# CSUN 2015 Submission

**Enabling math on the Web, in Word & PDF, emerging solutions & overcoming support problems.**

Submission Type: General

### Track

* Topic: Education (online)

### Submission Title:

Enabling math on the Web, in Word & PDF, emerging solutions & overcoming issues.

* Session Type: Interactive Lecture
* Session Length: 40 minutes
* Equipment: projector, sound input from two computers, internet connection
* Keywords: Web, Higher Education, K-12

### Author Information

Sina Bahram  
312 Montibello Drive

Cary NC, 27513  
[sina@sinabahram.com](mailto:sina@sinabahram.com)  
919-345-3832

David MacDonald

25 Bayswater Ave.

Unit 22, Ottawa, Ontario, Canada, K1Y 2E5  
613-235-4902

[david@can-adapt.com](mailto:david@can-adapt.com)

CB Averitt

3977 Creek Rd

Timmonsville, SC 29161

Phone: 843-206-5561

Email: [cb.averitt@deque.com](mailto:cb.averitt@deque.com)

### Short Abstract (23 words)

How can we make math accessible on the web, MS Word and PDF? History, current status, solutions and fall backs.

### Extended Abstract

#### Submission Title (75 characters)

Enabling math on the Web, Word & PDF, emerging solutions & overcoming issues.

#### Presenters

CB Averitt  
843-206-5561  
[cb.averitt@deque.com](mailto:cb.averitt@deque.com)

David MacDonald   
613-235-4902  
[david@can-adapt.com](mailto:david@can-adapt.com)

Name: Sina Bahram  
[sina@sinabahram.com](mailto:sina@sinabahram.com)  
919-345-3832

## Body

### Introduction

Wow, has it really been 16 years since MathML 1 was released as a W3C recommendation? After 16 years, you would think that the browser support would be superb, that assistive technology support would be amazing. Unfortunately, that is not the case.

However, this last year has brought a flurry of Math activity that is changing history.

Please join us as we present a short history of math in technology, provide a demonstration of current screen reader support, discuss emerging approaches, and share our experiences with current real world solutions.

Our experience as accessibility consultants has led us to working with math accessibility in the education realm. This is the real world! How can students with disabilities further their education without resources to comprehend simple math equations? We are not vendors of products; we are accessibility consultants, who have been working with educators. Our overview will be non-partisan, and an objective evaluation of our findings.

### Short history

* Up to 1993, math done with special characters, numbers and letters
* 1994: ASTER, first implementation of interactive math for the blind
* 1998: MathML Version 1 W3C Recommendation. XML structure and content 1
* 2003: MathML V2
* 2005: Design Science releases MathPlayer which speaks MathML
* 2009: Release of MathJax, a javascript library for rendering MathML, 2010: W3C MathML V3
* 2013: iOS 7 releases MathML support
* 2013: Chromvox 1.27 supports MathML
* 2014:, DesignScience, ETS, and Sina Bahram, demonstrate what will later be called MathPlayer4 at CSUN on math, using Window Eyes
* 2014: JAWS 16 has introduced MathML support in IE, relying on MathJax JavaScript library,
* 2014: IE11 introduces Enterprise mode acts like IE 8, allows Mathplayer to work.
* 2014: Michael Curran (Mick) writes code so that NDVA can consume MathML and hand it off the Design Science Math API
* 2015: Design Science releases MathPlayer4 public beta which includes rich eyes-free interactive exploration of Mathematics and an API that allows any developer to tap into it. NVDA and WindowEyes do so within Firefox, Microsoft Word, and Microsoft PowerPoint.

