purpose	The intent of this Success Criteria is to provide enough contrast between text and its background so that it can be read by people with moderately low vision (who do not use contrast-enhancing assistive technology). For people without color deficiencies, hue and saturation have minimal or no effect on legibility as assessed by reading performance (Knoblauch et al., 1991). Color deficiencies can affect luminance contrast somewhat. Therefore, in the recommendation, the contrast is calculated in such a way that color is not a key factor so that people who have a color vision deficit will also have adequate contrast between the text and the background. Text that is decorative and conveys no information is excluded. For example, if random words are used to create a background and the words could be rearranged or substituted without changing meaning, then it would be decorative and would not need to meet this criteria. Text that is larger and has wider character strokes is easier to read at lower contrast. The contrast requirement for larger text is therefore lower. This allows authors to use a wider range of color choices for large text, which is helpful for design of pages, particularly titles. 18-point text or 14 point bold text is judged to be large enough to require a lower contrast ratio. "18 point" and "bold" can both have different meanings in different fonts but, except for very thin or unusual fonts, they should be sufficient. Since there are so many different fonts, the general measures are used and a note regarding fancy or thin fonts is included. There must be sufficient contrast between the text and its background on the website. This will allow even users with moderately reduced vision to read the text. This also includes users with limited perception of colour. A contrast ratio of 4.5:1 is set as this will compensate for 20/40 vision loss. Such vision loss is common amongst the elderly. This contrast will make text easier for all users to read, e.g. on a small screen (smartphone, tablet) or in sunlight.

User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with low vision • Usage without perception of colour Secondary relationship • Usage with low cognition
Digdir interpretation and specification of the success criteria	The success criteria indicate that the contrast requirement • only applies to text and not to graphic representations or illustrations • varies according to font size and layout (bold vs. non-bold) The contrast requirement between text its background is 3:1 for text more than 24 pixels high in ordinary type or at least 19 pixels high in bold type. For all other text, the contrast requirement is 4.5:1. We cannot verify font sizes in images of text. We have therefore set the limit for contrast for images of text measured against the background in the image to 3:1 (which is the requirement for a large font or bold type), and not 4.5:1 (which is the requirement for other text). The contrast requirement is limited to text that conveys information. Other text is not included in the Authority's test procedures: • Non-informative text that is purely decorative. • Text in logos and trademarks. • Text in images where the text is not essential to convey the meaning of the content. • Disabled form elements. • Text that is not visible to anyone. • Graphic components in e.g. diagrams and graphs, such as guides, columns, etc. The contrast requirement may be met on web pages in ordinary view or in a high-contrast mode that is in accordance with the contrast requirements and has an accessible mechanism for activation. The mechanism for activation is positioned close to the start of the page and is visually marked with a text or icon. The mechanism meets the contrast requirement mentioned in second paragraph above. Text in the high-contrast version meets the contrast requirement mentioned in second paragraph above. The textual component measured with inadequate contrast in ordinary view is identical to the component in the high contrast version. If there is a gradient background and/or text, contrast is measured at the lowest-contrast point.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **afw4f7:** For any visible character in a text node that is a child in the flat tree of an HTML element, the highest possible contrast between the foreground colors and background colors is at least 4.5:1 or 3.0:1 for larger scale text, except if it is part of a text node that is purely decorative or does not express anything in human language.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Sufficient contrast between text and its background in an image of text.

	This is not a comprehensive list of all aspects that may be covered by this success criterion. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

14.2 Test Rule: Text has minimum contrast

- **WAI-Tools rule number:** 46
- **ACT Rules id:** afw4f7
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 23.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/afw4f7

Description	This rule checks that the highest possible contrast of every text character with its background meets the minimal contrast requirement.

Accessibility Requirements	1.4.3 Contrast (Minimum) (Level AA)
	• Required for conformance to WCAG 2.0 and later on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	1.4.6 Contrast (Enhanced) (Level AAA)
	• Required for conformance to WCAG 2.0 and later on level AAA.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	The rule applies to any visible character in a text node that is a child in the flat tree of an HTML element, except if the text node has an ancestor in the flat tree for which one of the following is true: • widget: the ancestor has a semantic role that inherits from widget; or • disabled label: the ancestor is used in the accessible name of a widget that is disabled; or • disabled group: the ancestor has a semantic role of group and is disabled.
Expectation	For each test target, the highest possible contrast between the foreground colors and background colors is at least 4.5:1 or 3.0:1 for larger scale text, except if the test target is part of a text node that is purely decorative or does not express anything in human language.
Assumptions	Success criterion 1.4.3: Contrast (Minimum) has exceptions for "incidental" text, which includes inactive user interface components and decorative texts. The rule assumes that text nodes that should be ignored are disabled or hidden from assistive technologies. If this isn't the case, the text node could fail this rule while the success criterion could still be satisfied.

	Success criterion 1.4.3: Contrast (Minimum) also has an exception for logos and brand names. Since logos and brand names are usually displayed through images to ensure correct rendering, this rule does not take logos or brand names into consideration. If a logo or brand name is included using text nodes, the text node could fail while the success criterion could still be satisfied. Text that has the same foreground and background color (a contrast ratio of 1:1) is not considered to be "visual presentation of text", making it inapplicable to the success criterion. Text hidden in this way can still cause accessibility issues under other success criteria, depending on the content.
Accessibility Support	Different browsers have different levels of support for CSS. This can cause contrast issues in one browser that do not appear in another. Because of that, this rule can produce different results depending on the browser that is used. For example, a text that is positioned using CSS transform may be on a different background in a browser that does not support CSS transform. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of none and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

15 1.4.4 Resize text (Level A)

15.1 About Success Criteria 1.4.4 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#resize-text

Success Criteria	Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.
Purpose	The intent of this Success Criteria is to ensure that visually rendered text, including text-based controls, can be scaled successfully so that it can be read directly by people with mild visual disabilities, without requiring the use of assistive technology such as a screen magnifier.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision Secondary relationship • Usage with limited manipulation or strength
Digdir interpretation and specification of the success criteria	All text can be enlarged to at least 200 % view without any loss of content or functionality. It is possible to zoom using the browser (Ctrl+ or similar). The requirement does not apply to captioning of media content or images of text. The requirement applies to all other text written in natural language, including text in form elements.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **b4f0c3:** The meta element retains the user agent ability to zoom. The content attribute does not specify the property user-scalable with a value of “no”, nor specify the property maximum-scale with a value of less than 2. • **59br37:** Any text node in a viewport size of 640 by 512 which is visible; and has an HTML element as a parent in the flat tree; and has an ancestor in the flat tree with a computed overflow-x or overflow-y of hidden or clip; and does not have an ancestor in the flat tree with an aria-hidden attribute value of true, does not clipped by overflow of an ancestor in flat tree. Except if the clipping ancestor has a computed white-space of nowrap, and a computed text-overflow that is not clip; or a computed line-height equal to or greater than the height of it’s bounding box, or in case of a computed overflow of clip, its content box that is line wrapping.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Digdir’s test procedure checks that the text can be resized in a technology agnostic way, using the browser’s zoom functionality. There may be ways to markup a website that

	prevents text from being enlarged, other than the markup described in the test rule. • The exceptions for captions and images of text are not covered. • The test rule does not check whether content or functionality other than text node is lost when enlarging text. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

15.2 Test Rule: meta viewport allows for zoom

- **WAI-Tools rule number:** 47
- **ACT Rules id:** b4f0c3
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.01.2020
- **Last updated in GitHub:** 01.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/b4f0c3

Description	This rule checks that the meta element retains the user agent ability to zoom.

Accessibility Requirements	1.4.4 Resize text (Level: AA)
	• Required for conformance to WCAG 2.0 and later on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	
	1.4.10 Reflow (Level AA)
	• Required for conformance to WCAG 2.1 on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	The rule applies to each meta element with a name attribute whose value is a case-insensitive match for viewport and has a content attribute.
Expectation	For each test target, the content attribute, whose value is mapped to a list of property/value pairs in a user-agent specific manner, does not: • specify the property user-scalable with a value of no; nor • specify the property maximum-scale with a value of less than 2
Assumptions	Desktop browsers ignore the viewport meta element, and most modern mobile browsers either ignore it by default or have an accessibility option which will allow zooming. This rule is not relevant for desktop browsers, nor for most modern mobile browsers. Only users with older mobile browsers can experience issues tested by this rule.

Accessibility Support	Desktop browsers ignore the viewport meta element, and most modern mobile browsers either ignore it by default or have an accessibility option which will allow zooming. This rule is not relevant for desktop browsers, nor for most modern mobile browsers. Only users with older mobile browsers can experience issues tested by this rule.

15.3 Test Rule: Zoomed text node is not clipped with CSS overflow

- **WAI-Tools rule number:** 50
- **ACT Rules id:** 59br37
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 05.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/59br37

Description	This rule checks that text nodes are not unintentionally clipped by overflow when a page is zoomed to 200% on 1280 by 1024 viewport.

Accessibility Requirements	1.4.4 Resize text (Level: AA)
	• Required for conformance to WCAG 2.0 and later on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any text node for which all of the following is true when in a viewport size of 640 by 512: • The text node is visible; and • The text node has an HTML element as a parent in the flat tree; and • The text node has an ancestor in the flat tree with a computed overflow-x or overflow-y of hidden or clip; and • The text node does not have an ancestor in the flat tree with an aria-hidden attribute value of true. Note: A viewport size of 640 by 512 is equivalent to a viewport size of 1280 by 1024 zoomed 200%.
Expectation	Each test target is not clipped by overflow of an ancestor in the flat tree when in a viewport size of 640 by 512, except if the clipping ancestor has one of the following: • text-overflow: A computed white-space of nowrap, and a computed text-overflow that is not clip; or • line wrapping: A computed line-height equal to or greater than the height of its bounding box, or in case of a computed overflow of clip, its content box.
Assumptions	If any of the following assumptions is true, failing this rule may not result in a failure of success criterion 1.4.4 Resize text: • There is no other mechanism for resizing text available on the page, that can be used to resize text to 200% without loss of information or

	functionality. This includes font resizing in the browser, or a javascript mechanism of resizing in the page. • Text nodes cannot be clipped by overflow without loss of information, except for text nodes with an ancestor with aria-hidden set to true, or when specific styles have been applied to ensure text is clipped cleanly (text-overflow, line wrapping or hidden text). • While success criterion 1.4.4 Resize text does not explicitly mention which viewport size has to be resized up to 200%, it is assumed that a viewport size of 1280 by 1024 is applicable. A 1280 by 1024 viewport size is explicitly mentioned under success criterion 1.4.10 Reflow.
Accessibility Support	Some user agents treat the value of the aria-hidden attribute as case-sensitive.

16 1.4.5 Images of Text (Level AA)

16.1 About Success Criteria 1.4.5 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 22.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#images-of-text

Success Criteria	If the technologies being used can achieve the visual presentation, text is used to convey information rather than images of text except for the following: • **Customizable:** The image of text can be visually customized to the user's requirements; • **Essential:** A particular presentation of text is essential to the information being conveyed. **NOTE:** Logotypes (text that is part of a logo or brand name) are considered essential.
Purpose	The intent of this Success Criteria is to encourage authors, who are using technologies which are capable of achieving their desired default visual presentation, to enable people who require a particular visual presentation of text to be able to adjust the text presentation as needed. This includes people who require the text in a particular font size, foreground and background color, font family, line spacing or alignment. If an author can use text to achieve the same visual effect, he or she should present the information as text rather than using an image. If for any reason, the author cannot format the text to get the same effect, the effect won't be reliably presented on the commonly available user agents, or using a technology to meet this criteria would interfere with meeting other criteria such as 1.4.4, then an image of text can be used. This includes instances where a particular presentation of text is essential to the information being conveyed, such as type samples, logotypes, branding, etc. Images of text may also be used in order to use a particular font that is either not widely deployed or which the author doesn't have the right to redistribute, or to ensure that the text would be anti-aliased on all user agents. Images of text can also be used where it is possible for users to customize the image of text to match their requirements. Images of text may make it difficult to perceive the information being conveyed. One objective is therefore for images of text to be used only when it is not possible to present content in any other way. Using text instead of images of text allows users who need specific presentation of the text to adjust the text according to their needs. This may, for example, involve changing the font size, colour or line spacing, or if assistive technology is needed that makes text available as speech. The definition of image of text contains the note: Note: This does not include text that is part of a picture that contains significant other visual content. Examples of such pictures include graphs, screenshots, and diagrams which visually convey important information through more than just text.

User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision • Usage without perception of colour Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	In some situations, images of text may be necessary in order to achieve a visual effect. According to the understanding article for the success criteria, this is when • it is not possible to achieve the desired visual effect by formatting text • the desired effect cannot be presented in the most common browsers • the presentation requires the use of techniques or technology not compliant with the requirements of WCAG 2.0 Images of text are in compliance with the requirement if the font type, size, colour and background can be adjusted.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

17 1.4.10 Reflow (Level AA)

17.1 About Success Criteria 1.4.10 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 22.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#reflow

Success Criteria	Content can be presented without loss of information or functionality, and without requiring scrolling in two dimensions for: • Vertical scrolling content at a width equivalent to 320 CSS pixels; • Horizontal scrolling content at a height equivalent to 256 CSS pixels. Except for parts of the content which require two-dimensional layout for usage or meaning. **Note:** • 320 CSS pixels is equivalent to a starting viewport width of 1280 CSS pixels wide at 400% zoom. For web content which are designed to scroll horizontally (e.g. with vertical text), the 256 CSS pixels is equivalent to a starting viewport height of 1024px at 400% zoom. • Examples of content which require two-dimensional layout are images, maps, diagrams, video, games, presentations, data tables, and interfaces where it is necessary to keep toolbars in view while manipulating content.
Purpose	The intent of this Success Criteria is to support people with low vision who need to enlarge text and read it in a single column. When the browser zoom is used to scale content to 400%, it reflows - i.e., it is presented in one column so that scrolling in more than one direction is not necessary. For people with low vision, enlarged text with reflow enables reading. It is critical. Enlargement enables perception of characters. Reflow enables tracking. Tracking is following along lines of text, including getting from the end of one line to the beginning of the next line. Avoiding the need to scroll in the direction of reading in order to reveal lines that are cut off by the viewport is important, because such scrolling significantly increases the effort required to read. It is also important that content is not hidden off-screen. For example, zooming into a vertically scrolling page should not cause content to be hidden to one side.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1

Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

18 1.4.11 Non-text Contrast (Level AA)

18.1 About Success Criteria 1.4.11 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 27.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#non-text-contrast

Success Criteria	The visual presentation of the following have a contrast ratio of at least 3:1 against adjacent color(s): • **User Interface Components:** Visual information required to identify user interface components and states, except for inactive components or where the appearance of the component is determined by the user agent and not modified by the author; • **Graphical Objects:** Parts of graphics required to understand the content, except when a particular presentation of graphics is essential to the information being conveyed.
Purpose	The intent of this Success Criteria is to ensure that active user interface components (i.e., controls) and meaningful graphics are distinguishable by people with moderately low vision. The requirements and rationale are similar to those for large text in 1.4.3 Contrast (Minimum). Low contrast controls are more difficult to perceive and may be completely missed by people with a visual impairment. Similarly, if a graphic is needed to understand the content or functionality of the webpage then it should be perceivable by people with low vision or other impairments without the need for contrast-enhancing assistive technology.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision • Usage without perception of colour Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

19 1.4.12 Text spacing (Level AA)

19.1 About Success Criteria 1.4.12 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 28.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#text-spacing

Success Criteria	In content implemented using markup languages that support the following text style properties, no loss of content or functionality occurs by setting all of the following and by changing no other style property: • Line height (line spacing) to at least 1.5 times the font size; • Spacing following paragraphs to at least 2 times the font size; • Letter spacing (tracking) to at least 0.12 times the font size; • Word spacing to at least 0.16 times the font size. Exception: Human languages and scripts that do not make use of one or more of these text style properties in written text can conform using only the properties that exist for that combination of language and script.
Purpose	The intent of this Success Criteria (SC) is to ensure that people can override author specified text spacing to improve their reading experience. Each of the requirements stipulated in the SC's four bullets helps ensure text styling can be adapted by the user to suit their needs. The specified metrics set a minimum baseline. The values in between the author's metrics and the metrics specified in this SC should not have loss of content or functionality. This SC focuses on the ability to increase spacing between lines, words, letters, and paragraphs. Any combination of these may assist a user with effectively reading text. As well, ensuring users can override author settings for spacing also significantly increases the likelihood other style preferences can be set by the user. For example, a user may need to change to a wider font family than the author has set in order to effectively read text.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with low vision • Usage with low cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.

Gap analysis suggested by Digdir – For discussion	None.

20 1.4.13 Content on Hover or Focus (Level AA)

20.1 About Success Criteria 1.4.13 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 28.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#content-on-hover-or-focus

Success Criteria	Where receiving and then removing pointer hover or keyboard focus triggers additional content to become visible and then hidden, the following are true: • **Dismissable:** A mechanism is available to dismiss the additional content without moving pointer hover or keyboard focus, unless the additional content communicates an input error or does not obscure or replace other content; • **Hoverable:** If pointer hover can trigger the additional content, then the pointer can be moved over the additional content without the additional content disappearing; • **Persistent:** The additional content remains visible until the hover or focus trigger is removed, the user dismisses it, or its information is no longer valid. **Exception:** The visual presentation of the additional content is controlled by the user agent and is not modified by the author.
Purpose	Additional content that appears and disappears in coordination with keyboard focus or pointer hover often leads to accessibility issues. Reasons for such issues include: 1. the user may not have intended to trigger the interaction 2. the user may not know new content has appeared 3. the new content may intefere with a user's ability to do a task Examples of such interactions can include custom tooltips, sub-menus and other nonmodal popups which display on hover and focus. The intent of this success criteria is to ensure that authors who cause additional content to appear and disappear in this manner must design the interaction in such a way that users can: • perceive the additional content AND • dismiss it without disrupting their page experience. There are usually more predictable and accessible means of adding content to the page, which authors are recommended to employ. If an author *does* choose to make additional content appear and disappear in coordination with hover and keyboard focus, this success criteria specifies three conditions that must be met: • dismissable • hoverable • persistent Each of these is discussed in a separate section.

	NOTE: This SC covers content that appears in addition to the triggering component itself. Therefore, a non-visible component, like a Skip to Main link, that is made visible on keyboard focus (with no *additional* content beyond the trigger becoming visible) is not covered by the SC.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

21 2.1.1 Keyboard (Level A)

21.1 About Success Criteria 2.1.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 28.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#keyboard

Success Criteria	All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints. **NOTE:** This exception relates to the underlying function, not the input technique. For example, if using handwriting to enter text, the input technique (handwriting) requires path-dependent input but the underlying function (text input) does not. This does not forbid and should not discourage providing mouse input or other input methods in addition to keyboard operation.
Purpose	The intent of this Success Criteria is to ensure that, wherever possible, content can be operated through a keyboard or keyboard interface (so an alternate keyboard can be used). When content can be operated through a keyboard or alternate keyboard, it is operable by people with no vision (who cannot use devices such as mice that require eye-hand coordination) as well as by people who must use alternate keyboards or input devices that act as keyboard emulators. Keyboard emulators include speech input software, sip-and-puff software, on-screen keyboards, scanning software and a variety of assistive technologies and alternate keyboards. Individuals with low vision also may have trouble tracking a pointer and find the use of software much easier (or only possible) if they can control it from the keyboard. Examples of "specific timings for individual keystrokes" include situations where a user would be required to repeat or execute multiple keystrokes within a short period of time or where a key must be held down for an extended period before the keystroke is registered. The phrase "except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints" is included to separate those things that cannot reasonably be controlled from a keyboard. Most actions carried out by a pointing device can also be done from the keyboard (for example, clicking, selecting, moving, sizing). However, there is a small class of input that is done with a pointing device that cannot be done from the keyboard in any known fashion without requiring an inordinate number of keystrokes. Free hand drawing, watercolor painting, and flying a helicopter through an obstacle course are all examples of functions that require path dependent input. Drawing straight lines, regular geometric shapes, re-sizing windows and dragging objects to a location (when the path to that location is not relevant) do not require path dependent input.