### Historical Browser and Assistive Technology Support

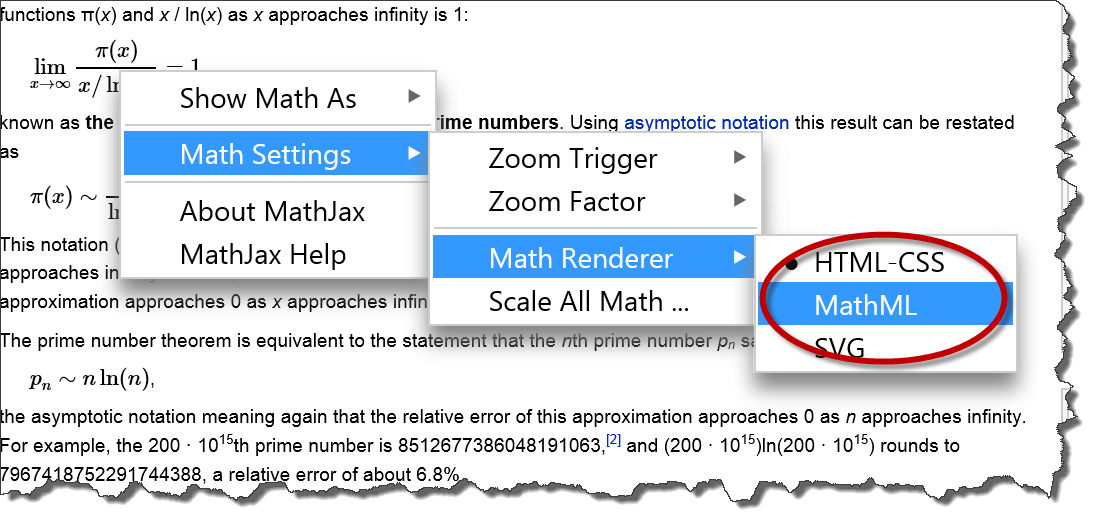
Browser support for MathML did not take off immediately. In recent years it has improved greatly from a visual perspective. As of this writing, MathML is not working with screen readers in Google Chrome (40) or Android (4.4.2), although they do render visually.

Until 2015, Internet Explorer V.9 with the Design Science MathPlayer plugin was the primary way to render math in a browser accessibly because Microsoft dropped binary plugin support in IE versions after IE9. IE11 allows users to switch to Enterprise mode to emulate IE8, enabling MathPlayer to work. This approach is suboptimal.

### Current solutions, Fallbacks: What works now?

#### NVDA (Nightly builds) with Firefox, as long as MathPlayer 4 has been installed on the machine.

When the new MathPlayer4 is installed on a Windows machine, it provides an API that allows the latest builds of NVDA to parse and explore the MathML without MathJax. If MathJax is on the page the user would render it as MathML. To enable this MathJax behavior, the user needs to place focus on a math problem, press the application key (right click), and in the context menu, select **Math Settings>Math Renderer>MathML**



#### MathPlayer4 (self-speaking or with a popular screen reader)

MathML can be read with MathPlayer, which is a [free binary program](https://www.dessci.com/en/products/mathplayer/) that can be installed on Windows. It is much faster than Mathjax which is JavaScript. It currently requires the user to have IE11 [set to Enterprise](http://www.howtogeek.com/184634/how-to-enable-and-use-internet-explorer-11s-enterprise-mode/) mode which mimics versions of IE8 which could run binary extensions. IE dropped binary extension support after IE 9. MathPlayer will either speak the math, or will let a user hear it with a screen reader. If MathJax is on the page, and the MathJax is set to MathML output as described in the section above, it will hand rendering off to MathPlayer because binary code is faster than JavaScript.

#### MathML with MathJax and JAWS16 in IE

MathJax uses JavaScript to render the MathML. It make math visible in all browsers by providing polyfills for older browsers and browsers that don’t support MathML. The author would [add a simple line of JavaScript](http://docs.mathjax.org/en/latest/start.html) code on the web page, in addition to the MathML. The latest version of JAWS16 is the first screen reader to work with the MathML/MathJax combination. JAWS can only read this combination in IE. Jaws can only read and interact with MathML if the author has chosen to use MathJax for rendering.

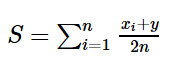
#### Static Images

Companies such as [WIRIS](http://www.wiris.com/) have created an interim solution to address the poor support for MathML. They have a Math WYSIWYG Editor that creates MathML and exports them to .png images and gives those images automatically written alt text that is meaningful. However, there are drawbacks also, such as image pixilation for low vision users who magnify the images. Also, MathML pioneer Neil Soiffer says:

What is best depends upon the input, the context, and the listener. For example, 2/3 might be spoken "two thirds", but if it was 2/3 + 1/30, it might be better to use "2 over 3 plus 1 over 30" for consistancy's sake. Something like 1/(x+1), should be spoken as "fraction with numerator 1, and denominator, x + 1 end fraction" to someone who is blind, but it may not say the fraction/end fraction parts to someone with dyslexia. While alt text is better than nothing (or meaningless file names), it is a distant second to doing the right thing and generating the speech based on the needs of the user.

WIRIS creates and includes MathML (via a data-mathml attribute on the img tag).

Example output from a WIRIS image.



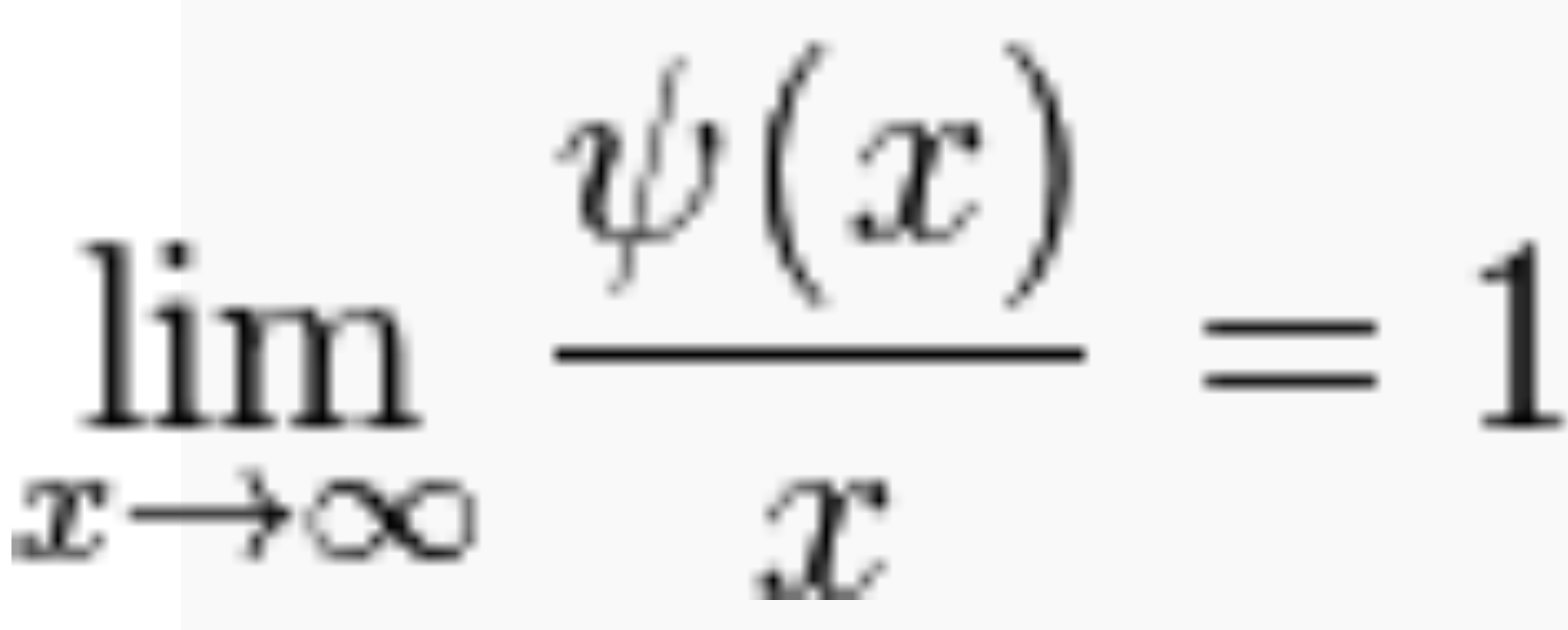
Here’s the code, when it’s created in WIRIS for CK Editor. Notice the data-MathML attribute which provides the source MathML, which is not used in the rendering, but could be extracted by a bookmarklet, and inserted into the DOM, with a few lines of JavaScript, for ingestion by JAWS. Such a bookmarklet would also need to inject MathJax if it were not already present.