	The use of MouseKeys would not satisfy this Success Criteria because it is not a keyboard equivalent to the application; it is a mouse equivalent (i.e., it looks like a mouse to the application). It is assumed that the design of user input features takes into account that operating system keyboard accessibility features may be in use. For example, modifier key locking may be turned on. Content continues to function in such an environment, not sending events that would collide with the modifier key lock to produce unexpected results.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength Secondary relationship • Usage without vocal capability
Digdir interpretation and specification of the success criteria	All content and functionality on the web page can be accessed and used with a keyboard. Content and/or functionality that it is not appropriate to use with a keyboard is excepted. It follows from the success criteria that this requirement does not apply to functionality that is not appropriate to operate with a keyboard. This may, for example, include freehand drawing or computer games where objects are guided through various obstacles, etc. If there is more than one way of operating certain functionalities, such as the possibility to both select the date in a calendar and enter the date directly in a date field, it is sufficient that one of the methods can be used with a keyboard.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **0ssw9k:** Any HTML element that has visible children in the flat tree which has a horizontal scroll distance greater than the computed left or right padding of element; or has a vertical scroll distance greater than the computed top or bottom padding of then element; is either included in sequential focus navigation or has a descendant in the flat tree that is included in sequential focus navigation.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • All content and functionality on the web page can be accessed and used with a keyboard. Content and/or functionality that it is not appropriate to use with a keyboard is expected. ○ Focusable content and functionality are keyboard accessible and operable. This is not a comprehensive list of all aspects that may be covered by this success criterion. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

21.2 Test Rule: Scrollable element is keyboard accessible

- **WAI-Tools rule number:** 58

- **ACT Rules id:** 0ssw9k
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/0ssw9k

Description	This rule checks that scrollable elements can be scrolled by keyboard.

Accessibility Requirements	2.1.1 Keyboard (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.1.3 Keyboard (No Exception) (Level AAA)
	• Required for conformance to WCAG 2.0 and later on level AAA.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	Any HTML element that has visible children in the flat tree for which one of the following is true: • It has a horizontal scroll distance greater than the computed left or right padding of the element; or • It has a vertical scroll distance greater than the computed top or bottom padding of the element
Expectation	Each test target is either included in sequential focus navigation or has a descendant in the flat tree that is included in sequential focus navigation.
Assumptions	This rule assumes that all scrollable elements with visible content need to be keyboard accessible. Scrollable elements that do not need to be keyboard accessible, perhaps because their content is purely decorative or because scroll can be controlled some other way, may fail this rule but still satisfy success criterion 2.1.1 Keyboard.
Accessibility Support	No accessibility support issues known.

22 2.1.2 No Keyboard Trap (Level A)

22.1 About Success Criteria 2.1.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#no-keyboard-trap

Success Criteria	If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away. **Note:** Since any content that does not meet this success criteria can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criteria. See Conformance Requirement 5: Non-Interference.
Purpose	The intent of this Success Criteria is to ensure that that content does not "trap" keyboard focus within subsections of content on a Web page.
User accessibility needs (Functional Performance Statements)	Primary relationship (Source EN 301 549 standard) • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength Secondary relationship • Usage without vocal capability
Digdir interpretation and specification of the success criteria	It is possible to navigate through all content on the web page using a keyboard without becoming trapped in any component.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that
	• **80af7b:** For any HTML or SVG element that is focusable, one of the following is true:
	  ○ **a1b64e:** Focus can cycle to the browser UI by using standard keyboard navigation (Tab key, Shift+Tab, Arrow keys, Esc key, Enter key and Space key).
	  ○ **ebe86a:** There is help information that is visible and included in the accessibility tree or can be accessed from within the keyboard trap, that explains how to cycle to the browser UI, or on how to get to a point from where it is possible to cycle to the browser UI, using standard keyboard navigation. Focus can cycle to the browser UI by using the method advised in the help information.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Focusable elements that are not HTML or SVG elements.

	Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

22.2 Test Rule: Focusable element has no keyboard trap

- **WAI-Tools rule number:** 26
- **ACT Rules id:** 80af7b
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Composite
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/80af7b

Description	This rule checks for keyboard traps. This includes use of both standard and non-standard keyboard navigation to navigate through all content without becoming trapped.

Accessibility Requirements	2.1.2 No Keyboard Trap (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input rules (composite only)	• Focusable element has no keyboard trap via standard navigation • Focusable element has no keyboard trap via non-standard navigation
Applicability	The rule only applies to any HTML or SVG element that is focusable. **Note:** This rule only applies to HTML and SVG. Thus, it is a partial check for WCAG 2.0 success criteria 2.1.2, which applies to all content.
Expectation	For each test target, the outcome of one of the following rules is "passed": • Focusable element has no keyboard trap via standard navigation • Focusable element has no keyboard trap via non-standard navigation
Assumptions	There are currently no assumptions.
Accessibility Support	There are no major accessibility support issues known for this rule.

22.2.1 Atomic rule: Focusable element has no keyboard trap via standard navigation

- **ACT Rules id:** a1b64e
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/a1b64e

Description	This rule checks if it is possible to use standard keyboard navigation to navigate through all content on a web page without becoming trapped in any element.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Focusable element has no keyboard trap
Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any HTML or SVG element that is focusable. **Note:** This rule only applies to HTML and SVG. Thus, it is a partial check for WCAG 2.0 success criteria 2.1.2, which applies to all content.
Expectation	For each target element focus can cycle to the browser UI by using standard keyboard navigation. **Note:** Cycling back to the browser UI can be done both by moving forward through the tab order and by moving backwards. It is not possible to fulfil this expectation by using browser specific shortcuts to return to the browser UI.
Assumptions	The focus order in keyboard navigation is cyclical, not linear, meaning that the focus order will cycle to the first/last element when it moves away from the last/first element. The Browser UI is part of the focus navigation cycle of the page.
Accessibility Support	There are no major accessibility support issues known for this rule.

22.2.2 Atomic rule: Focusable element has no keyboard trap via non-standard navigation

- **ACT Rules id:** ebe86a
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2019
- **Rule Type:** Atomic (belongs to composite rule)
- **URL:** https://act-rules.github.io/rules/ebe86a

Description	This rule checks if it is possible to use non-standard keyboard navigation to navigate through content where focus is trapped when using standard ways of keyboard navigation.
Accessibility Requirements	This rule is not required for conformance
Used in Rules	• Focusable element has no keyboard trap
Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any HTML or SVG element that is focusable where focus cannot cycle to the browser UI by using standard keyboard navigation. **Note:** This rule only applies to HTML and SVG. Thus, it is a partial check for WCAG 2.0 success criteria 2.1.2, which applies to all content.
Expectation	**Expectation 1**

	For each target element help information is visible and included in the accessibility tree or can be accessed from within the keyboard trap. **Note:** As per WCAG 2.0 Success Criteria 2.1.1 Keyboard the help information should be accessible through a keyboard interface. **Expectation 2** The help information explains how to cycle to the browser UI, or on how to get to a point from where it is possible to cycle to the browser UI, using standard keyboard navigation. **Expectation 3** For each target element focus can cycle to the browser UI by using the method advised in the help information. **Note:** Cycling back to the browser UI can be done both by moving forward through the tab order and by moving backwards. It is not possible to fulfil this expectation by using browser specific shortcuts to return to the browser UI.
Assumptions	It is not possible to use unmodified arrow or tab keys, or other standard exit methods to move focus away. The focus order in keyboard navigation is cyclical, not linear, meaning that the focus order will cycle to the first/last element when it moves away from the last/first element.
Accessibility Support	There are no major accessibility support issues known for this rule.

23 2.1.4 Character Key Shortcuts (Level A)

23.1 About Success Criteria 2.1.4 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#character-key-shortcuts

Success Criteria	If a keyboard shortcut is implemented in content using only letter (including upper- and lower-case letters), punctuation, number, or symbol characters, then at least one of the following is true: • **Turn off:** A mechanism is available to turn the shortcut off; • **Remap:** A mechanism is available to remap the shortcut to use one or more non-printable keyboard characters (e.g. Ctrl, Alt, etc); • **Active only on focus:** The keyboard shortcut for a user interface component is only active when that component has focus.
Purpose	The intent of this Success Criteria is to reduce accidental activation of keyboard shortcuts. Character key shortcuts work well for many keyboard users but are inappropriate and frustrating for speech input users — whose means of input is strings of letters — and for keyboard users who are prone to accidentally hit keys. To rectify this issue, authors need to allow users to turn off or reconfigure shortcuts that are made up of only character keys. Note that this success criteria doesn't affect components such as listboxes and drop-down menus. Although these components contain values (words) that may be selected by one or more character keys, the shortcuts are only active when the components have focus. Other components such as menus may be accessed or opened with a single non-character shortcut (e.g., Alt or Alt+F) before pressing a single character key to select an item. This makes the full path to invoking a menu a two-step shortcut that includes a non-printable key. Accesskeys are also not affected because they include modifier keys. Speech Input users generally work in a single mode where they can use a mix of dictation and speech commands. This works well because the user knows to pause before and after commands, and commands are usually at least two words long. So, for instance, a user might say a bit of dictation, such as "the small boat", then pause, and say a command to delete that dictation, such as "Delete Line". In contrast, if the user were to say the two phrases together without a pause, the whole phrase would come out as dictation (i.e., "the small boat delete line"). Although speech input programs often include modes that listen only for dictation or only for commands, most speech users use the all-encompassing mode all the time because it is a much more efficient workflow. It could decrease command efficiency significantly if users were to change to command mode and back before and after issuing each command. Speech users can also speak most keyboard commands (e.g., "press Control Foxtrot") without any problems. If the website or app is keyboard enabled, the speech user can also write a native speech

	macro that calls the keyboard command, such as "This Print" to carry out Ctrl+P. Single-key shortcuts are the exception. While using single letter keys as controls might be appropriate and efficient for many keyboard users, single-key shortcuts are disastrous for speech users. The reason for this is that when only a single key is used to trip a command, a spoken word can become a barrage of single-key commands if the cursor focus happens to be in the wrong place. For example, a speech-input user named Kim has her cursor focus in the main window of a web mail application that uses common keyboard shortcuts to navigate ("k"), archive ("y") and mute messages ("m"). A coworker named Mike enters her office and says "Hey Kim" and her microphone picks that up. The Y of "hey" archives the current message. K in "Kim" moves down one conversation and M mutes a message or thread. And, if Kim looks up and says "Hey Mike" without remembering to turn off the microphone, the same three things happen in a different sequence. A user interacting with a webpage or web app that doesn't use single-character shortcuts doesn't have this problem. Inadvertent strings of characters from the speech application are not interpreted as shortcuts if a modifier key is required. A speech user filling in a text input form may find that a phrase that is accidentally picked up by the speech microphone results in stray text being entered into the field, but that is easily seen and undone. The Resources section of this page contains links to videos demonstrating these types of issues.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited manipulation or strength • Usage with limited reach Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **ffbc54:** For any keyboard event implemented in content using a printable character key as event's attribute key, getModifierStates returns false for each of the valid modifier keys as event's method, and the event causes changes in the content of the HTML document. There exist at least one set of clearly labelled remapping instrument to block events that use the same key as the keyboard event and whose getModifierState method returns false for each of the valid modifier keys or on focus the event target has a semantic role that inherits from the abstract of widget.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

23.2 Test Rule: No keyboard shortcut uses only printable characters

- **WAI-Tools rule number:** 53
- **ACT Rules id:** ffbc54
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 06.10.2020
- **Last updated in GitHub:** 05.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/ffbc54

Description	This rule checks that if keyboard shortcuts are implemented using only printable characters, then there is a mechanism to disable the shortcut, or to remap the shortcut to use one or more non-printable character keys, or the shortcut for a user interface component is only available when that component has focus.

Accessibility Requirements	2.1.4 Character Key Shortcuts (Level A)
	• Required for conformance to WCAG 2.1 on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion is satisfied.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	The rule applies to any keyboard event for which all of the following is true: • the event's attribute key is a printable character key; and • the event's method getModifierState returns false for each of the valid modifier keys; and • the event causes changes in the content of the HTML document.
Expectation	For each test target at least one of the following is true: • (remap:) there is at least one set of clearly labelled instruments to block events that use the same key as the test target and whose getModifierState method returns false for each of the valid modifier keys; or • (focus:) the event target has a semantic role that inherits from the abstract role of widget.
Assumptions	If there are ways to disable the result of keyboard events that do not require the user to interact with the web page (e.g. a setting at the operating system level), failing this rule might not be a failure of the success criterion.
Accessibility Support	Currently keyboard events only support the types keydown and keyup. Keyboard events of type keypressed are considered legacy keyboard events and are thus ignored by this rule.

24 2.2.1 Timing Adjustable (Level A)

24.1 About Success Criteria 2.2.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#timing-adjustable

Success Criteria	For each time limit that is set by the content, at least one of the following is true: • **Turn off:** The user is allowed to turn off the time limit before encountering it; or • **Adjust:** The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or • **Extend:** The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action (for example, "press the space bar"), and the user is allowed to extend the time limit at least ten times; or • **Real-time Exception:** The time limit is a required part of a real-time event (for example, an auction), and no alternative to the time limit is possible; or • **Essential Exception:** The time limit is essential and extending it would invalidate the activity; or • **20 Hour Exception:** The time limit is longer than 20 hours. **Note:** This success criteria helps ensure that users can complete tasks without unexpected changes in content or context that are a result of a time limit. This success criteria should be considered in conjunction with Success Criteria 3.2.1, which puts limits on changes of content or context as a result of user action.
Purpose	The intent of this Success Criteria is to ensure that users with disabilities are given adequate time to interact with Web content whenever possible. If Web functions are time-dependent, it will be difficult for some users to perform the required action before a time limit occurs. In some cases, however, it is not possible to change the time limit (for example, for an auction or other real-time event) and exceptions are therefore provided for those cases.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without hearing • Usage with limited hearing • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • None

Digdir interpretation and specification of the success criteria	Any process that happens without user initiation after a set time or on a periodic basis is a time limit. This includes partial or full updates of content (for example, page refresh), changes to content, or the expiration of a window of opportunity for a user to react to a request for input. It also includes content that is advancing or updating at a rate beyond the user's ability to read and/or understand it. In other words, animated, moving or scrolling content introduces a time limit on a users ability to read content. This success criteria explains exceptions to the requirement when the time limit • is a necessary component in a real-time event (e.g. an auction), and there is an alternative • is necessary, and extension will invalidate the action • lasts more than 20 hours The requirement can be met in several different ways. For web pages that include time limits, one of the following requirements is met: • It is possible to disable the time limit. • It is possible to adjust the time limit to at least 10 times the standard duration. The mechanism for adjustment must come before the content that includes time limits. • It is possible to extend the time limit at least 20 seconds before it expires by executing a simple action. It must be possible to extend by at least 10 times the standard duration.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **bc659a:** The first valid <meta http-equiv="refresh"> element with a content attribute in a document has a time of the content attribute that is 0 (i.e. the page does not reload) or greater than 72000 (i.e. the page reloads after 20 hours or more).
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • It is possible to disable the time limit. • It is possible to adjust the time limit to at least 10 times the standard duration. • It is possible to extend the time limit at least 20 seconds before it expires by executing a simple action. It must be possible to extend the time limit by at least 10 times the standard duration. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

24.2 Test Rule: meta element has no refresh delay

- **WAI-Tools rule number:** 10
- **ACT Rules id:** bc659a
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 19.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto

- **URL:** https://act-rules.github.io/rules/bc659a

Description	This rule checks that the meta element is not used for delayed redirecting or refreshing.

Accessibility Requirements	2.2.1 Timing Adjustable (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.2.4 Interruptions (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	3.2.5 Change on Request (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	The rule applies to the first valid <meta http-equiv="refresh"> element with a content attribute in a document.
Expectation	The time of the content attribute is 0 or greater than 72000 (20 hours). **Note:** See Refresh state (http-equiv="refresh") for a precise description on how to determine the time.
Assumptions	• This test assumes no functionality was provided by the website for the user to adjust the timer. • This test assumes that the refresh was not essential, which is listed as a valid exception to SC 2.2.1.
Accessibility Support	Not all major web browsers parse the value of the content attribute in the same way. This can cause redirects to happen in some browsers, but not in others. This can cause some pages to pass this rule, while still having a redirect in a minority of web browsers.

25 2.2.2 Pause, Stop, Hide (Level A)

25.1 About Success Criteria 2.2.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#pause-stop-hide

Success Criteria	For moving, blinking, scrolling, or auto-updating information, all of the following are true: • **Moving, blinking, scrolling:** For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential; and • **Auto-updating:** For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential. **NOTE:** Content that is updated periodically by software or that is streamed to the user agent is not required to preserve or present information that is generated or received between the initiation of the pause and resuming presentation, as this may not be technically possible, and in many situations could be misleading to do so. An animation that occurs as part of a preload phase or similar situation can be considered essential if interaction cannot occur during that phase for all users and if not indicating progress could confuse users or cause them to think that content was frozen or broken.
Purpose	The intent of this Success Criteria is to avoid distracting users during their interaction with a Web page. "Moving, blinking and scrolling" refers to content in which the visible content conveys a sense of motion. Common examples include motion pictures, synchronized media presentations, animations, real-time games, and scrolling stock tickers. "Auto-updating" refers to content that updates or disappears based on a preset time interval. Common time-based content includes audio, automatically updated weather information, news, stock price updates, and auto-advancing presentations and messages. The requirements for moving, blinking and scrolling content and for auto-updating content are the same except that: authors have the option of providing the user with a means to control the frequency of updates when content is auto-updating and there is no five second exception for auto-updating since it makes little sense to auto-update for a few seconds and then stop Content that moves or auto-updates can be a barrier to anyone who has trouble reading stationary text quickly as well as anyone who has trouble tracking moving objects. It can also cause problems for screen readers.

	Moving content can also be a severe distraction for some people. Certain groups, particularly those with attention deficit disorders, find blinking content distracting, making it difficult for them to concentrate on other parts of the Web page. Five seconds was chosen because it is long enough to get a user's attention, but not so long that a user cannot wait out the distraction if necessary to use the page. Content that is paused can either resume in real-time or continue playing from the point in the presentation where the user left off. • Pausing and resuming where the user left off is best for users who want to pause to read content and works best when the content is not associated with a real-time event or status. NOTE: See 2.2.1: Timing Adjustable for additional requirements related to time-limits for reading. • Pausing and jumping to current display (when pause is released) is better for information that is real-time or "status" in nature. For example, weather radar, a stock ticker, a traffic camera, or an auction timer, would present misleading information if a pause caused it to display old information when the content was restarted. NOTE: Hiding content would have the same result as pausing and jumping to current display (when pause is released). For a mechanism to be considered "a mechanism for the user to pause," it must provide the user with a means to pause that does not tie up the user or the focus so that the page cannot be used. The word "pause" here is meant in the sense of a "pause button" although other mechanisms than a button can be used. Having an animation stop only so long as a user has focus on it (where it restarts as soon as the user moves the focus away) would not be considered a "mechanism for the user to pause" because it makes the page unusable in the process and would not meet this SC.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage without limited vision • Usage without hearing • Usage with limited hearing • Usage with limited manipulation or strength
Digdir interpretation and specification of the success criteria	According to the understanding article for the success criteria, there must be a separate mechanism for pausing content, e.g. a pause button. It must be possible to use other content on the page when content has been paused. It is not sufficient for content only to be paused when it is in focus (with the cursor or keyboard focus). The objective of avoiding distracting the user is not met in this case. The requirement does not apply to content that only moves for up to five seconds. This is because the content producer then has the opportunity to use movement, rolling or blinking as a way of attracting the user’s attention, while at the same time the distraction is brief enough for the user to be able to wait. For web pages with content that moves, blinks or scrolls for more than 5 seconds and that is displayed together with other content and is not a necessary part of an action, one of the following requirements is met:

	• There is a mechanism for pausing, stopping or hiding the content. The mechanism is located either directly next to the content or at the start of the page. • There is a documented keyboard shortcut for pausing, stopping or hiding the content. • There is a mechanism close to the start of the page for reloading the page without this type of content. For web pages with auto-updating content that is displayed together with other content and is not a necessary part of an action, one of the following requirements is met: • There is a mechanism for pausing, stopping or hiding the content or controlling the update frequency. The mechanism is located either directly next to the content or at the start of the page. • There is a documented keyboard shortcut for pausing, stopping or hiding the content or controlling the update frequency.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **efbfc7:** Any HTML element that has a visible text node as a descendant in the flat tree if: *changed*, the innerText property of the element changes multiple times within a 10 minute time span where there is no user interaction; and *no child changed*, the element does not have children in the flat tree whose innerText property also changes; and *not alone*, the element has an ancestor element in the flat tree with a non-empty innerText property whose value is different from the innerText of the test target. For each test target there is at least one set of instruments, where each instrument is in the same web page as the test target or can be found in a clearly labeled location from that web page, to achieve one of the following objectives: pause and resume the change of the visible text content; or stop the change of the visible text content; or hide the changing visible text content; or control the frequency of the changes of the visible text content.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • It is possible to pause, stop or hide content that move, blinks or scrolls (that starts automatically and does not stop within five seconds). • It is possible to pause, stop, hide, or change the update frequency for auto-updating content other than text content. This is not a comprehensive list of all aspects that may be covered by this success criterion. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

25.2 Test Rule: Text content that changes automatically can be paused, stopped or hidden

- **WAI-Tools rule number:** 55
- **ACT Rules id:** efbfc7
- **Status in WAI-Tools:** Completed

- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 03.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/efbfc7

Description	This rule checks that for any text content that automatically changes in a 10 minute time span, there are instruments to pause, stop, or hide it or to control its changing frequency. The arbitrary 10 minute time span, selected so that testing this rule would not be impractical, is not included in WCAG. Content that changes less frequently may fail success criteria 2.2.2 without failing this rule.