<img class="Wirisformula" src="/plugins/demo/ckeditor/php/ckeditor4/plugins/ckeditor\_wiris/integration/showimage.php?formula=13e21b289c84d945761b3c8aa589dace&cw=107&ch=43&cb=28&text=S%20equals%20sum%20from%20i%20equals%201%20to%20n%20of%20fraction%20numerator%20x%20subscript%20i%20plus%20y%20over%20denominator%202%20n%20end%20fraction" style="vertical-align: -15px; height: 43px; width: 107px;"   
data-mathml="<math xmlns="<http://www.w3.org/1998/Math/MathML>"><mi>S</mi><mo>=</mo><munderover><mo>∑</mo><mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow><mi>n</mi></munderover><mfrac><mrow><msub><mi>x</mi><mi>i</mi></msub><mo>+</mo><mi>y</mi></mrow><mrow><mn>2</mn><mi>n</mi></mrow></mfrac></math>" alt="S equals sum from i equals 1 to n of fraction numerator x subscript i plus y over denominator 2 n end fraction">

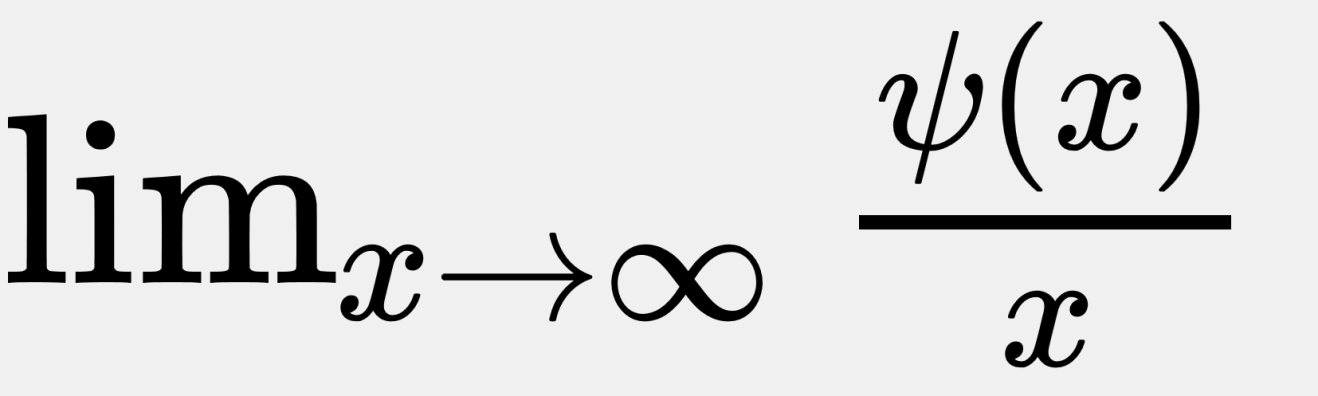
#### Sina Bahram has begun writing a bookmarklet as described above, which extracts the MathML from the WIRIS image and writes it to the DOM while also including MathJax. Such a bookmarklet could not only be used for enabling Jaws to read via MathJax, but it can also be used to enable NVDA to access the MathML via the mathPlayer4 public beta. If you are interested in contributing to this effort, checkout the github repository for MathInjector. <http://www.github.com/sinabahram/MathInjector> .