Accessibility Requirements	2.2.2 Pause, Stop, Hide (Level: A)
	• Required for conformance to WCAG 2.0 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any HTML element that has a visible text node as a descendant in the flat tree if: • **changed:** the innerText property of the element changes multiple times within a 10 minute time span where there is no user interaction; and • **no child changed:** the element does not have children in the flat tree whose innerText property also changes; and • **not alone:** the element has an ancestor element in the flat tree with a non-empty innerText property whose value is different from the innerText of the test target.
Expectation	For each test target there is at least one set of instruments, where each instrument is in the same web page as the test target or can be found in a clearly labeled location from that web page, to achieve one of the following objectives: • pause and resume the change of the visible text content; or • stop the change of the visible text content; or • hide the changing visible text content; or • control the frequency of the changes of the visible text content. **Note:** If there is more than one test target, the same instrument may be used to pause (or stop, or hide or alter the frequency) of several or even all test targets.
Assumptions	• The auto-updating of the content is not essential, which is listed as valid exception to Success Criterion 2.2.2: Pause, Stop, Hide. When the auto-updating of content is essential this rule may produce incorrect results. • The content being changed automatically is information. If the automatically changing content is not information (for example, an ASCII rendered spinning icon that does not provide information on what time is left for a process to end or how much progress has been made) the rule might fail but the success criterion might still be satisfied. • Any content changes are enabled by the content of the HTML document the test target belongs to. Changes originating from any other sources (e.g. browser shortcuts, browser extensions, browser settings, user agents, external browser applications) are not considered.

	• All user actions are transmitted by the user agent to the HTML document. If there are other event sources that result from a user action this rule might fail but the success criterion might still be satisfied. • Available mechanisms for controlling the content changes rely on activation. If there are other mechanisms that do not rely on activation then the rule might fail but the success criterion might still be satisfied. • If there are other ways to control the automatically changing content that do not require the user to interact with the web page, failing this rule might not be a failure of the success criterion. • This rule does not check that the pausing instrument does not tie up the user focus. If that happens, then this rule might pass but the success criterion would not be satisfied.
Accessibility Support	There are no major accessibility support issues known for this rule.

26 2.2.4 Interruptions (Level AAA)

26.1 About Success Criteria 2.2.4 and Interpretation

- **Level:** AAA
- **WAD:** No
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#interruptions

Success Criteria	Interruptions can be postponed or suppressed by the user, except interruptions involving an emergency.
Purpose	The intent of this Success Criteria is to allow users to turn off updates from the author/server except in emergencies. Emergencies would include civil emergency alert messages or any other messages that warn of danger to health, safety, or property, including data loss, loss of connection, etcetera. This allows access by people with cognitive limitations or attention disorders by enabling them to focus on the content. It also allows users who are blind or have low vision to keep their "viewing" focus on the content they are currently reading.
User accessibility needs (Functional Performance Statements)	Does not apply. SC 2.2.4 is not a part of WAD.
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, test rules check that • **bc659a:** The first valid <meta http-equiv="refresh"> element with a content attribute in a document has a time of the content attribute that is 0 (i.e. the page does not reload) or greater than 72000 (i.e. the page reloads after 20 hours or more).
Gap analysis suggested by Digdir – For discussion	Irrelevant.

26.2 Test Rule: meta element has no refresh delay

- **WAI-Tools rule number:** 10
- **ACT Rules id:** bc659a
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 19.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/bc659a

Description	This rule checks that the meta element is not used for delayed redirecting or refreshing.

Accessibility Requirements	2.2.1 Timing Adjustable (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.2.4 Interruptions (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	3.2.5 Change on Request (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	The rule applies to the first valid <meta http-equiv="refresh"> element with a content attribute in a document.
Expectation	The time of the content attribute is 0 or greater than 72000 (20 hours). **Note:** See Refresh state (http-equiv="refresh") for a precise description on how to determine the time.
Assumptions	• This test assumes no functionality was provided by the website for the user to adjust the timer. • This test assumes that the refresh was not essential, which is listed as a valid exception to SC 2.2.1.
Accessibility Support	Not all major web browsers parse the value of the content attribute in the same way. This can cause redirects to happen in some browsers, but not in others. This can cause some pages to pass this rule, while still having a redirect in a minority of web browsers.

27 2.3.1 Three Flashes or Below (Level A)

27.1 About Success Criteria 2.3.1 and Interpretation

- **Level:** A
- **WAD:** Yes/No
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#three-flashes-or-below-threshold

Success Criteria	Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds.
Purpose	The intent of this Success Criteria is to allow users to access the full content of a site without inducing seizures due to photosensitivity. Individuals who have photosensitive seizure disorders can have a seizure triggered by content that flashes at certain frequencies for more than a few flashes. People are even more sensitive to red flashing than to other colors, so a special test is provided for saturated red flashing. These guidelines are based on guidelines for the broadcasting industry as adapted for computer screens, where content is viewed from a closer distance (using a larger angle of vision). Flashing can be caused by the display, the computer rendering the image or by the content being rendered. The author has no control of the first two. They can be addressed by the design and speed of the display and computer. The intent of this criteria is to ensure that flicker that violates the flash thresholds is not caused by the content itself. For example, the content could contain a video clip or animated image of a series of strobe flashes, or close-ups of rapid-fire explosions. This Success Criteria replaces a much more restrictive criteria in WCAG 1.0 that did not allow any flashing (even of a single pixel) within a broad frequency range (3 to 50 Hz). This Success Criteria is based on existing specifications in use in the UK and by others for television broadcast and has been adapted for computer display viewing. The 1024 x 768 screen is used as the reference screen resolution for the evaluation. The 341 x 256 pixel block represents a 10 degree viewport at a typical viewing distance. (The 10 degree field is taken from the original specifications and represents the central vision portion of the eye, where people are most susceptible to photo stimuli.) The combined area of flashes occurring concurrently and contiguously means the total area that is actually flashing at the same time. It is calculated by adding up the contiguous area that is flashing simultaneously within any 10 degree angle of view.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with minimize photosensitive seizure triggers
Digdir interpretation and specification of the success criteria	All content on the website must comply with the requirement, with the exception of blinking or flashing due to the screen or loading of images (cf. the understanding article). The web page has no content that flashes. If the web page has content that blinks, then the content that

	flashes is smaller than 21,824 square pixels in size or there are fewer than three flashes per second. In WCAG's definitions of terms, there is a specific explanation of what constitutes flashing and red flashing, with associated threshold values: • Blinking is switch back and forth between two visual states in a way that is meant to draw attention. • Flashing is a pair of opposing changes in relative luminance that can cause seizures in some people if it is large enough and in the right frequency range. • Red flashing is defined as any pair of opposing transitions involving a saturated red. • Flashing must not occur at a frequency of more than three times per second, or cover more than 25% of any 10-degree visual field on screen. • Blinking above the threshold values must not occur at all. It is not sufficient to make the user aware that the page includes flashing. It is difficult and time-consuming to verify web solutions against this success criteria. The test procedure also requires access to tools for measurement.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

28 2.4.1 Bypass Blocks (Level A)

28.1 About Success Criteria 2.4.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#bypass-blocks

Success Criteria	A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.
Purpose	The intent of this Success Criteria is to allow people who navigate sequentially through content more direct access to the primary content of the Web page. Web pages and applications often have content that appears on other pages or screens. Examples of repeated blocks of content include but are not limited to navigation links, heading graphics, and advertising frames. Small repeated sections such as individual words, phrases or single links are not considered blocks for the purposes of this provision. This is in contrast to a sighted user's ability to ignore the repeated material either by focusing on the center of the screen (where main content usually appears) or a mouse user's ability to select a link with a single mouse click rather than encountering every link or form control that comes before the item they want. It is not the intent of this Success Criteria to require authors to provide methods that are redundant to functionality provided by the user agent. Most web browsers provide keyboard shortcuts to move the user focus to the top of the page, so if a set of navigation links is provided at the bottom of a web page providing a "skip" link may be unnecessary. Although the success criteria does not specifically use the term “within a set of web pages”, the concept of the pages belonging to a set is implied. An author would not be expected to avoid any possible duplication of content in any two pages that are not in some way related to each other; that are not "Web pages that share a common purpose and that are created by the same author, group or organization” (the definition of set of web pages). NOTE: Even for web pages that are not in a set, if a web page has blocks of text that are repeated within the page it may be helpful (but not required) to provide a means to skip over them.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • Usage without vocal capability

Digdir interpretation and specification of the success criteria	The understanding article for the success criteria shows that the requirement applies to web pages that belong to the same website, i.e. web pages that share a collective purpose and that are created by the same author, group or organisation. An explanation is also provided of situations that are excepted from the requirement: • Small, repeated sections such as individual words, phrases or simple links are not deemed to be blocks. • There is no requirement to offer methods or navigation options that are already available in the user agent (browser or screen reader). If the user is able to tab through to the primary content relatively quickly (within five tab steps), no requirements are defined for skip links. For web pages with blocks of content that are repeated in multiple locations on the website and that come before the primary content: • There is a mechanism for skipping to the primary content. • This mechanism is within the first three tab steps, and before the block with content on the web page. • The mechanism is visible when it receives keyboard focus. • The mechanism has a descriptive text. • Visual and functional focus are transferred to the primary content when the mechanism is activated.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

29 2.4.2 Page Titled (level A)

29.1 About Success Criteria 2.4.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#page-titled

Success Criteria	Web pages have titles that describe topic or purpose.
Purpose	The intent of this Success Criteria is to help users find content and orient themselves within it by ensuring that each Web page has a descriptive title. User agents make the title of the page easily available to the user for identifying the page. For instance, a user agent may display the page title in the window title bar or as the name of the tab containing the page.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The page title must provide relevant information on the content of the page, even when the page title is read out of context (cf. technique G88). There is no requirement for page titles to be unique for each individual web page on the website, as long as it provides a relevant description of the content. Note that the page title must not be confused with the main web page heading. Page titles are located in the <title> element, which is nested in the <head> element at the start of the HTML markup.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **2779a5:** The root element of the web page, if it is an html element, has at least one descendant that is an HTML title element. The first HTML title element that is a descendant of the document element has children that are text nodes that are not only whitespace. • **c4a8a4:** The first HTML title element that is a descendant of the html element of a web page and contains children that are text nodes that are not only whitespace, describes the topic or purpose of overall content of the document.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

29.2 Test Rule: HTML page has non-empty title

- **WAI-Tools rule number:** 5
- **ACT Rules id:** 2779a5
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 03.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/2779a5

Description	This rule checks that a non-embedded HTML page has a non-empty title.

Accessibility Requirements	2.4.2 Page Titled (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	The root element of the web page, if it is an html element.
Expectation	**Expectation 1** Each target element has at least one descendant that is a title element. **Expectation 2** For each target element, the first HTML title element that is a descendant of the document element has children that are text nodes that are not only whitespace.
Assumptions	This rule assumes that Success Criterion 2.4.2 Page Titled does not require that a document only has one title element, nor that it is a child of the head element of a document. While this is invalid in HTML, the HTML 5.2 specification describes what should happen in case of multiple titles, and titles outside the head element. Because of this, neither of these validation issues causes a conformance problem for WCAG. Regardless of whether this is required by 2.4.2 Page Titled, failing this rule means the success criterion is not satisfied.
Accessibility Support	There are no major accessibility support issues known for this rule.

29.3 Test Rule: HTML page title is descriptive

- **WAI-Tools rule number:** 28
- **ACT Rules id:** c4a8a4
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 11.03.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/c4a8a4

Description	This rule checks that the first title in an HTML page describes the topic or purpose of that page.

Accessibility Requirements	WCAG 2.0, 2.4.2
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	This rule applies to the first HTML title element that • is a descendant of the html element of a web page, and • contains children that are text nodes that are not only whitespace. **Note:** The title elements of embedded documents, such as those in iframe, object, or svg elements, are not applicable because those are not web pages according to the definition in WCAG. **Note:** The HTML specification - The title element requires that a document only has one title element, and that it is a child of the head element of a document. However, HTML 5.2 also describes what should happen in case of multiple titles, and titles outside the head element. Because of this, neither of these validation issues causes a conformance problem for WCAG.
Expectation	The target element describes the topic or purpose of the overall content of the document.
Assumptions	This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	This rule assumes that browsers only recognize the first title element if multiple title elements are present in the document. Testing shows that this in general is the case. Therefore the scope of this rule is limited to only checking the first title element in a document.

30 2.4.3 Focus Order (Level A)

30.1 About Success Criteria 2.4.3 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#focus-order

Success Criteria	If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.
Purpose	The intent of this Success Criteria is to ensure that when users navigate sequentially through content, they encounter information in an order that is consistent with the meaning of the content and can be operated from the keyboard. This reduces confusion by letting users form a consistent mental model of the content. There may be different orders that reflect logical relationships in the content. For example, moving through components in a table one row at a time or one column at a time both reflect the logical relationships in the content. Either order may satisfy this Success Criteria. The way that sequential navigation order is determined in Web content is defined by the technology of the content. For example, simple HTML defines sequential navigation via the notion of tabbing order. Dynamic HTML may modify the navigation sequence using scripting along with the addition of a tabindex attribute to allow focus to additional elements. If no scripting or tabindex attributes are used, the navigation order is the order that components appear in the content stream. The focus order may not be identical to the programmatically determined reading order (Success Criteria 1.3.2) as long as the user can still understand and operate the Web page. Since there may be several possible logical reading orders for the content, the focus order may match any of them. However, when the order of a particular presentation differs from the programmatically determined reading order, users of one of these presentations may find it difficult to understand or operate the Web page. Authors should carefully consider all these users as they design their Web pages. It must be possible to understand and use the content of a web page when the user navigates through the website using a keyboard. When keyboard focus follows the visual reading order, the web page will be displayed in the same order as in the case of other navigation methods. For example, a screen reader user interacts with the programmatically determined reading order, while a sighted keyboard user interacts with the visual presentation of the Web page. Care should be taken so that the focus order makes sense to both of these sets of users and does not appear to either of them to jump around randomly. For clarity: 1. Focusable components need to receive focus in an order that preserves meaning and operability only when navigation sequences affect meaning and operability.

	2. In those cases where it is required, there may be more than one order that will preserve meaning and operability. 3. If there is more than one order that preserves meaning and operability, only one of them needs to be provided.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • Usage without hearing
Digdir interpretation and specification of the success criteria	The understanding article for the success criteria indicates that the components on a web page must receive focus in an order that preserves both meaning and operability of the content and functionality. This applies when the focus order affects the meaning of the content and operation. Focus order in this context is the same as keyboard order. Focus order does not have to be identical to reading order as long as the user is able to understand and use the content on the web page. Nevertheless, it is emphasised that a focus order that is very different to the reading order may confuse the user. There may be more than one focus order that preserves the meaning and operability of the web page. It is sufficient for one focus order to comply with the requirement. For content that receives keyboard focus and must be navigated in a specific order. Keyboard navigation order and visual order are the same or keyboard navigation order and visual order are not the same, but it is possible to understand and use the content.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

31 2.4.4 Link Purpose (In Context) (Level A)

31.1 About Success Criteria 2.4.4 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#link-purpose-in-context

Success Criteria	The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.
Purpose	The intent of this Success Criteria is to help users understand the purpose of each link so they can decide whether they want to follow the link. Link text that is as meaningful as possible will aid users who want to choose from this list of links. Meaningful link text also helps those who wish to tab from link to link. Meaningful links help users choose which links to follow without requiring complicated strategies to understand the page.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • Usage without vocal capability
Digdir interpretation and specification of the success criteria	The text of, or associated with, the link describes the purpose of the link. This can be achieved by putting the description of the link in the same sentence, paragraph, list item, or table cell as the link, or in the table header cell for a link in a data table, because these are directly associated with the link itself. Alternatively, authors may choose to use an ARIA technique to associate additional text on the page with the link. The Success Criteria includes an exception for links for which the purpose of the link cannot be determined from the information on the Web page. The success criteria does not apply if the purpose of the link would be ambiguous to users in general. It is difficult to establish objective evaluation criteria to be able to make consistent assessments of whether link targets are ambiguous to users in general. Verifications will involve a high level of subjective judgement. Digdir has not verified this part of the requirement. The wording of the success criteria in the original language refers to “the link text”. It is assumed that the success criteria only applies to links that contain text. For links that contain images, see success criteria 1.1.1.

	There is no requirement for the link target to be described precisely in the link text, but rather that the description of the link target must not be directly incorrect.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **c487ae:** HTML elements with the semantic role of link, that is included in the accessibility tree, have an accessible name that is not empty ("""). • **5effbb:** any HTML or SVG element which has the semantic role of 'link' or a semantic role that inherits from the 'link' role; and is included in the accessibility tree; and has a non-empty (""") accessible name. The accessible name of each target element together with its programmatically determined link context describe the purpose of the link. • **fd3a94:** any set of two or more HTML or SVG elements that have the semantic role of link, or a role that inherits from the link role; and are in the same web page (HTML; and have matching accessible name that are not empty (“”); and have the same programmatically determined link context, the links in each set of these target elements resolve to the same resource or to equivalent resources.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Part of the description of the link in logically related text that provides the context for the link. This is not a comprehensive list of all aspects that may be covered by this success criteria Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

31.2 Test Rule: Link has non-empty accessible name

- **WAI-Tools rule number:** 9
- **ACT Rules id:** c487ae
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 12.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/c487ae

Description	This rule checks that each link has a non-empty accessible name.

Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.9 Link Purpose (Link Only) (Level: AAA)

	• Required for conformance to WCAG 2.0 and above on level AAA and above.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	
	4.1.2 Name, Role, Value (Level: A)
	
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling
Applicability	The rule applies to any HTML element with the semantic role of link that is included in the accessibility tree.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	The rule assumes that all links are user interface components as defined by WCAG 2. When the link role is used on elements that do not behave as links, failing this rule might not mean that the success criteria are failed.
Accessibility Support	There are assistive technologies that do not support using the title attribute for an accessible name, or in which this feature can be disabled. For area elements that have an href attribute, but are not nested inside a map element, there are differences between browsers and assistive technology on if the area is included in the accessibility tree. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of link and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

31.3 Test Rule: Links with identical accessible names and context serve equivalent purpose

- **WAI-Tools rule number:** 49
- **ACT Rules id:** fd3a94
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 25.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/fd3a94

Description	This rule checks that links with identical accessible names and context resolve to the same or equivalent resources.
Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A) • Required for conformance to WCAG 2.0 and later on level A and higher. • Outcome mapping: ○ Any failed outcomes: success criterion is not satisfied.