#### SVG Images vs static images (JPG, PNG, GIF etc.)

SVG will allow a low vision user to zoom in on the image without pixilation. So the image will be sharper. A png image in a wiki math image file at 10x Zoom looks like this:

**

*The same image at the same magnification in SVG follows:*

**

Source for equations: <https://en.wikipedia.org/wiki/Prime_number_theorem>

SVG is a clearer rendering. However it does not contain any semantics, and therefore is far inferior to MathML, even if it has alt text.

#### Inline text

In the absence of using MathML, we recommend that simple one line math problems with basic operators that have no subscript or superscript be written as plain text. Group terms together such that spacing is used only to indicate grouping. Use common sense on groupings. Examples:

* (2x+4)
* 3/4 + 2/3
* 2x + 2y = z

For anything more complex, we recommend using MathML for representing mathematics on the web or MathType within word documents or PowerPoint presentations.

## What should web developers do?

* Use MathML with MathJax
* Author Math in an MathML editor such as [MathType](http://www.dessci.com/en/products/mathtype/) or a competitor ([Math works](http://www.mathworks.com/help/symbolic/mupad_ref/generate-mathml.html), [Math Magic](http://www.mathmagic.com/), [WIRIS](http://www.wiris.com/), [MathHear](http://www.gh-accessibility.com), [etc](https://en.wikipedia.org/wiki/MathML#Editors).) and export to MathML either embedded in a web page or as straight MathML which can be [embedded into a web page](https://en.wikipedia.org/wiki/MathML#Embedding_MathML_in_HTML.2FXHTML_files).
* Use fallbacks, add alt text and/or long descriptions below the MathML. The description should be in an [accessible expand/collapse](http://wet-boew.github.io/wet-boew/demos/details/details-en.html) div to prevent double speak and make the page cleaner for sighted users.
* In some cases SVG or static images (png, jpg etc.) with alternate or long descriptions text (<title> or <description> in SVG) can work if there is not a lot of math, or it is not central to the content. SVG has the advantage over static images. [WIRIS](http://www.wiris.com/) or MathJax do a good job of this and can work stand alone or [with MathType](http://www.dessci.com/en/support/MathType/works_with.asp#!target=wiris_editor).
* Very simple math (addition, subtraction etc.) can be rendered inline as text.

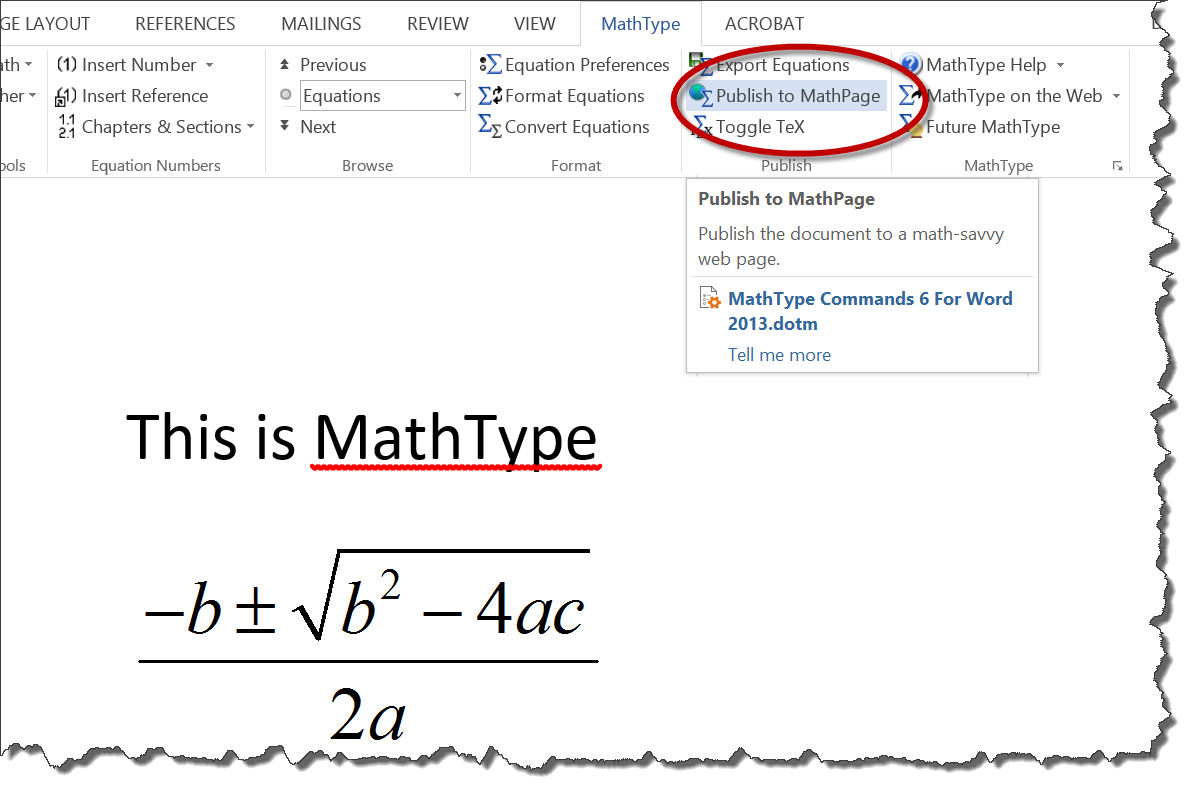
## MS Word

#### What doesn’t work

Math created with the EquationEditor which is included in MS Word doesn’t work with screen readers.

#### What should authors of MS Word documents do?

* Create math problems with MathType ($99 or $57 for academic)
* Render simple math inline as text. See HTML section above for examples.
* The latest versions of NVDA and WindowEyes both can read this math in Word if they have MathType installed
* For Jaws 16 users, export from MathType to an HTML page which will embed proper MathML with MathJax into the web page. Provide that as the alternative to the Word document. A screen shot of how to do that is below.



* If it is incidental math, and not much of it, it can be rendered in an image with alt text. [WIRIS](http://www.wiris.com/) can help with that because it provides the automatic alt text for the equation when it is exported as an image. Sage [MathCloud](https://cloud.sagemath.com/) in the cloud is also a way to provide alt text, leveraging the ChromeVox engine.

### PDF

## In PDF, MathML but we have seen a demo of it working with NVDA and the MathPlayer4. MathML tags were in inserted between opening and closing <figure> tags. The problem right now is there is no authoring environment. Adobe is aware of this and it is likely that an authoring tool such as InDesign, will be able to export to MathML in the near future.

In the meantime, simple equations can be rendered inline (see inline examples above), with an advisory to screen reader users to adjust verbosity and punctuation. However, as soon as any subscript or multilevel equations are used, then images with alternate text or long description (for complex equations) written in the document below the image are necessary.

## Call to Action

* AT manufacturers need to write a little bit of code to sniff for MathML and hand it off to the MathPlayer API.
* Content providers, Generate your mathematics in MathML for the web and MathType for Office documents

## Conclusion

Math on the web and in documents has come a long way. The combination of The W3C’s support for MathML, browser support for HTML5 <math> tags, MathJax, MathML support by assistive technology, and the MathPlayer4 release that works with browsers yield an amazing and equitable experience for people with disabilities. As of this writing a blind user can access Mathematics with JAWS16 + ie11 **OR** with NVDA or WindowEyes with Firefox, Microsoft Word and PowerPoint.