	○ All passed outcomes: success criterion needs further testing. ○ An inapplicable outcome: success criterion needs further testing. 2.4.9 Link Purpose (Link Only) (Level AAA) • Required for conformance to WCAG 2.0 and later on level AAA. • Outcome mapping: ○ Any failed outcomes: success criterion is not satisfied. ○ All passed outcomes: success criterion needs further testing. ○ An inapplicable outcome: success criterion needs further testing.
Input Aspect	DOM Tree, CSS Styling, Language
Applicability	This rule applies to any set of two or more HTML or SVG elements which • have the semantic role of link, or a role that inherits from the link role; and • are in the same web page (HTML); and • are included in the accessibility tree; and • have matching accessible names that are not empty (""); and • have the same programmatically determined link context. **Note:** The test target for this rule is the full set of link elements that share the same matching accessible name and programmatically determined link context.
Expectation	When followed, the links in each set of target elements resolve to the same resource or to equivalent resources. **Note:** Resolving the links includes potential redirects, if the redirects happen instantly.
Assumptions	• This rule assumes that the purpose of the links with identical accessible names and context would not be ambiguous to users in general, which is the exception mentioned in Success Criterion 2.4.4 Link Purpose (In Context). If the links are ambiguous to users in general, users of assistive technologies are not at a disadvantage when viewing the links, which makes it more of a general user experience concern than an accessibility issue. • This rule assumes that, within the context of the test subject, the description provided by the accessible name of a link can only accurately describe one resource (notably, homonyms alone are not used as link names). Thus, if two or more links have the same accessible name but resolve to different resources, at least one of them does not describe its purpose. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood. • This rule assumes that assistive technologies are exposing all links on the page in the same way no matter which document tree they are in. If an assistive technology requires the user to "enter" an iframe or a shadow tree before exposing its links, then it is possible for two links to have identical name and context but resolve to different resources without failing Success Criterion 2.4.4 Link Purpose (In Context) (if said links are in separate documents or shadow trees)
Accessibility Support	There are no major accessibility support issues known for this rule.

31.4 Test Rule: Link in context is descriptive

- **WAI-Tools rule number:** 51
- **ACT Rules id:** 5effbb
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 29.04.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/5effbb

Description	This rule checks that the accessible name of a link together with its context describe its purpose.

Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A)
	• Required for conformance to WCAG 2.0 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.9 Link Purpose (Link Only) (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	This rule applies to any HTML or SVG element which: • has the semantic role of 'link' or a semantic role that inherits from the 'link' role; and • is included in the accessibility tree; and • has a non-empty ("") accessible name.
Expectation	The accessible name of each target element together with its programmatically determined link context describe the purpose of the link.
Assumptions	• This rule assumes that the purpose of the link is not ambiguous to users in general when seen in context on the web page, which is the exception mentioned in success criterion 2.4.4 Link Purpose (In Context). If the link is ambiguous to users in general, users of assistive technologies are not at a disadvantage when viewing the link out of context which makes it more of a general user experience concern than an accessibility issue. • This rule assumes that all elements with the semantic role of 'link' are used as links.
Accessibility Support	Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of link and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

32 2.4.5 Multiple Ways (Level AA)

32.1 About Success Criteria 2.4.5 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#multiple-ways

Success Criteria	More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process.
Purpose	The intent of this Success Criteria is to make it possible for users to locate content in a manner that best meets their needs. Users may find one technique easier or more comprehensible to use than another. Even small sites should provide users some means of orientation. For a three or four page site, with all pages linked from the home page, it may be sufficient simply to provide links from and to the home page where the links on the home page can also serve as a site map. It must be possible to access web pages, content or functionality on web pages in multiple ways, suited to the needs of the individual. This may involve links, search functionality, site maps, etc.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • Usage without vocal capability
Digdir interpretation and specification of the success criteria	The wording of this success criteria indicates that the requirement applies to web solutions comprising a set of web pages. Even smaller websites that comprise just a few pages must have multiple navigation options. Websites that comprise just one page are not covered by the requirement. Nor does the requirement apply to web pages that are a result of or a step in a process. The various steps that have to be performed in order to buy a product in an online shop is one example of such a process. The results page produced after a search is another example. There must be at least two ways of navigating to content on web pages (on websites with more than one web page)
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

33 2.4.6 Headings and Labels (Level AA)

33.1 About Success Criteria 2.4.6 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#headings-and-labels

Success Criteria	Headings and labels describe topic or purpose.
Purpose	The intent of this Success Criteria is to help users understand what information is contained in Web pages and how that information is organized. When headings are clear and descriptive, users can find the information they seek more easily, and they can understand the relationships between different parts of the content more easily. Descriptive labels help users identify specific components within the content. Labels and headings do not need to be lengthy. A word, or even a single character, may suffice if it provides an appropriate cue to finding and navigating content. This Success Criteria does not require headings or labels. This Success Criteria requires that if headings or labels are provided, they be descriptive. This Success Criteria also does not require that content acting as a heading or label be correctly marked up or identified - this aspect is covered separately by 1.3.1: Info and Relationships. It is possible for content to pass this Success Criteria (providing descriptive content that acts as headings or labels) while failing Success Criteria 1.3.1 (if the headings or labels aren't correctly marked up/identified). Conversely, it is also possible for content to pass Success Criteria 1.3.1 (with headings or labels correctly marked up or identified), while failing this Success Criteria (if those headings or labels are not sufficiently clear or descriptive). Further, in the case of labels, this Success Criteria does not take into consideration whether or not alternative methods of providing an accessible name for form controls and inputs has been used - this aspect is covered separately by 4.1.2: Name, Role and Value. It is possible for controls and inputs to have an appropriate accessible name (e.g. using aria-label="...") and therefore pass Success Criteria 4.1.2, but to still fail this Success Criteria (if the label is not sufficiently clear or descriptive). This success criteria does not require the use of labels; however, it does require that if labels are present, they must be sufficiently clear or descriptive.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • Usage without hearing

	• Usage without vocal capability
Digdir interpretation and specification of the success criteria	This success criteria does not require headings and labels. Rather, this success criteria states that if headings or labels are provided, they must be descriptive Success criteria 2.4.6 and 3.3.2 are closely interlinked: • 2.4.6 relates to both headings and labels. It does not require that they be provided, but states that if they are provided, they must be descriptive. • 3.3.2 requires that instructions or labels are provided that identify form elements (including whether the form element is mandatory) so that users know what input data is expected. If headings are used, they must describe the subject or purpose of the content to which they belong. Labels in forms describe the subject or purpose of the form element.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **cc0f0a**: any programmatic label of an element which has one of the following semantic roles: "*checkbox, combobox, listbox, menuitemcheckbox, menuitemradio, radio, searchbox, slider, spinbutton, switch, textbox*" and where both the element and the programmatic label are visible. Each test target, together with its visual context, describes the purpose of the associated element. • **ffd0e9:** HTML element with the semantic role of heading that is included in the accessibility tree, has a non-empty ("") accessible name.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • If headings are used, they must describe the subject or purpose of the content to which they belong. This is not a comprehensive list of all aspects that may be covered by this success criteria. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

33.2 Test Rule: Form control label is descriptive

- **WAI-Tools rule number:** 29
- **ACT Rules id:** cc0f0a
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 29.04.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/cc0f0a

Description	This rule checks that labels describe the purpose of form field elements.

Accessibility Requirements	2.4.6 Headings and Labels (Level: AA)
	• Required for conformance to WCAG 2.0 and later on level AA and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	This rule applies to any programmatic label of an element which has one of the following semantic roles: • checkbox • combobox • listbox • menuitemcheckbox • menuitemradio • radio • searchbox • slider • spinbutton • switch • textbox and where both the element and the programmatic label are visible. **Note:** The list of applicable semantic roles is derived by taking all the roles from WAI-ARIA Specifications that: • inherit from the abstract input or select role; and • do not have a required context role that itself inherits from one of those roles. **Note:** The option role is not part of the list of applicable roles, because it has a required context role that inherits from the select role. Furthermore, option does not meet the definition of a User interface component. This means that WCAG 2.1 does not require it to have an accessible name. **Note:** Labels in WCAG are not restricted to the label element of HTML and can be any element. This rule is only concerned about actual label elements, and elements that are programmatically marked as labels via the aria-labelledby attribute.
Expectation	Each test target, together with its visual context, describes the purpose of the associated element. **Note:** It is possible for an element to have an accessible name but still have a non-descriptive label element (and even a non-descriptive label). In that case, it would pass Success Criterion 4.1.2: Name, Role and Value but still fail this rule and Success Criterion 2.4.6: Heading and Labels. **Note:** Having a label which is not included in the accessible name is a violation of Success Criterion 2.5.3: Label in Name but not of this rule nor of Success Criterion 2.4.6: Heading and Labels.
Assumptions	This rule assumes that labels are intended for sighted users, and that hiding a visible label from assistive technologies, is a failure of Success Criterion 4.1.2: Name, Role and Value, but not of Success Criterion 2.4.6: Heading and Labels. This rule assumes that the programmatic labels of an element are also part of its visual context.

	This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have one of the applicable semantic roles and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

33.3 Test Rule: Heading has non-empty accessible name

- **WAI-Tools rule number:** 34
- **ACT Rules id:** ffd0e9
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 21.08.2020
- **Last updated in GitHub:** 03.06.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/ffd0e9

Description	This rule checks that each heading has a non-empty accessible name.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.6 Headings and Labels (Level: AA)
	• Required for conformance to WCAG 2.0 and above on level AA and above.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: further testing needed.
	  ○ An inapplicable outcome: further testing needed.

Input Aspect	Accessibility, DOM Tree, CSS Styling
Applicability	This rule applies to any HTML element with the semantic role of heading that is included in the accessibility tree.
Expectation	Each test target has a non-empty ("") accessible name.
Assumptions	There are currently no assumptions.
Accessibility Support	Some assistive technologies may hide headings with empty accessible name from the users. This depends both on the user agent and how the accessible name was computed (the accessible name and description computation is not clear concerning which characters should be trimmed) and of the assistive technology itself. Hence, there are cases where the outcome of this rule is failed, but users of certain assistive technology and browser combinations will not experience an issue. **Note:** Completely empty headings ( ) seem to be consistently ignored by assistive technologies. However, they fail Technique H42: Using h1-h6 to identify headings (by using heading markup for content which is not heading). Moreover, they may be rendered on screen (by breaking flow content, or because of custom

	styling), thus causing concerns for sighted users. Therefore, this rule also fails on these. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of heading and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

34 2.4.7 Focus Visible (Level AA)

34.1 About Success Criteria 2.4.7 and Interpretation

- **Level:** AA
- **WAD:** Yes/No
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#focus-visible

Success Criteria	Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.
Purpose	The purpose of this success criteria is to help a person know which element has the keyboard focus. The purpose of this success criteria is to help a person know which element among multiple elements has the keyboard focus. If there is only one keyboard actionable control on the screen, the success criteria would be met because the visual design presents only one keyboard actionable item. Note that a keyboard focus indicator can take different forms. One common way is a caret within the text field to indicate that the text field has the keyboard focus. Another is a visual change to a button to indicate that the button has the keyboard focus.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition Secondary relationship • Usage without vocal capability
Digdir interpretation and specification of the success criteria	The requirement only applies to content and functionality that receives keyboard focus. The understanding article for the success criteria indicates that if there is only one component on the page that can be used with a keyboard, there does not need to be a visual focus indicator. This is because there is no other content from which this component has to be differentiated. The focus indicator should be highly visible, and the user should easily be able to see which element is in focus. Examples of a visible focus indicator include • a frame, line, or underlining • changed colour of background or text • shading • text cursor (vertical bar) or marking of text in form fields • other form of visual change
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.

Gap analysis suggested by Digdir – For discussion	None.

35 2.4.9 Link Purpose (Link Only) (Level AAA)

35.1 About Success Criteria 2.4.9 and Interpretation

- **Level:** AAA
- **WAD:** No
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#link-purpose-link-only

Success Criteria	A mechanism is available to allow the purpose of each link to be identified from link text alone, except where the purpose of the link would be ambiguous to users in general.
Purpose	The intent of this Success Criteria is to help users understand the purpose of each link in the content, so they can decide whether they want to follow it. Because the purpose of a link can be identified from its link text, links can be understood when they are out of context, such as when the user agent provides a list of all the links on a page. The Success Criteria includes an exception for links for which the purpose of the link cannot be determined from the information on the Web page. The word "mechanism" is used to allow authors to either make all links fully understandable out of context by default or to provide a way to make them this way. This is done because for some pages, making the links all unambiguous by themselves makes the pages easier for some users and harder for others. Providing the ability to make the links unambiguous (by themselves) or not provides both users with disabilities with the ability to use the page in the format that best meets their needs.
User accessibility needs (Functional Performance Statements)	Does not apply. SC 2.4.9 is not a part of WAD.
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, test rules check that • **c487ae:** HTML elements with the semantic role of link, that is included in the accessibility tree, have an accessible name that is not empty (""). • **b20e66:** any set of any two or more HTML or SVG elements which have the semantic role of link, or a role that inherits from the link role; and are in the same web page (HTML); and are included in an accessibility tree; and that have matching accessible names that are not empty (""). When followed, the links in each set of target elements resolve to the same resource or to equivalent resources. Resolving the links includes potential redirects, if the redirects happen instantly. • **5effbb:** any HTML or SVG element which has the semantic role of 'link' or a semantic role that inherits from the 'link' role; and is included in the accessibility tree; and has a non-empty

	("") accessible name. The accessible name of each target element together with its programmatically determined link context describe the purpose of the link.
Gap analysis suggested by Digdir – For discussion	Irrelevant.

35.2 Test Rule: Link has non-empty accessible name

- **WAI-Tools rule number:** 9
- **ACT Rules id:** c487ae
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 12.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/c487ae

Description	This rule checks that each link has a non-empty accessible name.

Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.9 Link Purpose (Link Only) (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling
Applicability	The rule applies to any HTML element with the semantic role of link that is included in the accessibility tree.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	The rule assumes that all links are user interface components as defined by WCAG 2. When the link role is used on elements that do not behave as links, failing this rule might not mean that the success criteria are failed.

Accessibility Support	There are assistive technologies that do not support using the title attribute for an accessible name, or in which this feature can be disabled. For area elements that have an href attribute, but are not nested inside a map element, there are differences between browsers and assistive technology on if the area is included in the accessibility tree. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of link and fail this rule with some technology, but users of other technologies would not experience any accessibility issue.

35.3 Test Rule: Links with identical accessible names have equivalent purpose

- **WAI-Tools rule number:** 38
- **ACT Rules id:** b20e66
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 25.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/b20e66

Description	This rule checks that links with identical accessible names resolve to the same resource or equivalent resources.

Accessibility Requirements	2.4.9 Link Purpose (Link Only) (Level: AAA)
	• Required for conformance to WCAG 2.0 and later on level AAA.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	This rule applies to any set of any two or more HTML or SVG elements which • have the semantic role of link, or a role that inherits from the link role; and • are in the same web page (HTML); and • are included in an accessibility tree; and • that have matching accessible names that are not empty (""). **Note:** The test target for this rule is the full set of link elements that share the same matching accessible name.
Expectation	When followed, the links in each set of target elements resolve to the same resource or to equivalent resources. Resolving the links includes potential redirects, if the redirects happen instantly.
Assumptions	This rule assumes that the purpose of the links with identical accessible names would not be ambiguous to users in general when seen in context on the web page, which is the exception mentioned in Success Criterion 2.4.9 Link Purpose (Link Only). If the links are ambiguous to users in general, users of assistive technologies are not at a disadvantage when viewing the links out of context, e.g.

	on a list of links in a screen reader, which makes it more of a general user experience concern than an accessibility issue. This rule assumes that, within the context of the test subject, the description provided by the accessible name of a link can only accurately describe one resource (notably, homonyms alone are not used as link names). Thus, if two or more links have the same accessible name but resolve to different resources, at least one of them does not describe its purpose. This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	This rule assumes that assistive technologies are exposing all links on the page in the same way no matter which document tree they are in. If an assistive technology requires the user to "enter" an iframe or a shadow tree before exposing its links, then it is possible for two links to have identical name but resolve to different resources without failing Success Criterion 2.4.9 Link Purpose (Link Only) (if said links are in separate documents or shadow trees). Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of link and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

35.4 Test Rule: Links with identical accessible names and context serve equivalent purpose

- **WAI-Tools rule number:** 49
- **ACT Rules id:** fd3a94
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 25.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/fd3a94

Description	This rule checks that links with identical accessible names and context resolve to the same or equivalent resources.

Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	
	2.4.9 Link Purpose (Link Only) (Level AAA)
	• Required for conformance to WCAG 2.0 and later on level AAA.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling, Language

Applicability	This rule applies to any set of two or more HTML or SVG elements which • have the semantic role of link, or a role that inherits from the link role; and • are in the same web page (HTML); and • are included in the accessibility tree; and • have matching accessible names that are not empty (""); and • have the same programmatically determined link context. **Note:** The test target for this rule is the full set of link elements that share the same matching accessible name and programmatically determined link context.
Expectation	When followed, the links in each set of target elements resolve to the same resource or to equivalent resources. **Note:** Resolving the links includes potential redirects, if the redirects happen instantly.
Assumptions	• This rule assumes that the purpose of the links with identical accessible names and context would not be ambiguous to users in general, which is the exception mentioned in Success Criterion 2.4.4 Link Purpose (In Context). If the links are ambiguous to users in general, users of assistive technologies are not at a disadvantage when viewing the links, which makes it more of a general user experience concern than an accessibility issue. • This rule assumes that, within the context of the test subject, the description provided by the accessible name of a link can only accurately describe one resource (notably, homonyms alone are not used as link names). Thus, if two or more links have the same accessible name but resolve to different resources, at least one of them does not describe its purpose. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood. • This rule assumes that assistive technologies are exposing all links on the page in the same way no matter which document tree they are in. If an assistive technology requires the user to "enter" an iframe or a shadow tree before exposing its links, then it is possible for two links to have identical name and context but resolve to different resources without failing Success Criterion 2.4.4 Link Purpose (In Context) (if said links are in separate documents or shadow trees)
Accessibility Support	There are no major accessibility support issues known for this rule.

35.5 Test Rule: Link in context is descriptive

- **WAI-Tools rule number:** 51
- **ACT Rules id:** 5effbb
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 29.04.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/5effbb

Description	This rule checks that the accessible name of a link together with its context describe its purpose.

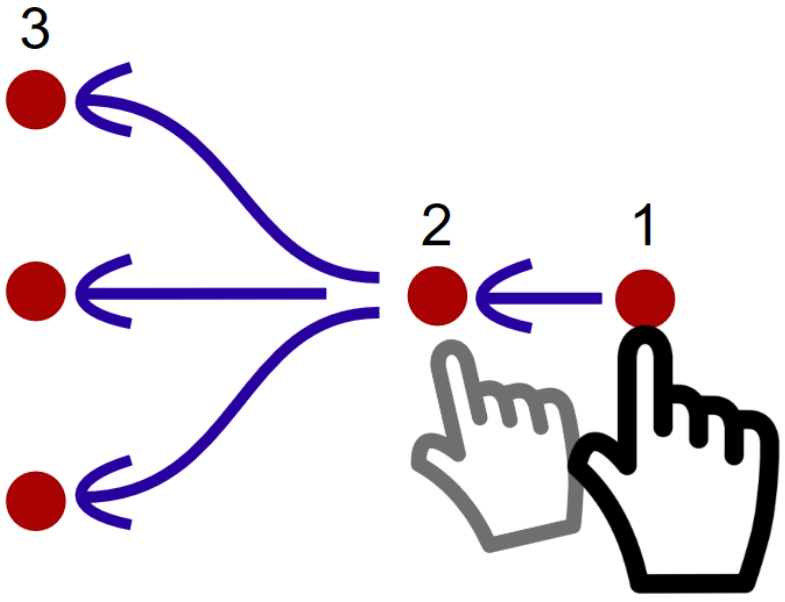
Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A)
	• Required for conformance to WCAG 2.0 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.9 Link Purpose (Link Only) (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling, Language
Applicability	This rule applies to any HTML or SVG element which: • has the semantic role of 'link' or a semantic role that inherits from the 'link' role; and • is included in the accessibility tree; and • has a non-empty ("") accessible name.
Expectation	The accessible name of each target element together with its programmatically determined link context describe the purpose of the link.
Assumptions	• This rule assumes that the purpose of the link is not ambiguous to users in general when seen in context on the web page, which is the exception mentioned in success criterion 2.4.4 Link Purpose (In Context). If the link is ambiguous to users in general, users of assistive technologies are not at a disadvantage when viewing the link out of context which makes it more of a general user experience concern than an accessibility issue. • This rule assumes that all elements with the semantic role of 'link' are used as links.
Accessibility Support	Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of link and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

36 2.5.1 Pointer Gestures (Level A)

36.1 About Success Criteria 2.5.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#pointer-gestures

Success Criteria	All functionality that uses multipoint or path-based gestures for operation can be operated with a single pointer without a path-based gesture, unless a multipoint or path-based gesture is essential. NOTE: This requirement applies to web content that interprets pointer actions (i.e. this does not apply to actions that are required to operate the user agent or assistive technology).
Purpose	The intent of this Success Criteria is to ensure that content can be controlled with a range of pointing devices, abilities, and assistive technologies. Some people cannot perform gestures in a precise manner, or they may use a specialized or adapted input device such as a head pointer, eye-gaze system, or speech-controlled mouse emulator. Some pointing methods lack the capability or accuracy to perform multipoint or path-based gestures. A **path-based gesture** involves an interaction where not just the endpoints matter. If going through an intermediate point (usually near the start of the gesture) also affects its meaning then it is a path-based gesture. The user engages a pointer (starting point), carries out a movement that goes through at least one intermediate-point before disengaging the pointer (end point). The intermediate point defines the gesture as requiring a specific path, even if the complete path is not defined.  **Figure 1A path-based gesture involves starting a pointer movement that goes through at least one intermediate point before the end-point. The end-point may be a continuation, or allow for various movements.**

Examples of path-based gestures include swiping, sliders and carousels dependent on the direction of interaction, and other gestures which trace a prescribed path such as the drawing a specific shape. Such paths may be drawn with a finger or stylus on a touchscreen, graphics tablet, or trackpad, or with a mouse, joystick, or similar pointer device.

Pointer interactions where the movement between the start and end points can go in any direction and are entirely non-prescriptive are not path-based.

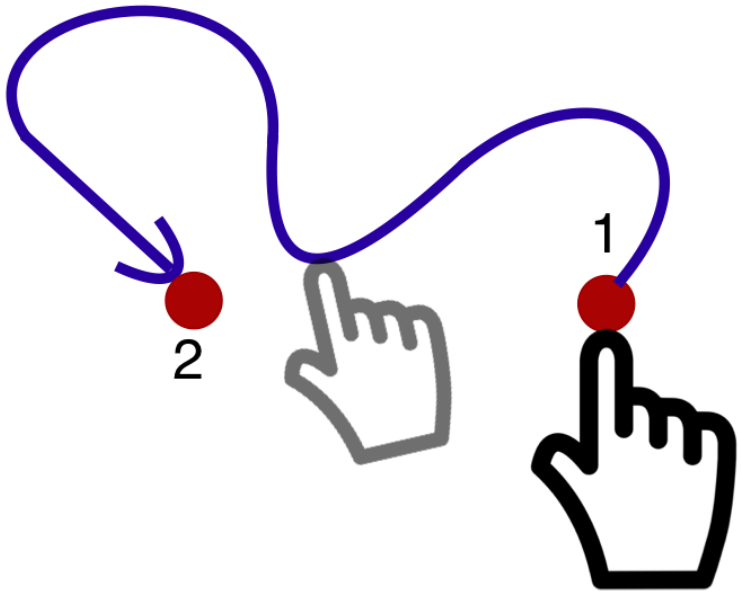


Figure 2A free-form gesture does not require any particular path before the end-point, only the start and (optionally) the end point matter.

NOTE: Any movement of a pointer could be difficult or impossible to use for someone who cannot perform precise movements, therefore alternative forms of interaction are always recommended. This success criteria is scoped to *path-based gestures* as it may be difficult or impossible to provide an alternative for free-form paths.

Examples of **multipoint** gestures include a two-finger pinch zoom, a split tap where one finger rests on the screen and a second finger taps, or a two- or three-finger tap or swipe. Users may find it difficult or impossible to accomplish these if they type and point with a single finger or stick.

Authors must ensure that their content can be operated without multipoint or path-based gestures. Multipoint or path-based gestures can be used so long as the functionality can also be operated by another method, such as a tap, click, double tap, double click, long press, or click & hold.

This Success Criteria applies to gestures in the author-provided content, not gestures defined by the operating system, user agent, or assistive technology. Examples of operating system gestures would be swiping down to see system notifications and gestures for built-in assistive technologies (AT). Examples of user-agent-implemented gestures would be horizontal swiping implemented by browsers for navigating within the page history, or vertical swiping to scroll page content.

There are times when a component requires a path-based gesture for touch screen devices but not with a mouse. Taking an example of a generic slider:

	• **Using a mouse:** If the user clicks on the thumb control of the slider and moves vertically, the slider will respond by moving to the right or left, even if the movement is mostly upwards. There will be no page scrolling as a result of the vertical movement as long as they drag with focus on the slider. Therefore, the slider does not require a path-based gesture with mouse pointer. • **Using a touch-screen:** If the user puts their finger on the thumb control of the slider and moves upwards more than sideways, the slider may not respond because the browser takes control of the swipe and interprets it as a scroll, and will move the page up and down. Moving left or right on the slider thumb engages the slider and then the user can vary their vertical movement. This implementation has the 3-point requirement to work with a finger on a touch screen device so is a path-based gesture. As touch screen devices can apply default gestures it is important to test with them if you are unsure whether a particular component does require a path-based gesture. Browsers on a touch screen device generally provide some default gestures that impact whether a path-based gesture is needed. For example, a web browser on a touch-screen devices might detect a vertical gesture and scroll the page. If a user places their finger on a slider thumb and moves up (to scroll down) that might not activate the slider (depending on implementation). If the user moves horizontally first then the slider could capture that gesture and ignore vertical movement, resulting in a path-based gesture. If you include touch-screen devices as accessibility supported then these types of interaction need testing with a touch screen as using a mouse in a similar way would not trigger the same browser behavior. This Success Criteria does not require all functionality to be available through pointing devices, but if it is available to pointer devices then it should not require path-based gestures. While content authors generally need to provide keyboard commands or other non-pointer mechanisms that perform actions equivalent to complex gestures (see Success Criteria 2.1.1 Keyboard), this is not sufficient to conform to this Success Criteria. That is because some users rely entirely on pointing devices, or find simple pointer inputs much easier than alternatives. For example, a user relying on a head-pointer would find clicking a control to be much more convenient than activating an on-screen keyboard to emulate a keyboard shortcut, and a person who has difficulty memorizing a series of keys (or gestures) may find it much easier to simply click on a labeled control. Therefore, if one or more pointer-based mechanisms are supported, then their benefits should be afforded to users through simple, single-point actions alone. An exception is made for functionality that is inherently and necessarily based on complex paths or multipoint gestures. For example, entering your signature may be inherently path-based (although acknowledging something or confirming your identity need not be). Gestures that involve dragging in any direction are not in scope for this SC because only the start and end points matter in a dragging operation. However, such gestures do require fine motor control. Authors are encouraged to provide non-dragging methods, for instance, a drag and drop operation could also be achieved by selecting an item (with a tap or keyboard interaction) and then selecting its destination as a second step.
User accessibility needs (Functional	Primary relationship

Performance Statements)	• Usage with limited manipulation or strength • Usage with limited reach • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

37 2.5.2 Pointer Cancellation (Level A)

37.1 About Success Criteria 2.5.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 29.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#pointer-cancellation

Success Criteria	For functionality that can be operated using a single pointer, at least one of the following is true: • **No Down-Event:** The down-event of the pointer is not used to execute any part of the function; • **Abort or Undo:** Completion of the function is on the up-event, and a mechanism is available to abort the function before completion or to undo the function after completion; • **Up Reversal:** The up-event reverses any outcome of the preceding down-event; • **Essential:** Completing the function on the down-event is essential. **NOTE:** • Functions that emulate a keyboard or numeric keypad key press are considered essential. • This requirement applies to web content that interprets pointer actions (i.e. this does not apply to actions that are required to operate the user agent or assistive technology).
Purpose	The intent of this success criteria is to make it easier for users to prevent accidental or erroneous pointer input. People with various disabilities can inadvertently initiate touch or mouse events with unwanted results. Each of the following subsections roughly aligns with the bullets of this Success Criteria, and outlines a means of allowing users to cancel pointer operations. **Up-Event activation or completion:** The most accessible way to incorporate pointer cancellation is to make activation occur on the up-event. Up-event activation refers to the activation of a target when the pointer is released. In a touchscreen interaction, when the finger touches a target, the up-event activation only occurs when the finger is lifted while still being within the target boundary. Similarly in mouse interaction, the up-event occurs when the mouse button is released while the cursor is still within the boundary of the initial target set when the mouse button was pressed. Authors can reduce the problem of users inadvertently triggering an action by using generic platform activation/click events that activate functionality on the up-event. For example, the click event in JavaScript triggers on release of the primary mouse button, and is an example of an implicit up-event. Despite its name, the click event is device-independent and also works for touch and keyboard interaction. The preference for up-events is implicit in the Success Criteria wording of the first bullet: The down-event of the pointer is not used to execute

	any part of the function. Authors meet the first bullet by using only the up-event. **Up-Event Abort or Undo:** Where the interaction is equivalent to a simple "click", up-event activation has a built-in ability to cancel. There is a distinction between when someone touches a screen and when they remove their finger. Similarly, in mouse interaction, there is a difference between pressing and releasing the mouse button. When activation occurs only as the pointer is released, users have the opportunity to Abort (cancel) the activation. Users who have difficulty accurately using a mouse or touchscreen benefit greatly from this basic behaviour. They normally receive visual feedback when an item is pressed. If they discover they have selected the wrong item, they can cancel the action by moving their pointer or finger away from the target before releasing. For more complex interactions, such as drag and drop, the down- and up-events may initiate and end a series of actions to complete a process. For example, with drag and drop, the item may be: 1. selected with a press (down-event), 2. moved to a new location, while still being depressed, and 3. released (up-event) to conclude the drop action. In such a complex action, the need for an Abort or Undo function increases. Designers may elect to confirm the move through something like a confirmation dialog or an undo button, giving the user the ability to Undo the process just completed. Alternatively, the ability to Abort the action can be acheived if, before completing step 3, the user returns the selected item to its original location and concludes the process there. If other parts of the screen disallow a move, the user can conclude the drag and drop there, effectively nullifying the operation. **Up Reversal:** In other interactions, the down-event may trigger a behaviour which can be reversed when the up-event concludes. Examples of this include press-and-hold actions such as where a transient popup appears (or a video plays) when the user presses on an object (down-event), but the popup (or video) disappears as soon as the user releases the pointer (up-event). Since the up-event reverses the preceding down event, the user is returned to their prior point, and has effectively cancelled the operation. **Down-Event:** Completing the function on the down-event is only permitted when it is essential that the up-event not be used. The most prevalent essential down-event activation occurs in keyboard emulation. On a physical keyboard, keys by default activate on the down-event -- a letter appears when the key is pressed. If a software keyboard emulator tried to override this expected behaviour by making letters appear when the key is released, the behaviour would be unexpected and would adversely affect expected interaction. Note that a keyboard has a built-in Backspace or Delete button, which effectively provides an Undo option. Undo is not a requirement of the down-event Essential exception; however, providing an easy way for users to undo any action is a recommended practice (and may be a functional necessity), even where it is not a requirement of this Success Criteria. Other examples where the timing of an activation is essential and requires the down-event would be:

	• An activity that emulates a physical on-press trigger, such as when playing an on-screen piano keyboard. Activation on the up-event would significantly alter the desired behaviour. • A program for shooting skeets where waiting for the "up" event would invalidate the precise timing necessary for the activation.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

38 2.5.3 Label in Name (Level A)

38.1 About Success Criteria 2.5.3 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#label-in-name

Success Criteria	For user interface components with labels that include text or images of text, the name contains the text that is presented visually. **NOTE:** A best practice is to have the text of the label at the start of the name.
Purpose	The intent of this Success Criteria is to ensure that the words which visually label a component are also the words associated with the component programmatically. This helps ensure that people with disabilities can rely on visible labels as a means to interact with the components. Most controls are accompanied by a visible text label. Those same controls have a programmatic name, also known as the Accessible Name. Users typically have a much better experience if the words and characters in the visible label of a control match or are contained within the accessible name. When these match, speech-input users (i.e., users of speech recognition applications) can navigate by speaking the visible text labels of components, such as menus, links, and buttons, that appear on the screen. Sighted users who use text-to-speech (e.g., screen readers) will also have a better experience if the text they hear matches the text they see on the screen. Note that where a visible text label does not exist for a component, this Success Criteria does not apply to that component. Where text labels exist and are properly linked to the user interface components through established authoring practices, the label and name will normally match. When they don't match, speech-input users who attempt to use the visible text label as a means of navigation or selection (e.g., "move to Password") will be unsuccessful. The speech-based navigation fails because the visible label spoken by the users does not match (or is not part of) the accessible name that is enabled as a speech-input command. In addition, when the accessible name is different from the visible label, it may function as a hidden command that can be accidentally activated by speech-input users. Mismatches between visible labels and programmatic names for controls are even more of an issue for speech-input and text-to-speech users who also have cognitive challenges. Mismatches create an extra cognitive load for speech-input users, who must remember to say a speech command that is different from the visible label they see on a control. It also creates extra cognitive load for a text-to-speech user to absorb and understand speech output that does not match the visible label. In order for the label text and accessible name to be matched, it is first necessary to determine which text on the screen should be considered a label for any given control. There are often multiple text strings in a user interface that may be relevant to a control. However, there are

	reasons why it is best to conservatively interpret the label as being only the text in close proximity. Conventionally the label for user interface components is the adjacent text string. The typical positioning for left to right languages is: immediately to the left of comboboxes, dropdown lists, text inputs, and other widgets (or in the absence of left-side labels, immediately above and aligned with the left edge of each input) • immediately to the right of checkboxes and radio buttons • inside buttons and tabs or immediately below icons serving as buttons • The rationale for some of these conventions is explained in G162: Positioning labels to maximize predictability of relationships. It is important to bias towards treating only the adjacent text as a label because liberal interpretations of what constitutes a text label can jeopardize the value of this Success Criteria (SC) by lessening predictability. Isolating the label to the single string in close proximity to the component makes it easier for developers, testers, and end users to identify the label targeted for evaluation in this SC. Predictable interpretation of labeling allows users of speech recognition technologies to interact with the element via its conventionally positioned label, and allows users of screen reading technologies to enjoy consistency between the nearby visible label and the announced name of the component. Note that placeholder text within an input field is not considered an appropriate means of providing a label. The HTML5 specification states The placeholder attribute should not be used as an alternative to a <label>. However, it is worth noting that "label" in that HTML5 statement is in code brackets and links to the label element. For the purposes of this Label in Name Success Criteria, "label" is not used in such a programmatic sense but is simply referring to a text string in close visual proximity to a component. As such, in the absence of any other nearby text string (as described in the preceding list), if an input contains placeholder text, such text may be a candidate for Label in Name. This is supported both through the accessible name calculation (discussed later) and from the practical sense that where a visible label is not otherwise provided, it is likely that a speech-input user may attempt to use the placeholder text value as a means of interacting with the input.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited manipulation or strength • Usage with limited reach Secondary relationship • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.

Gap analysis suggested by Digdir – For discussion	None.

39 2.5.4 Motion Actuation (Level A)

39.1 About Success Criteria 2.5.4 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#motion-actuation

Success Criteria	Functionality that can be operated by device motion or user motion can also be operated by user interface components and responding to the motion can be disabled to prevent accidental actuation, except when • Supported Interface: The motion is used to operate functionality through an accessibility supported interface; • Essential: The motion is essential for the function and doing so would invalidate the activity.
Purpose	The intent of this success criteria is to ensure that functions triggered by moving a device (for example, shaking or tilting) or by gesturing towards the device (so that sensors like a camera can pick up and interpret the gesturing), can also be operated by more conventional user interface components. **NOTE:** This criterion concerns input through sensors which respond directly to motions such as gesturing towards, tilting or shaking a device. It does not cover the movement of users through space as registered by geolocation sensors or beacons, or events observed by the device other than intentional gesturing by the user. It also does not cover incidental motion associated with operating a keyboard, pointer, or assistive technology. Devices often have sensors that can act as inputs, such as accelerometer and gyroscope sensors on a phone or tablet device. These sensors can allow the user to control something by simply changing the orientation or moving the device in particular ways. In other situations, web content can interpret user gestures via the camera or other sensors to actuate functions. For example, shaking the device might issue an "Undo" command, or a gentle hand wave might be used to move forward or backward in a sequence of pages. Some users with disabilities are not able to operate these device sensors (either not at all, or not precisely enough) because the device is on a fixed mount (perhaps a wheelchair) or due to motor impairments. Therefore, functionality offered through motion must also be available by another mechanism. In addition, some users may accidentally activate sensors due to tremors or other motor impairments. The user must have the ability to turn off motion actuation to prevent such accidental triggering of functions. Applications may be able to meet this requirement by supporting operating system settings which allow the user to disable motion detection at the system level. There is an exception where motion is essential for the function or not using motions or gestures would invalidate the activity. Some applications are specifically created to use device sensor data. Examples of content that are exempt from this requirement include a pedometer that relies on device motion to count steps.

User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited manipulation or strength • Usage with limited reach Secondary relationship • Usage without vision • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **c249d5:** An HTML document with an associated window object that has an event listener list with one or more event listeners for device orientation events or device motion events where each of these registered events does not cause changes to the content of web page with in a 10 minutes time span of event firing; or there is at least one set of instruments is available, where each instrument is on the same web page of the register event or can be found in a clearly labelled location from that web page. The set of instruments can be used to prevent changes to the content of the web page resulting from the event within a 10 minutes time span of the event firing. • **7677a9:** Any HTML document with an associated window object that has an event listener list with one or more event listeners for device orientation events or device motion events where each of these registered events in the test target does not cause changes to the content of the web page; or there is at least one set of instruments available, where each instrument is in the same website of the registered event or can be found in a clearly labelled location from that web page, causing the same changes in content as the event.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

39.2 Test Rule: Device motion based changes to the content can be disabled

- **WAI-Tools rule number:** 54
- **ACT Rules id:** c249d5
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 15.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/c249d5

Description	This rule checks that it is possible to disable any changes to the content of the web page resulting from device motion based events.