#### References

<http://davidmacd.com/mathml/> (this document)

<https://developer.mozilla.org/en-US/docs/Web/MathML>

<http://en.wikipedia.org/wiki/MathML>

<http://www.wiris.com/>

<https://www.dessci.com/en/>

<http://docs.mathjax.org/en/latest/start.html>

<http://www.freedomscientific.com/About/News/Article/15>

<http://wet-boew.github.io/wet-boew/demos/details/details-en.html>

<http://wet-boew.github.io/wet-boew/demos/mathml/mathml-en.html>

<https://en.wikipedia.org/wiki/Prime_number_theorem>

# Appendix – Biographies of the authors and presenters

## Sina Bahram

Sina is an accessibility consultant, researcher, and entrepreneur. He is the founder of Prime Access Consulting (PAC), an accessibility firm whose clients include high-tech startups, fortune 1000 companies, and museums. Sina is also a doctoral candidate in computer science at North Carolina State University. His field of research is Human Computer Interaction (HCI) focusing on multi-modal approaches for eyes-free exploration of spatial information. Whether wearing his academic or business hat, Sina enjoys devising innovative and user-centered solutions to difficult real-world problems. In 2012, Sina was recognized as one of President Barack Obama's White House Champions of Change for his work in enabling users with disabilities to succeed in Science, Technology, Engineering, and Math (STEM) fields. He presented at CSUN 2014 on the state of MathML with Neil Soiffer. You can read more about Sina and his interests on his website (http://www.SinaBahram.com) and his blog (<http://blog.SinaBahram.com> ). He is @SinaBahram on Twitter. Prime Access Consulting's homepage is at <http://www.pac.bz>

# David MacDonald

David MacDonald is the president of CanAdapt Solutions Inc. [(can-adapt.com](http://www.can-adapt.com)), a Canadian company with a global reach. David presented at CSUN in 2014, on how he helped the Canadian Government meet its court orders to follow WCAG as the result of the Donna Jodhan federal court case.

Since 2002, he has been an Invited Expert on the Web Content Accessibility Guidelines Working Group (WCAG WG) of the W3C. He was on the 15 member WCAG ICT task force which reported on the application of WCAG to desktop software which served as a foundation for the new US Government 508 Refresh, and the European M376. He is a W3C Invited Expert for the HTML5 working group and its Accessibility Task Force, and an editor of the W3C Document “Using WAI-ARIA in HTML5”, a contributor to the Website Accessibility Conformance Evaluation Methodology, and is a member of the WCAG Mobile Accessibility Task Force.

David consulted to the Ontario Office for Disability issues while they created AODA, the first legislation in the world specifically requiring the private sector to follow WCAG. David was a member of the advisory board to the Ontarian with Disabilities Support Program and was the director of the United Nations International Day of Disabled Persons at the Ottawa Congress Centre. He has performed audits for organizations such as Intel, Bayer Pharmaceuticals, PC Financial, Sasktel, Great West Life, University of Ottawa, Vision Australia numerous other Fortune 500 companies. He has taught WCAG to thousands of developers, QA, managers, and content providers. Twitter [@davidmacd](https://twitter.com/davidmacd) LinkedIn <http://ca.linkedin.com/in/davidmacdonald100>

# CB Averitt

CB Averitt is a Senior Consultant with Deque Systems. The company is a digital accessibility company and produces accessibility software in addition to offering accessibility consulting services. CB has been in web development for over 14 years. Having owned and operated Averitt Web Services, he has experience with front-end and back-end development as well as design and user experience.

In 2005, CB was introduced to accessibility from working in the education sector. CB was heavily involved with making SC colleges Section 508 compliant. Since that time CB shares his expertise and passion for the open web with government, education and companies that rage from small business to Fortune 50. CB has extensive knowledge in WCAG, AODA, and Section 508 regulations.

CB has completed hundreds of assessments and remediation in numerous technologies such as web, PDF and mobile. He has presented trainings in CSS as it relates to accessibility as well as trainings for web testers. CB has been a volunteer with The South Carolina Assistive Technology Advisory Committee (SC ATAC) for over 7 years. Twitter @dive4cb.