Accessibility Requirements	2.5.4 Motion Actuation (Level: A)
	• Required for conformance to WCAG 2.1 on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling, Accessibility Tree
Applicability	The rule applies to an HTML document with an associated Window object that has an event listener list with one or more event listeners for device orientation events or device motion events.
Expectation	For each registered device orientation event or device motion event in the test target one of the following is true: • no changes: The registered event does not cause changes to the content of the web page within a 10 minutes time span of the event firing; or • disabled: There is at least one set of instruments, where each instrument is in the same web page of the registered event or can be found in a clearly labeled location from that web page. The set of instruments can be used to prevent changes to the content of the web page resulting from the event within a 10 minutes time span of the event firing. **Note:** The 10 minutes time span is an arbitrary limit which is not included in WCAG. Results that happen after this period will not fail this rule but may nonetheless fail Success Criterion 2.5.4: Motion Actuation. The accessibility problem tends to be less severe for longer time periods, and without a time limit, testing this rule consistently would be impractical.
Assumptions	• The motion to operate the device is not used through an accessibility supported interface, which is listed as a valid exception to Success Criterion 2.5.4: Motion Actuation. • The motion is not essential for the functionality it triggers, which is listed as a valid exception to Success Criterion 2.5.4: Motion Actuation. • The event listeners listening to device motion events trigger a functionality in the web page. If they do not trigger any such functionality failing this rule might not be a failure of the success criterion. • If there are ways to disable the device motion based functionality that do not require the user to interact with the web page (e.g. a setting at the operating system level), failing this rule might not be a failure of the success criterion. • This rule assumes that there are no changes in the content of the web page caused by another event. If this is not the case, changes may be attributed to the wrong event and the rule may fail while Success Criterion 2.5.4: Motion Actuation is still satisfied. • This rule assumes that the changes happen within a 1 minute time span after the event firing and therefore the comparison between the page before and after the event firing can be made at any time after that time span elapses. If there are changes after this time span, they may not be detected as changes in content and the rule may pass but Success Criterion 2.5.4: Motion Actuation is not satisfied. The arbitrary 1 minute time span, selected so that testing this rule would not be impractical, is not included in WCAG. • After being disabled, the event remains disabled until being re-enabled again. If the event is re-enabled through other non-user controlled means

	(e.g. a timeout) then this rule may pass while Success Criterion 2.5.4: Motion Actuation is not satisfied.
Accessibility Support	There are no major accessibility support issues known for this rule.

39.3 Test Rule: Device motion based changes to the content can also be created from the user interface

- **WAI-Tools rule number:** 67
- **ACT Rules id:** 7677a9
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/7677a9

Description	This rule checks that changes to the content of a web page that result from device motion events can also be caused by user interface components.

Accessibility Requirements	2.5.4 Motion Actuation (Level: A)
	• Required for conformance to WCAG 2.1 on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling, Accessibility Tree
Applicability	The rule applies to any HTML document with an associated Window object that has an event listener list with one or more event listeners for device orientation events or device motion events.
Expectation	For each registered device orientation event or device motion event in the test target one of the following is true: • no changes: The registered event does not cause changes to the content of the web page; or • same result: There is at least one set of instruments, where each instrument is in the same web page of the registered event or can be found in a clearly labeled location from that web page, causing the same changes in content as the event.
Assumptions	• The motion to operate the device is not used through an accessibility supported interface, which is listed as a valid exception to Success Criterion 2.5.4: Motion Actuation. • The motion is not essential for the functionality it triggers, which is listed as a valid exception to Success Criterion 2.5.4: Motion Actuation. • This rule assumes that there are no changes in the content of the web page caused by another event. If this is not the case, changes may be attributed to the wrong event and the rule may fail while Success Criterion 2.5.4: Motion Actuation is still satisfied. • This rule assumes that the changes happen within a 1 minute time span after the event firing and therefore the comparison between the page before and after the event firing can be made at any time after that time

	span elapses. If there are changes after this time span, they may not be detected as changes in content and the rule may pass but Success Criterion 2.5.4: Motion Actuation is not satisfied. The arbitrary 1 minute time span, selected so that testing this rule would not be impractical, is not included in WCAG.
Accessibility Support	There are no major accessibility support issues known for this rule.

40 3.1.1 Language of page (Level A)

40.1 About Success Criteria 3.1.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 15.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#language-of-page

Success Criteria	The default human language of each Web page can be programmatically determined.
Purpose	The default language of the page must be declared in the code. The purpose is to ensure that text and other linguistic content is presented correctly. Both assistive technologies and user agents can reproduce text more accurately when the language on the web page is identified. Similarly, screen readers will make text accessible with correct pronunciation, and captions in media players will be correct. This makes it easier to understand the content on web pages.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision • Usage without hearing • Usage with limited hearing • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The understanding article for the success criteria specifies that the default language is the language used for the majority of the content. If multiple languages are used in equal quantities, the default language is the language used first.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check the following for any document element if it is an html element that is in a top-level browsing context and has a node document with a content type of text/html: • **b5c3f8:** HTML page has a lang attribute that is neither empty ("") nor only ASCII whitespace • **5b7ae0:** The values of the primary language subtags, if any exist, for the lang and xml:lang attributes are the same. • **bf051a:** the lang attribute has a valid language tag (A text string following the syntax as defined in BCP 47, with a primary language subtag exists in the language subtag registry with a Type field whose field-body value is language). • **ucwvc8:** The primary language of the valid language tag of a page matches the default page language.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. Aspects of the ACT rules that go beyond the scope of the success criteria:

	• Partially: If both the lang and the xml:lang attributes are used, the values of the primary language subtag (characters before the first dash) are the same.

40.2 Test Rule: HTML page has lang attribute

- **WAI-Tools rule number:** 6
- **ACT Rules id:** b5c3f8
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 25.08.2020
- **Last updated in GitHub:** 19.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/b5c3f8

Description	This rule checks that an HTML page has a non-empty lang attribute.

Accessibility Requirements	3.1.1 Language of Page (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	This rule applies to any document element if it is an html element that: • is in a top-level browsing context; and • has a node document with a content type of text/html. **Note:** html elements within iframe and object elements are not applicable as iframe and object elements create nested browsing contexts. However, as these elements are meant to provide a layer of isolation, the declared language of their parent browsing context will likely not be inherited, making it possible for empty lang attributes in nested browsing contexts to also cause accessibility issues.
Expectation	Each test target has a lang attribute that is neither empty ("") nor only ASCII whitespace.
Assumptions	The language of the page can be set by other methods than the lang attribute, for example using HTTP headers or the meta element. These methods are not supported by all assistive technologies. This rule assumes that these other methods are insufficient to satisfying Success Criterion 3.1.1: Language of Page.
Accessibility Support	There are no major accessibility support issues known for this rule.

40.3 Test Rule: HTML page lang and xml:lang attributes have matching values

- **WAI-Tools rule number:** 7
- **ACT Rules id:** 5b7ae0
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 25.08.2020
- **Last updated in GitHub:** 03.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/5b7ae0

Description	This rule checks that both lang and xml:lang attributes on the root element of a non-embedded HTML page, have the same primary language subtag.

Accessibility Requirements	3.1.1 Language of Page (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	This rule applies to any document element if it is an html element that: • is in a top-level browsing context; and • has a node document with a content type of text/html; and • has a lang attribute that has a valid language subtag; and • has a non-empty xml:lang attribute.
Expectation	For each test target, the values of the primary language subtags, if any exist, for the lang and xml:lang attributes are the same.
Assumptions	• The language of the page can be set by other methods than the lang attribute, for example using HTTP headers or the meta element. These methods are not supported by all assistive technologies. This rule assumes that these other methods are insufficient to satisfying Success Criterion 3.1.1: Language of Page. • This rule assumes that user agents and assistive technologies can programmatically determine valid language tags even if these do not conform to the BCP 47 syntax. • This rule assumes that grandfathered tags are not used as these will not be recognized as valid language tags. • The rule assumes that having lang and xml:lang attributes with matching primary language subtags but non-matching language tags overall, will not cause accessibility issues. This is not necessarily the case for all languages. One notable case is the language tags for Cantonese (zh-yue) and Mandarin (zh-cmn) where the primary language subtags match, but the extended language subtags don't. Such a case would not fail this rule, but could lead to accessibility issues.
Accessibility Support	Since most assistive technologies will consistently use lang over xml:lang when both are used, violation of this rule may not necessarily be a violation of WCAG 2. Only when there are inconsistencies between assistive technologies as to which

	attribute is used to determine the language does this lead to a violation of SC 3.1.1.

40.4 Test Rule: HTML page lang attribute has valid language tag

- **WAI-Tools rule number:** 14
- **ACT Rules id:** bf051a
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 25.08.2020
- **Last updated in GitHub:** 19.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/bf051a

Description	This rule checks that the lang attribute of the root element of a non-embedded HTML page has a language tag with a known primary language subtag.

Accessibility Requirements	3.1.1 Language of Page (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	This rule applies to any document element if it is an html element that: • has a lang attribute that is neither empty ("") nor only ASCII whitespace; and • is in a top-level browsing context; and • has a node document with a content type of text/html.
Expectation	For each test target, the lang attribute has a valid language tag.
Assumptions	• The language of the page can be set by other methods than the lang attribute, for example using HTTP headers or the meta element. These methods are not supported by all assistive technologies. This rule assumes that these other methods are insufficient to satisfying Success Criterion 3.1.1: Language of Page. • This rule assumes that user agents and assistive technologies can programmatically determine valid language tags even if these do not conform to the BCP 47 syntax. • This rule assumes that grandfathered tags are not used as these will not be recognized as valid language tags.
Accessibility Support	There are no major accessibility support issues known for this rule.

40.5 Test Rule: HTML page language subtag matches default language

- **WAI-Tools rule number:** 65
- **ACT Rules id:** ucwvc8
- **Status in WAI-Tools:** Completed

- **Last updated in this report:** 29.09.2020
- **Last updated in GitHub:** 07.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/ucwvc8

Description	This rule checks that the primary language subtag of the page language matches the default language of the page

Accessibility Requirements	3.1.1 Language of Page (Level A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion is satisfied.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, Accessibility Tree, CSS Styling, Language
Applicability	This rule applies to any document element if it is an html element for which all of the following are true: • The document element has a lang attribute with a value that is a valid language tag; and • The document element is in a top-level browsing context; and • The document element has a content type of text/html; and • The document element has a defined default page language.
Expectation	For each test target, the primary language of the valid language tag matches the default page language of the test target.
Assumptions	This rule assumes that the default human language of a page, as described in WCAG 2, can be determined by counting the number of words used in each language. If the default language needs to be derived in some other way (such as frequency analysis, mutual information-based distance, …), this rule may fail while Success Criterion 3.1.1: Language of Page is still satisfied. The language of the page can be set by other methods than the lang attribute, for example using HTTP headers or the meta element. These methods are not supported by all assistive technologies. This rule assumes that these other methods are insufficient to satisfying Success Criterion 3.1.1: Language of Page. This rule assumes that user agents and assistive technologies can programmatically determine valid language tags even if these do not conform to the BCP 47 syntax. This rule assumes that grandfathered tags are not used as these will not be recognized as valid language tags. This rule assumes that iframe title elements are not exposed to assistive technologies and so does not consider them as part of the default page language.
Accessibility Support	There are no major accessibility support issues known for this rule.

41 3.1.2 Language of Parts (Level AA)

41.1 About Success Criteria 3.1.2 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#language-of-parts

Success Criteria	The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.
Purpose	The intent of this Success Criteria is to ensure that user agents can correctly present content written in multiple languages. This makes it possible for user agents and assistive technologies to present content according to the presentation and pronunciation rules for that language.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision • Usage without hearing • Usage with limited hearing • Usage with limited cognition
Digdir interpretation and specification of the success criteria	For sentences or sections written in a language that is different to the default language for the web page, the following applies: • The component containing the sentence or section has a valid language code in the HTML code. • The language code corresponds to the language in which the sentence or section is written. The success criteria does not apply to words or expressions of a different linguistic origin to the surrounding text when the word or expression is commonly used in the language in which the page is written. Individual words are deemed to be part of the language of the text in which the word is included, unless it is clearly evident that the language has been changed deliberately. If there is any doubt as to whether such words and expressions have been used deliberately, a check should be carried out to find out whether the word or expression is pronounced in the same way as in the language in which the rest of the text is written.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **de46e4:**The value of lang attribute is a valid language subtag (A text string following the syntax as defined in BCP 47, with a primary language subtag from the language subtag registry) for any HTML element that contains text, is a descendant in the flat tree of a body element, and has a lang attribute that is not empty (""),.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • The value of the lang tag corresponds to the language used in the specified part of the web page. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

41.2 Test Rule: Element with lang attribute has valid language tag

- **WAI-Tools rule number:** 13
- **ACT Rules id:** de46e4
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/de46e4

Description	This rule checks that a non-empty lang attribute of an element in the page body has a language tag with a known primary language subtag.

Accessibility Requirements	3.1.2 Language of Parts (Level: AA)
	• Required for conformance to WCAG 2.0 and above on level AA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	This rule applies to any HTML element that: • has a node document with a content type of text/html; and • is a descendant in the flat tree of a body element; and • has a lang attribute that is not empty ("").
Expectation	For each test target, the value of the lang attribute is a valid language tag.
Assumptions	The lang attribute is assumed to be used to indicate the language of a section of the content. If the lang attribute is used for something else (for example to indicate a code element contains CSS), the content may still conform to WCAG despite failing this rule. This rule assumes that user agents and assistive technologies can programmatically determine valid language tags even if these do not conform to the BCP 47 syntax. This rule assumes that grandfathered tags are not used as these will not be recognized as valid language tags. The language of the page can be set by other methods than the lang attribute, for example using HTTP headers or the meta element. These methods are not

	supported by all assistive technologies. This rule assumes that these other methods are insufficient to satisfying Success Criterion 3.1.1: Language of Page.
Accessibility Support	There are differences in how assistive technologies handle unknown and invalid language tags. Some will default to the language of the page, whereas others will default to the closest ancestor with a valid lang attribute.

42 3.2.1 On Focus (Level A)

42.1 About Success Criteria 3.2.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#on-focus

Success Criteria	When any user interface component receives focus, it does not initiate a change of context.
Purpose	The intent of this Success Criteria is to ensure that functionality is predictable as visitors navigate their way through a document. Any component that is able to trigger an event when it receives focus must not change the context. Examples of changing context when a component receives focus include, but are not limited to: • forms submitted automatically when a component receives focus; • new windows launched when a component receives focus; • focus is changed to another component when that component receives focus; Focus may be moved to a control either via the keyboard (e.g. tabbing to a control) or the mouse (e.g. clicking on a text field). Moving the mouse over a control does not move the focus unless scripting implements this behavior. Note that for some types of controls, clicking on a control may also activate the control (e.g. button), which may, in turn, initiate a change in context. It must be predictable to the user what will happen when navigating a web page using a mouse or keyboard. The context must only be changed when the user actively executes an action, e.g. clicks on a link with a mouse or keyboard or presses a button. **NOTE:** What is meant by "component" here is also sometimes called "user interface element" or "user interface component".
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The requirement means that context changes of the types listed below must not be the result of a content component receiving focus. The WCAG glossary provides an extended explanation of what a context change is. Context change includes • changing the user agent (content is opened in another program) • changing the presentation frame • changing the focus and content in a way that changes the meaning of the web page This requirement applies to both mouse and keyboard navigation. The Authority has restricted the verification to testing with a keyboard, as

	this has been deemed the most effective way of assessing predictable navigation. Standard implementation in HTML is such that error situations relating to focus that occur during keyboard navigation will generally occur when navigating with a mouse as well. Therefore, we have chosen to carry out the test with a keyboard for the purposes of efficiency.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

43 3.2.2 On Input (Level A)

43.1 About Success Criteria 3.2.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#on-input

Success Criteria	Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component.
Purpose	The intent of this Success Criteria is to ensure that entering data or selecting a form control has predictable effects. Changing the setting of any user interface component is changing some aspect in the control that will persist when the user is no longer interacting with it. So checking a checkbox, entering text into a text field, or changing the selected option in a list control changes its setting, but activating a link or a button does not. Changes in context can confuse users who do not easily perceive the change or are easily distracted by changes. Changes of context are appropriate only when it is clear that such a change will happen in response to the user's action. The purpose of this requirement is that changes in user interface components do not result in context changes. Consequences of making changes to user interface components must be predictable and known to the user. The user must be notified when such actions will involve changing the context. **NOTE:** • This Success Criteria covers changes in context due to changing the setting of a control. Clicking on links or tabs in a tab control is activating the control, not changing the setting of that control. • What is meant by "component" and "user interface component" here is also sometimes called "user interface element".
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited manipulation or strength • Usage with limited cognition
Digdir interpretation and specification of the success criteria	This requirement is that changes to settings in user interface components must not result in automatic context changes of the types listed below. The WCAG glossary provides an extended explanation of what a context change is. Context change includes • changing the user agent (content is opened in another program) • changing the presentation frame • changing the focus and content in a way that changes the meaning of the web page

	If such context changes take place, the user must be advised before the component is used. Context changes that users must be advised of (and which are therefore relevant to this success criteria) are • opening new content in either the same tab, window or program or a new one • keyboard focus is moved to a different component on the web page • significant change in the meaning of the web page • form submission This requirement applies to both mouse and keyboard navigation. The Authority has restricted the verification to testing with a keyboard, as this has been deemed the most effective way of assessing predictable navigation. Standard implementation in HTML is such that error situations relating to the setting of user interface components that are displayed during keyboard navigation will generally occur when navigating with a mouse as well.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

44 3.2.3 Consistent Navigation (Level AA)

44.1 About Success Criteria 3.2.3 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#consistent-navigation

Success Criteria	Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.
Purpose	The intent of this Success Criteria is to encourage the use of consistent presentation and layout for users who interact with repeated content within a set of Web pages and need to locate specific information or functionality more than once. Individuals with low vision who use screen magnification to display a small portion of the screen at a time often use visual cues and page boundaries to quickly locate repeated content. Presenting repeated content in the same order is also important for visual users who use spatial memory or visual cues within the design to locate repeated content. It is important to note that the use of the phrase "same order" in this section is not meant to imply that subnavigation menus cannot be used or that blocks of secondary navigation or page structure cannot be used. Instead, this Success Criteria is intended to assist users who interact with repeated content across Web pages to be able to predict the location of the content they are looking for and find it more quickly when they encounter it again. The purpose of this success criteria is to encourage the use of consistent presentation of components for navigation on the website. Components that are repeated on multiple pages on a website must be presented consistently so that users can orient themselves easily. Clickable logos, search fields, menus, etc. are examples of navigational components. Users may initiate a change in the order by using adaptive user agents or by setting preferences so that the information is presented in a way that is most useful to them.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited cognition
Digdir interpretation and specification of the success criteria	Navigation mechanisms that are available on multiple pages on the website must appear in the same relative order. Nevertheless, the requirement for consistent presentation (the same relative order) does not prevent components being inserted into or removed from the original order. In navigation menus that can be expanded, it is possible to add an extra level of detail or a secondary navigation element in the reading order, for example.

Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

45 3.2.4 Consistent Identification (Level AA)

45.1 About Success Criteria 3.2.4 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#consistent-identification

Success Criteria	Components that have the same functionality within a set of Web pages are identified consistently.
Purpose	The intent of this Success Criteria is to ensure consistent identification of functional components that appear repeatedly within a set of Web pages. A strategy that people who use screen readers use when operating a Web site is to rely heavily on their familiarity with functions that may appear on different Web pages. If identical functions have different labels (or, more generally, a different accessible name) on different Web pages, the site will be considerably more difficult to use. It may also be confusing and increase the cognitive load for people with cognitive limitations. Therefore, consistent labeling will help. This consistency extends to the text alternatives. If icons or other non-text items have the same functionality, then their text alternatives should be consistent as well. If there are two components on a web page that both have the same functionality as a component on another page in a set of web pages, then all 3 must be consistent. Hence the two on the same page will be consistent. While it is desirable and best practice always to be consistent within a single web page, 3.2.4 only addresses consistency within a set of web pages where something is repeated on more than one page in the set. Components that have the same functionality that are repeated in multiple locations on a website must have consistent identification in the code. The search function is one example of the component covered by this success criteria.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage with limited vision • Usage with limited cognition Secondary relationship • Usage without vision
Digdir interpretation and specification of the success criteria	The requirement for consistent identification applies to components with the same functionality that are repeated: • on multiple pages on a website • several times on the same page, and also on other pages The requirement does not apply to components that are only present on one web page, even if they occur several times on a single page. The fact that components with the same functionality have consistently formulated labels does not mean that the labels have to be identical.

	When, for example, links for scrolling through pages in a long document are formulated as follows: “Page 1”, “Page 2”, “Page 3”, they have the same functionality and are formulated consistently but are not identical in content. The requirement covers a range of different component types, such as links, text alternatives to icons and images, toolbars, buttons in forms and the search function.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

46 3.2.5 Change on Request (Level AAA)

46.1 About Success Criteria 3.2.5 and Interpretation

- **Level:** AAA
- **WAD:** No
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#change-on-request

Success Criteria	Changes of context are initiated only by user request or a mechanism is available to turn off such changes.
Purpose	The intent of this Success Criteria is to encourage design of Web content that gives users full control of changes of context. This Success Criteria aims to eliminate potential confusion that may be caused by unexpected changes of context such as automatic launching of new windows, automatic submission of forms after selecting an item from a list, etcetera. Such unexpected changes of context may cause difficulties for people with motor impairments, people with low vision, people who are blind, and people with certain cognitive limitations.
User accessibility needs (Functional Performance Statements)	Does not apply. SC 3.2.5 is not a part of WAD.
Digdir interpretation and specification of the success criteria	None. Not a part of current legislation.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, test rules check that • **bc659a:** The first valid <meta http-equiv="refresh"> element with a content attribute in a document has a time of the content attribute that is 0 (i.e. the page does not reload) or greater than 72000 (i.e. the page reloads after 20 hours or more).
Gap analysis suggested by Digdir – For discussion	Irrelevant.

46.2 Test Rule: meta element has no refresh delay

- **WAI-Tools rule number:** 10
- **ACT Rules id:** bc659a
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 24.08.2020
- **Last updated in GitHub:** 19.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/bc659a

Description	This rule checks that the meta element is not used for delayed redirecting or refreshing.

Accessibility Requirements	2.2.1 Timing Adjustable (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.2.4 Interruptions (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	3.2.5 Change on Request (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	The rule applies to the first valid <meta http-equiv="refresh"> element with a content attribute in a document.
Expectation	The time of the content attribute is 0 or greater than 72000 (20 hours). **Note:** See Refresh state (http-equiv="refresh") for a precise description on how to determine the time.
Assumptions	• This test assumes no functionality was provided by the website for the user to adjust the timer. • This test assumes that the refresh was not essential, which is listed as a valid exception to SC 2.2.1.
Accessibility Support	Not all major web browsers parse the value of the content attribute in the same way. This can cause redirects to happen in some browsers, but not in others. This can cause some pages to pass this rule, while still having a redirect in a minority of web browsers.

47 3.3.1 Error Identification (Level A)

47.1 About Success Criteria 3.3.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#error-identification

Success Criteria	If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.
Purpose	The intent of this Success Criteria is to ensure that users are aware that an error has occurred and can determine what is wrong. The error message should be as specific as possible. In the case of an unsuccessful form submission, re-displaying the form and indicating the fields in error is insufficient for some users to perceive that an error has occurred. Screen reader users, for example, will not know there was an error until they encounter one of the indicators. They may abandon the form altogether before encountering the error indicator, thinking that the page simply is not functional. Per the definition in WCAG 2.0, an "input error" is information provided by the user that is not accepted. This includes: • information that is required by the web page but omitted by the user, or • information that is provided by the user but that falls outside the required data format or allowed values. For example: • the user fails to enter the proper abbreviation in to state, province, region, etc. field; • the user enters a state abbreviation that is not a valid state; • the user enters a non existent zip or postal code; • the user enters a birth date 2 years in the future; • the user enters alphabetic characters or parentheses into their phone number field that only accepts numbers; • the user enters a bid that is below the previous bid or the minimum bid increment. **NOTE:** If a user enters a value that is too high or too low, and the coding on the page automatically changes that value to fall within the allowed range, the user's error would still need to be described to them as required by the success criteria. Such an error description telling the person of the changed value would meet both this success criteria (Error Identification) and Success Criteria 3.3.3 (Error Suggestion). The identification and description of an error can be combined with programmatic information that user agents or assistive technologies can use to identify an error and provide error information to the user. For example, certain technologies can specify that the user's input must not fall outside a specific range, or that a form field is required. Currently, few technologies support this kind of programmatic information, but the Success Criteria does not require, nor prevent it.

	It is perfectly acceptable to indicate the error in other ways such as image, color etc, in addition to the text description. This also includes information on where in the form the error has occurred.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without perception of colour • Usage with limited cognition
Digdir interpretation and specification of the success criteria	The success criteria relate to input errors that are detected automatically. Input errors include: • information required by the web page but which the user has omitted (mandatory fields left blank) • information that is entered in the wrong data format, or the wrong value Information that is entered must remain in the form even if an input error is identified. The purpose of this success criteria is to help the user fill in the form correctly, and in this case keeping the information in the form is an advantage. Nevertheless, the success criteria does not prevent automatic suggestion of corrections, cf. success criteria 3.3.3. If a user enters a value that is too high, for example, and the value is automatically changed so that it falls within the permitted range, the user must nevertheless receive an error description. An error description that tells the user about the change of value will meet both success criteria 3.3.1 (Error identification) and 3.3.3 (Error suggestion).

Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that
	• **36b590:** Each HTML element that has one of the semantic roles of checkbox, combobox, listbox, menuitemcheckbox, menuitemradio, radio, searchbox, slider, spinbutton, switch or textbox. For each test target with no form field error indicator, at least one of the form field error indicators:
	  ○ allows the identification of the related test target, through text, or through non-text content, or through presentation; and
	  ○ describes the cause of error, or how to resolve it in text that is visible and is included in the accessibility tree or included in the accessible name or accessible description of test target.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • None. This is not a comprehensive list of all aspects that may be covered by this success criterion. Aspects of the ACT rules that go beyond the scope of the success criteria: • None.

47.2 Test Rule: Error message describes invalid form field value

- **WAI-Tools rule number:** 48

- **ACT Rules id:** 36b590
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 19.06.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/36b590

Description	This rule checks that text error messages provided when the user completes a form field with invalid values or using an invalid format, identify the cause of the error or how to fix the error.

Accessibility Requirements	3.3.1 Error Identification (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling, Language
Applicability	The rule applies to each HTML element that has one of the following semantic roles: • checkbox, • combobox, • listbox, • menuitemcheckbox, • menuitemradio, • radio, • searchbox, • slider, • spinbutton, • switch or • textbox. **Note**: The list of applicable semantic roles is derived by taking all the ARIA 1.1 roles that: • inherit from the abstract **input** or **select** role, and • do not have a required context role that inherits from the abstract **input** or **select** role.
Expectation	**Expectation 1** Each test target either has no form field error indicators, or at least one of the form field error indicators allows the identification of the related test target, through text, or through non-text content, or through presentation. Note: This rule does not test form field error indicators shown on a different page than that of the test target.

	Note: A single form field error indicator can be related to multiple test targets. For example, an error message at the top of a form can list all the form fields that are required and are empty. Note: A single test target can be related to multiple form field error indicators. For example, a text field can have a red border around it, an error icon adjacent to it, an error message below it, and another error message at the top of the form. All of these are error indicators for the same form field. **Expectation 2** Each test target either has no form field error indicators, or at least one of the form field error indicators describes: • the cause of the error, or • how to resolve it, in text that is visible. **Expectation 3** Each test target either has no form field error indicators, or at least one of the form field error indicators describes: • the cause of the error, or • how to resolve it, in text that is included in the accessibility tree or included in the accessible name or accessible description of the test target.
Assumptions	There are currently no assumptions.
Accessibility Support	There are no major accessibility support issues known for this rule.

48 3.3.2 Labels or Instructions (Level A)

48.1 About Success Criteria X.Y.Z and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 30.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#labels-or-instructions

Success Criteria	Labels or instructions are provided when content requires user input.
Purpose	The intent of this Success Criteria is to have content authors present instructions or labels that identify the controls in a form so that users know what input data is expected. Instructions or labels may also specify data formats for fields especially if they are out of the customary formats or if there are specific rules for correct input. Content authors may also choose to make such instructions available to users only when the individual control has focus especially when instructions are long and verbose. The intent of this Success Criteria is not to clutter the page with unnecessary information but to provide important cues and instructions that will benefit people with disabilities. Too much information or instruction can be just as harmful as too little. The goal is to make certain that enough information is provided for the user to accomplish the task without undue confusion or navigation. This Success Criteria does not require that labels or instructions be correctly marked up, identified, or associated with their respective controls - this aspect is covered separately by 1.3.1: Info and Relationships. It is possible for content to pass this Success Criteria (providing relevant labels and instructions) while failing Success Criteria 1.3.1 (if the labels or instructions aren't correctly marked up, identified, or associated). Further, this Success Criteria does not take into consideration whether or not alternative methods of providing an accessible name or description for form controls and inputs has been used - this aspect is covered separately by 4.1.2: Name, Role and Value. It is possible for controls and inputs to have an appropriate accessible name or description (e.g. using aria-label="...") and therefore pass Success Criteria 4.1.2, but to still fail this Success Criteria (if the labels or instructions aren't presented to all users, not just those using assistive technologies). While this Success Criteria requires that controls and inputs have labels, whether or not these labels are sufficiently clear or descriptive is covered separately by 2.4.6: Headings and Labels.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited cognition Secondary relationship • Usage without vocal capability

	• Usage with limited manipulation or strength
Digdir interpretation and specification of the success criteria	The requirement for labels or instructions applies to forms with more than one field. It is sufficient for labels and instructions to be displayed when the form field is in focus. The understanding article for the success criteria indicates that it is important to assess what information the user needs to be able to use the form. Too much information may make it difficult to get an overview of the content and understand it. However, this does not apply if the form only contains one field. Success criteria 2.4.6 and 3.3.2 are closely interlinked: • 2.4.6 relates to both headings and labels. It does not require that they be provided, but states that if they are provided, they must be descriptive. • 3.3.2 requires that instructions or labels are provided that identify form elements (including whether the form element is mandatory) so that users know what input data is expected. 3.3.2 does not require that the relationship between the label and the form element can be programmatically determined. This is covered by success criteria 1.3.1 and 4.1.2. Nor does 3.3.2 require error correction and help functionality in the form. This is covered by success criteria 3.3.1 and 3.3.3. Form instructions that are images, are covered by success criteria 1.1.1 and 3.3.2. All form elements have labels or instructions that enable the user to use the form. The following requirements must be met: • The form element has a visual identification in the form of a label, instruction or icon, symbol or image. • The identification is visually positioned in or directly next to the form element. • The identification is always visible when the form element is in focus. • The icon, symbol or image is identified in the code. • Information is provided if the form element is mandatory, and any marking with a symbol is explained before it is used.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

49 3.3.3 Error Suggestion (Level AA)

49.1 About Success Criteria 3.3.3 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 31.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#error-suggestion

Success Criteria	If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.
Purpose	The intent of this Success Criteria is to ensure that users receive appropriate suggestions for correction of an input error if it is possible. The WCAG 2.0 definition of "input error" says that it is "information provided by the user that is not accepted" by the system. Some examples of information that is not accepted include information that is required but omitted by the user and information that is provided by the user but that falls outside the required data format or allowed values. Success Criteria 3.3.1 provides for notification of errors. However, persons with cognitive limitations may find it difficult to understand how to correct the errors. People with visual disabilities may not be able to figure out exactly how to correct the error. In the case of an unsuccessful form submission, users may abandon the form because they may be unsure of how to correct the error even though they are aware that it has occurred. The content author may provide the description of the error, or the user agent may provide the description of the error based on technology-specific, programmatically determined information.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage with limited cognition Secondary relationship • Usage without vocal capability • Usage with limited manipulation or strength
Digdir interpretation and specification of the success criteria	This requirement applies to form solutions in which errors in input data are detected automatically. This is information provided by the user and includes • information required by the web page but which the user has omitted (mandatory fields left blank) • information that is provided but is in the wrong format or has the wrong value The Authority also assumes, given success criteria 3.3.1, that information entered, or a suggested correction, must remain in the form element after the error has been detected automatically. This is because the purpose of this success criteria is to help the user to fill in

	the form correctly, and in this case the information or the suggested correction remaining in the form is an advantage. The requirement does not apply if • there are no exact or known ways to correct the error, e.g. spelling, how to write a name, etc. • the input data is linked to security, e.g. a password • a suggested correction would undermine the purpose, e.g. an answer to a quiz
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

50 3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)

50.1 About Success Criteria 3.3.4 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 31.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#error-prevention-legal-financial-data

Success Criteria	For Web pages that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true: • Reversible: Submissions are reversible. • Checked: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them. • Confirmed: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.
Purpose	The intent of this Success Criteria is to help users with disabilities avoid serious consequences as the result of a mistake when performing an action that cannot be reversed. For example, purchasing non-refundable airline tickets or submitting an order to purchase stock in a brokerage account are financial transactions with serious consequences. If a user has made a mistake on the date of air travel, he or she could end up with a ticket for the wrong day that cannot be exchanged. If the user made a mistake on the number of stock shares to be purchased, he or she could end up purchasing more stock than intended. Both of these types of mistakes involve transactions that take place immediately and cannot be altered afterwards, and can be very costly. Likewise, it may be an unrecoverable error if users unintentionally modify or delete data stored in a database that they later need to access, such as their entire travel profile in a travel services web site. When referring to modification or deletion of 'user controllable' data, the intent is to prevent mass loss of data such as deleting a file or record. It is not the intent to require a confirmation for each save command or the simple creation or editing of documents, records or other data. Users with disabilities may be more likely to make mistakes. People with reading disabilities may transpose numbers and letters, and those with motor disabilities may hit keys by mistake. Providing the ability to reverse actions allows users to correct a mistake that could result in serious consequences. Providing the ability to review and correct information gives the user an opportunity to detect a mistake before taking an action that has serious consequences. User-controllable data is user-viewable data that the user can change and/or delete through an intentional action. Examples of the user controlling such data would be updating the phone number and address for the user's account, or deleting a record of past invoices from a website. It does not refer such things as internet logs and search engine monitoring data that the user can't view or interact with directly.
User accessibility needs (Functional	Primary relationship • Usage without vision

Performance Statements)	• Usage with limited vision • Usage with limited cognition Secondary relationship • Usage with limited manipulation or strength
Digdir interpretation and specification of the success criteria	The requirement is restricted to transactions that involve legally binding obligations or rights for the user. This may, for example, include ticket purchases, bank transactions, information in profiles in various data storage systems, etc. At least one of the three options listed below must be provided in the form process: • Form submission can be reversed. • Data can be checked and corrected. • There is a mechanism for review, confirming and correcting information prior to submission.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

51 4.1.1 Parsing (Level A)

51.1 About Success Criteria 4.1.1 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** 16.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#parsing

Success Criteria	In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features. **Note:** Start and end tags that are missing a critical character in their formation, such as a closing angle bracket or a mismatched attribute value quotation mark are not complete.
Purpose	The intent of this Success Criteria is to ensure that user agents, including assistive technologies, can accurately interpret and parse content. If the content has the wrong code syntax, various user agents and assistive devices may interpret and present the content in different ways, or not be able to interpret the content at all.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision Secondary relationship • Usage with limited vision
Digdir interpretation and specification of the success criteria	Web pages must not contain the following errors: • Elements with incomplete start and end tags. • Elements that are not nested in compliance with the specifications. • Elements that contain duplicate attributes. • Elements that have non-unique IDs. In some cases, the W3C HTML validator stops and returns the result “Fatal error”. If the code contains a fatal error, the success criteria is assessed as not met. The success criteria does not apply if it is specified that the markup language used in the web solution permits the situations listed in the bulleted list above. HTML and XHTML do not have such exceptions.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **3ea0c8:** For all id attributes are specified on HTML and SVG elements that are not empty, the id attribute values are unique. • **e6952f:** HTML and SVG starting tags do not contain duplicated attributes.

Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • The web page must not contain elements with incomplete start and end tags. • The web page must not contain elements that are not nested in compliance with the specifications. Aspects of the ACT rules that go beyond the scope of the success criteria: • Applicable markup languages other than HTML.

51.2 Test Rule: Id attribute value is unique

- **WAI-Tools rule number:** 18
- **ACT Rules id:** 3ea0c8
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 28.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi (correct?)
- **URL:** https://act-rules.github.io/rules/3ea0c8

Description	This rule checks that all id attribute values on a single page are unique.

Accessibility Requirements	4.1.1 Parsing (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	Any id attribute whose value is not an empty string (""), specified on a HTML or SVG element. **Note:** Elements that are neither included in the accessibility tree nor visible are still considered for this rule.
Expectation	The value of the attribute is unique across all other id attributes specified on HTML or SVG elements that exist within the same document tree or shadow tree as the element on which the applicable id attribute is specified.
Assumptions	There are currently no assumptions.
Accessibility Support	There are no major accessibility support issues known for this rule.

51.3 Test Rule: Attribute is not duplicated

- **WAI-Tools rule number:** 27
- **ACT Rules id:** e6952f
- **Status in WAI-Tools:** Completed

- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Manual
- **URL:** https://act-rules.github.io/rules/e6952f

Description	This rule checks that HTML and SVG starting tags do not contain duplicated attributes.

Accessibility Requirements	4.1.1 Parsing (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Source Code
Applicability	The rule applies to any starting tag in an HTML or SVG document. **Note:** This rule cannot be tested on the DOM Tree because the browser removes duplicates of any attribute that is already present on an element.
Expectation	For each test target, there are no duplicated attributes.
Assumptions	There are currently no assumptions.
Accessibility Support	There are no major accessibility support issues known for this rule.

52 4.1.2 Name, Role, Value (Level: A)

52.1 About Success Criteria 4.1.2 and Interpretation

- **Level:** A
- **WAD:** Yes
- **WCAG version:** 2.0 and 2.1
- **Last updated in this report:** XX.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#name-role-value

Success Criteria	For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies. (Level A) **Note:** This success criteria is primarily for Web authors who develop or script their own user interface components. For example, standard HTML controls already meet this success criteria when used according to specification.
Purpose	The intent of this Success Criteria is to ensure that Assistive Technologies (AT) can gather information about, activate(or set) and keep up to date on the status of user interface controls in the content. When standard controls from accessible technologies are used, this process is straightforward. If the user interface elements are used according to specification the conditions of this provision will be met. (See examples of Success Criteria 4.1.2 below) If custom controls are created, however, or interface elements are programmed (in code or script) to have a different role and/or function than usual, then additional measures need to be taken to ensure that the controls provide important information to assistive technologies and allow themselves to be controlled by assistive technologies.
User accessibility needs (Functional Performance Statements)	Primary relationship (Source EN 301 549 standard) • Usage without vision • Usage with limited vision Secondary relationship • Usage with limited manipulation or strength
Digdir interpretation and specification of the success criteria	The requirement is for these kinds of user interface components to be identified with names and roles in the code. Names are text that software can use to identify the component itself. Roles are text or numbers that software uses to identify the function of a component. The success criteria applies to all user interface components, such as form elements, links, buttons and iframe elements. For form elements: • Form elements are linked to a label in the code. There are several ways of doing this. The label identifies the form element.

	• If the form element belongs to a group, it is also linked to a label that applies to the group. The label identifies the group For buttons: • Buttons are linked to a label in the code. There are several ways of doing this. The label identifies the function of the button • Buttons that are not coded with standard components in (X)HTML must have information about their roles For iframe elements • If an iframe element is used in the code, the iframe element has a label that identifies the content
Coverage of Success Criteria by ACT rules developed in WAI-Tools	To test the success criteria, the test rules check that • **97a4e1:** Button elements (that are included in the accessibility tree with the semantic role of button, except for input elements of type="image") have an accessible name that is not empty (""). • **e086e5:** Elements, that are included in the accessibility tree, and with the following semantic roles: checkbox, combobox (select elements), listbox, menuitemcheckbox, menuitemradio, radio, searchbox, slider, spinbutton, switch, textbox, have an accessible name that is not empty (""). • **c487ae:** HTML elements with the semantic role of link, that is included in the accessibility tree, have an accessible name that is not empty (""). • **674b10:** Role attribute has a value that is neither empty ("") nor only ASCII whitespace, and that is specified on an HTML or SVG element that is included in the accessibility tree, have a valid value that corresponds to a non-abstract role from WAI-ARIA Specifications. • **4e8ab6:** For HTML or SVG elements that have an explicit semantic role, the WAI-ARIA required states and properties for the role are set and not empty (“”), unless the state or property has a default value listed under WAI-ARIA implicit value for role. • **6cfa84:** Elements with an aria-hidden="true" attribute are not part of sequential focus navigation, nor do they have descendants in the flat tree that are part of sequential focus navigation. • **cae760:** Iframe elements that are included in the accessibility tree and that can be accessed by sequential focus navigation, have an accessible name that is not empty (""). • **59796f:** HTML input elements where state of the type attribute is image, that is included in the accessibility tree, has an accessible name that is not empty (""). • **4b1c6c:** Any set of any two or more iframe elements which are in the same web page (HTML); and are included in an accessibility tree; and that have matching accessible names that are not empty (""). These iframe elements in each set of target elements embed the same resource or equivalent resources.
Gap analysis suggested by Digdir – For discussion	Aspects of the success criteria that are not covered by ACT rules in WAI-Tools: • Whether form elements, buttons and iframes have a name that identifies said element. Most of the test rules only check whether the name is not empty

	Note: This success criteria covers many possible situations. This is not a complete list of possible ways to comply with the requirement. Aspects of the ACT rules that go beyond the scope of the success criteria: • Undetermined.

52.2 Test Rule: Button has non-empty accessible name

- **WAI-Tools rule number:** 3
- **ACT Rules id:** 97a4e1
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/97a4e1

Description	This rule checks that each button element has a non-empty accessible name.

Accessibility Requirements	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility, DOM Tree, CSS Styling
Applicability	The rule applies to elements that are included in the accessibility tree with the semantic role of button, except for input elements of type="image".
Expectation	Each target element has an accessible name that is not empty (""). **Note:** input elements of type submit and reset can get their accessible name from a default text, as well as from a value or other attribute.
Assumptions	The rule assumes that all buttons are user interface components as defined by WCAG 2.
Accessibility Support	Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of button and fail this rule with some technology but users of other technologies would not experience any accessibility issue.

52.3 Test Rule: Form field has non-empty accessible name

- **WAI-Tools rule number:** 4
- **ACT Rules id:** e086e5
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 26.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto

- **URL:** https://act-rules.github.io/rules/e086e5

Description	This rule checks that each form field element has a non-empty accessible name.

Accessibility Requirements	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.1 and above on level A and above
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility, CSS Styling, DOM Tree
Applicability	This rule applies to any element that is included in the accessibility tree, and that has one of the following semantic roles: checkbox, combobox (select elements), listbox, menuitemcheckbox, menuitemradio, radio, searchbox, slider, spinbutton, switch, textbox.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	There are currently no assumptions
Accessibility Support	Certain assistive technologies can be set up to ignore the title attribute, which means that to some users the title attribute will not act as an accessible name. Several assistive technologies have a functionality to list all form fields on a page, including the disabled ones. Therefore, this rule is still applicable to disabled form fields. If an assistive technology consistently ignores disabled form fields in all its interactions, then it is possible to have a disabled form field with no accessible name without creating accessibility issues for the user. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have one of the applicable semantic roles and fail this rule with some technology, but users of other technologies would not experience any accessibility issue. Elements with the option role are not tested in this rule because they do not meet the definition of a User interface component. If these elements are presented as user interface components, these need to be tested separately from this rule.

52.4 Test Rule: Link has non-empty accessible name

- **WAI-Tools rule number:** 9
- **ACT Rules id:** c487ae
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 10.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/c487ae

Description	This rule checks that each link has a non-empty accessible name.
Accessibility Requirements	2.4.4 Link Purpose (In Context) (Level: A) • Required for conformance to WCAG 2.0 and above on level A and higher.

	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	2.4.9 Link Purpose (Link Only) (Level: AAA)
	• Required for conformance to WCAG 2.0 and above on level AAA and above.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	Accessibility Tree, DOM Tree, CSS Styling
Applicability	The rule applies to any HTML element with the semantic role of link that is included in the accessibility tree.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	The rule assumes that all links are user interface components as defined by WCAG 2. When the link role is used on elements that do not behave as links, failing this rule might not mean that the success criteria are failed.
Accessibility Support	There are assistive technologies that do not support using the title attribute for an accessible name, or in which this feature can be disabled. For area elements that have an href attribute, but are not nested inside a map element, there are differences between browsers and assistive technology on if the area is included in the accessibility tree. Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of link and fail this rule with some technology, but users of other technologies would not experience any accessibility issue.

52.5 Test Rule: role attribute has valid value

- **WAI-Tools rule number:** 11
- **ACT Rules id:** 674b10
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 23.09.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/674b10

Description	This rule checks that each role attribute has a valid value.

Accessibility Requirements	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	Any role attribute for which all the following are true: • the attribute has a value that is neither empty ("") nor only ASCII whitespace; and • the attribute is specified on an HTML or SVG element that is included in the accessibility tree.
Expectation	Each test target has at least one token which is a valid value corresponding to a non-abstract role from WAI-ARIA Specifications.
Assumptions	This rule assumes that the implicit role of elements is not enough to satisfy Success Criterion 4.1.2 Name, Role, Value. In case of invalid role attribute, the semantic role defaults to the implicit role. If this is the correct role for the element, the rule will fail but Success Criterion 4.1.2 Name, Role, Value is still satisfied. For example, the element will fail this rule (because image is not a valid role) but satisfies Success Criterion 4.1.2 Name, Role, Value because the element defaults to its implicit role of img.
Accessibility Support	Older browsers do not support more than one token in the value for a role attribute. If multiple values are used in the role attribute, the attribute is ignored in these browsers.

52.6 Test Rule: Element with role attribute has required states and properties

- **WAI-Tools rule number:** 12
- **ACT Rules id:** 4e8ab6
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 20.05.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/4e8ab6

Description	This rule checks that elements that have an explicit role also specify all required states and properties.

Accessibility Requirements	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree
Applicability	Any HTML or SVG element that has an explicit semantic role, except if the element has an implicit semantic role that is identical to the explicit semantic role.
Expectation	For each test target, the WAI-ARIA required states and properties for the role are set and not empty (""), unless the state or property has a default value listed under WAI-ARIA implicit value for role. **Note:** In WAI-ARIA 1.2, required states and properties will no longer have a default value.
Assumptions	The applicability of this rule is limited to explicit semantic roles based on an assumption that all native HTML and SVG elements have native attributes that are mapped to all of the WAI-ARIA required states and properties for the implicit semantic role of the element. The ARIA role is being used to comply to WCAG.
Accessibility Support	This rule relies on browsers and assistive technologies to support leaving out WAI-ARIA required states and properties when a WAI-ARIA implicit value for role is specified in WAI-ARIA Specifications. **Note:** The required states and properties with implicit values can be found in the Core Accessibility API Mappings 1.1 Overview of default values for missing required attributes.

52.7 Test Rule: Element with aria-hidden has no focusable content

- **WAI-Tools rule number:** 15
- **ACT Rules id:** 6cfa84
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 15.10.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/6cfa84

Description	This rule checks that elements with an aria-hidden attribute do not contain focusable elements.

Accessibility Requirements	1.3.1 Info and Relationships (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.
	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling

Applicability	The rule applies to any element with an aria-hidden="true" attribute. **Note:** Using aria-hidden="false" on a descendant of an element with aria-hidden="true" does not expose that element. aria-hidden="true" hides itself and all its content from assistive technologies.
Expectation	None of the target elements are part of sequential focus navigation, nor do they have descendants in the flat tree that are part of sequential focus navigation.
Assumptions	There are currently no assumptions
Accessibility Support	Some user agents treat the value of aria-hidden attribute as case-sensitive.

52.8 Test Rule: iframe element has non-empty accessible name

- **WAI-Tools rule number:** 16
- **ACT Rules id:** cae760
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 17.01.2020
- **Last updated in GitHub:** 24.06.2019
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/cae760

Description	This rule checks that each iframe element has a non-empty accessible name.

Accessibility Requirements	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to iframe elements that are included in the accessibility tree and that can be accessed by sequential focus navigation. **Note:** frame element is deprecated, this rule does not consider frame or frameset elements.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	If an iframe is not perceived by the user as a single control, it does not qualify as a user interface component under WCAG 2. In such a scenario, failing this rule would not fail success criterion 4.1.2. Unless the iframe is both removed from the accessibility tree and removed from sequential focus navigation, they usually are considered to be user interface components.
Accessibility Support	• Some browsers include iframe elements in the sequential focus navigation. This ensures that the contents of iframe elements can be scrolled and accessed by using the keyboard. When an iframe is removed from the accessibility tree, this rule is still applicable for those browsers, unless the iframe is explicitly removed from sequential focus navigation (by having the tabindex attribute set to a negative value).

	• Browser and assistive technology support for iframe elements is currently inconsistent. Some examples of inconsistencies include (but are not limited to):
	  ○ Assistive technologies being set up to ignore the title attribute, which means that to some users the title attribute will not act as an accessible name,
	  ○ There is a known combination of a popular browser and assistive technology that ignores aria-label and only announces title attribute as an accessible name
	  ○ Some assistive technologies ignore empty iframe elements, regardless of if they are focusable or if they have an accessible name.

52.9 Test Rule: Image button has non-empty accessible name

- **WAI-Tools rule number:** 31
- **ACT Rules id:** 59796f
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 20.08.2020
- **Last updated in GitHub:** 29.06.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/59796f

Description	This rule checks that each image button element has a non-empty accessible name.

Accessibility Requirements	1.1.1 Non-text Content (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing
	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and above on level A and higher
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling
Applicability	The rule applies to any input element where the state of the type attribute is image, and that is included in the accessibility tree.
Expectation	Each target element has an accessible name that is not empty ("").
Assumptions	This rule assumes that all image buttons are user interface components as defined by WCAG 2.
Accessibility Support	There is a known combination of a popular browser and assistive technology that does not by default support title as an accessible name.

52.10 Test Rule: iframe elements with identical accessible names have equivalent purpose

- **WAI-Tools rule number:** 39
- **ACT Rules id:** 4b1c6c
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Semi
- **URL:** https://act-rules.github.io/rules/4b1c6c

Description	This rule checks that iframe elements with identical accessible names embed the same resource or equivalent resources.

Accessibility Requirements	4.1.2 Name, Role, Value (Level: A)
	• Required for conformance to WCAG 2.0 and later on level A and higher.
	• Outcome mapping:
	  ○ Any failed outcomes: success criterion is not satisfied.
	  ○ All passed outcomes: success criterion needs further testing.
	  ○ An inapplicable outcome: success criterion needs further testing.

Input Aspect	DOM Tree, CSS Styling, Language
Applicability	This rule applies to any set of any two or more iframe elements which: • are in the same web page (HTML); and • are included in an accessibility tree; and • that have matching accessible names that are not empty (""). **Note:** The test target for this rule is the full set of iframe elements that share the same matching accessible name.
Expectation	The iframe elements in each set of target elements embed the same resource or equivalent resources. **Note:** Resolving the embedded resource includes any redirects that are instant.
Assumptions	• This rule assumes that, within the context of the test subject, the description provided by the accessible name of an iframe can only accurately describe one resource (notably, homonyms alone are not used as iframe names). Thus, if two or more iframe elements have the same accessible name but embed different resources, at least one of them does not describe its purpose. • This rule assumes that the language of each test target can be correctly determined (either programmatically or by analyzing the content), and sufficiently understood.
Accessibility Support	This rule assumes that assistive technologies are exposing all iframe elements on the page in the same way no matter which document tree they are in. If an assistive technology requires the user to "enter" an iframe or a shadow tree before exposing its content (notably nested iframe), then it is possible for two iframe to have identical name but embed different resources without failing Success Criterion 4.1.2: Name, Role, Value (if said iframe are in separate documents or shadow trees)

53 4.1.3 Status Messages (Level AA)

53.1 About Success Criteria 4.1.3 and Interpretation

- **Level:** AA
- **WAD:** Yes
- **WCAG version:** 2.1
- **Last updated in this report:** 31.01.2020
- **URL:** https://www.w3.org/TR/WCAG21/#status-messages

Success Criteria	In content implemented using markup languages, status messages can be programmatically determined through role or properties such that they can be presented to the user by assistive technologies without receiving focus.
Purpose	The intent of this Success Criteria is to make users aware of important changes in content that are not given focus, and to do so in a way that doesn't unnecessarily interrupt their work. The intended beneficiaries are blind and low vision users of assistive technologies with screen reader capabilities. An additional benefit is that assistive technologies for users with cognitive disabilities may achieve an alternative means of indicating (or even delaying or supressing) status messages, as preferred by the user. The scope of this Success Criteria is specific to changes in content that involve status messages. A status message is a defined term in WCAG. There are two main criteria that determine whether something meets the definition of a status message: 1. the message provides information to the user on the success or results of an action, on the waiting state of an application, on the progress of a process, or on the existence of errors; 2. the message is not delivered via a change in context. Information can be added to pages which does not meet the definition of a status message. For example, the list of results obtained from a search are not considered a status update and thus are not covered by this Success Criteria. However, brief text messages displayed *about* the completion or status of the search, such as "Searching...", "18 results returned" or "No results returned" would be status updates if they do not take focus. This Success Criteria specifically addresses scenarios where new content is added to the page without changing the user's context. Changes of context, by their nature, interrupt the user by taking focus. They are already surfaced by assistive technologies, and so have already met the goal to alert the user to new content. As such, messages that involve changes of context do not need to be considered and are not within the scope of this Success Criteria.
User accessibility needs (Functional Performance Statements)	Primary relationship • Usage without vision • Usage with limited vision • Usage without perception of colour • Usage without hearing • Usage with limited hearing • Usage with limited manipulation or strength

	• Usage with limited reach • Usage with minimize photosensitive seizure triggers • Usage with limited cognition Secondary relationship • Usage without vocal capability
Digdir interpretation and specification of the success criteria	None. New requirement in WCAG 2.1.
Coverage of Success Criteria by ACT rules developed in WAI-Tools	None.
Gap analysis suggested by Digdir – For discussion	None.

54 Test rules not related to success criteria in WCAG

54.1 About

In this capital, the test rules that are not required for conformance with WCAG Success Criteria document. These test rules are either related to WCAG Success Criteria Techniques or WAI-ARIA (Accessible Rich Internet Application)[16]. WAI-ARIA is essentially helping with dynamic content and advanced user interface controls developed with web related technologies. Currently certain of these functionalities used in websites with advanced user interface controls that are not available to some users with disabilities. WAI-ARIA helps to make these advance functionalities of websites accessible and useable to people with disabilities. It is therefore important to have tests related to these technologies regardless that are not required for conformance with WCAG Success Criteria.

Following table present the list of test rules which are not required for conformance with WCAG:

Table 2. Overview of ACT Rules not required for conformance with WCAG success criteria

#	ACT-R ID	ACT Rule Name	WCAG SC	Type
1	5c01ea	ARIA state or property is permitted	N/A	Atomic
19	5f99a7	aria-* attribute is defined in WAI-ARIA	N/A	Atomic
20	6a7281	ARIA state or property has valid value	N/A	Atomic
57	46ca7f	Element marked as decorative is not exposed	N/A	Atomic

54.2 Test Rule: ARIA state or property is permitted

- **WAI-Tools rule number:** 1
- **ACT Rules id:** 5c01ea
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 25.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/5c01ea

Description	This rule checks that WAI-ARIA states or properties are allowed for the element they are specified on.

Accessibility Requirements	ARIA5: Using WAI-ARIA state and property attributes to expose the state of a user interface component
	• Not required to conformance to any W3C accessibility recommendation.
	• Outcome mapping:
	  ○ Any failed outcomes: technique is not satisfied.
	  ○ All passed outcomes: technique needs further testing.
	  ○ An inapplicable outcome: technique needs further testing.
	ARIA 1.1, 7.6 State and Property Attribute Processing
	• Required for conformance
	• Outcome mapping:

[16] WAI-ARIA Overview, https://www.w3.org/WAI/standards-guidelines/aria/

	○ Any failed outcomes: success criterion is not satisfied. ○ All passed outcomes: satisfied ○ An inapplicable outcome: satisfied
Input Aspect	Accessibility, CSS Styling, DOM Tree
Applicability	Any WAI-ARIA state or property that is specified on an HTML or SVG element that is included in the accessibility tree.
Expectation	Each test target is either an inherited, supported, or required state or property of the semantic role of the element on which the attribute is specified. If the element has no semantic role, the attribute must be a global state or property. **Note:** Assessing the value of the attribute is out of scope for this rule.
Assumptions	There are currently no assumptions.
Accessibility Support	Implementation of Presentational Roles Conflict Resolution varies from one browser or assistive technology to another. Depending on this, some elements can have a semantic role of none and their attributes fail this rule with some technologies but users of other technology would not experience any accessibility issue.

54.3 Test Rule: aria-* attribute is defined in WAI-ARIA

- **WAI-Tools rule number:** 19
- **ACT Rules id:** 5f99a7
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 07.07.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/5f99a7

Description	This rule checks that each aria- attribute specified is defined in ARIA 1.1.
Accessibility Requirements	This rule is not required for conformance
Input Aspect	DOM Tree
Applicability	Any attribute that starts with aria-.
Expectation	Each target attribute is defined in WAI-ARIA Specifications.
Assumptions	There are currently no assumptions
Accessibility Support	There are no major accessibility support issues known for this rule.

54.4 Test Rule: ARIA state or property has valid value

- **WAI-Tools rule number:** 20
- **ACT Rules id:** 6a7281
- **Status in WAI-Tools:** Completed

- **Last updated in this report:** 28.08.2020
- **Last updated in GitHub:** 24.08.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/5c01ea

Description	This rule checks that each ARIA state or property has a valid value.
Accessibility Requirements	This rule is not required for conformance
Input Aspect	DOM Tree, CSS Styling
Applicability	Any WAI-ARIA 1.1 state or property that is not empty (""), and that is specified on an HTML or SVG element.
Expectation	Each test target has a valid value according to its WAI-ARIA 1.1 value type. For value types ID Reference and ID Reference List for WAI-ARIA required properties at least one of the elements with the given ids exists in the same document tree or shadow tree as the element that specifies the target attribute. For value type URI the value matches the generic URI syntax. **Note:** Only for WAI-ARIA required properties with value types ID Reference and ID Reference List is there a requirement that the elements with the given ids actually exists. For non-required properties, this is not a requirement. **Note:** For value type URI, this rule does not require that the destination URI exists.
Assumptions	This rule catches values that are undefined in WAI-ARIA Specifications, and where the resulting behavior in user agents are also undefined in WAI-ARIA. This might lead to accessibility issues, if the intention was to use behavior defined in WAI-ARIA Specifications. When values are used that do not have a defined behavior in WAI-ARIA Specifications, the HTML/SVG specification decides what default values should be used, since it is defined here what should happen when an invalid value is used for an attribute. If the default value for invalid attribute values happens to match the author's intention for the value, there will not be an accessibility issue.
Accessibility Support	Some user agents treat the value of aria-* attribute as case-sensitive (even when these are not ID) while some treat them as case-insensitive.

54.5 Test Rule: Element marked as decorative is not exposed

- **WAI-Tools rule number:** 57
- **ACT Rules id:** 46ca7f
- **Status in WAI-Tools:** Completed
- **Last updated in this report:** 06.10.2020
- **Last updated in GitHub:** 09.10.2020
- **Rule Type:** Atomic
- **Test Mode:** Auto
- **URL:** https://act-rules.github.io/rules/46ca7f

Description	This rule checks that elements marked as decorative either are not included in the accessibility tree, or have a presentational role.

Accessibility Requirements	This rule is not required for conformance.
Input Aspect	Accessibility Tree, CSS Styling, DOM Tree
Applicability	The rule applies to any element which is marked as decorative.
Expectation	Each target element either is not included in the accessibility tree or has a semantic role of none or presentation.
Assumptions	There are currently no assumptions
Accessibility Support	Implementation of the Presentational Roles Conflict Resolution differs slightly from one user agent to the other. Hence, some elements might be exposed by one user agent and not by another, and consequently might create accessibility issues only for some users. Nevertheless, triggering the conflict is a bad practice.