CiTO Tools

User Guide - The JavaScript Reference Annotation Tool

The purpose of this document is to inform users of the functionality of the CiTO JavaScript Reference Annotation Tool, that inserts a CiTO Properties choice box after every reference in an article's reference list, thereby permitting users to choose the CiTO properties that best explain why the Citing Article cites the Cited Article.

This functionality has been created by Tanya Gray ([trgatgra@gmail.com](mailto:trgatgra@gmail.com)) and David Shotton ([david.shotton@zoo.ox.ac.uk](mailto:david.shotton@zoo.ox.ac.uk)), and was funded by the JISC Open Citations Extension Project.

Further details about CiTO are given in:

Peroni S and Shotton D (2012). FaBiO and CiTO: ontologies for describing bibliographic resources and citations. *Web Semantics: Science, Services and Agents on the World Wide Web*. **17**: 33-34. [doi:10.1016/j.websem.2012.08.001](http://www.sciencedirect.com/science/article/pii/S1570826812000790?v=s5).

**Contents**

[Examples](#h.xosd5k7a3ha7)

[Usage by publishers](#_Usage_by_publishers)

[What happens when a user clicks on a CiTO property?](#h.5mp0jqs0k5z3)

[CiTO property that was previously unselected](#h.1arkqa2zmbxq)

[Value stored in user’s web browser](#h.6cmk0t7ke1bt)

[Value stored in external database](#h.86aj5zz4rh50)

[CiTO property that was previously selected](#h.3z4rlvbalcoi)

[Value stored in user’s web browser](#h.5atiwhbp2ohv)

[Value stored in external database](#h.rkcvubt3fbfx)

[Web Services](#h.nqfvmfhae6ta)

[Browser support](#_Browser_support)

[XMLHttpRequest](#_XMLHttpRequest)

[HTML5 Local Storage](#h.i561pzt2d4gm)

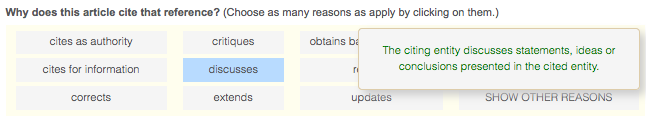
# Examples

Examples of journal articles that have the CiTO JavaScript Reference Annotation Tool functionality embedded in them are provided for five articles from different journals/sources:

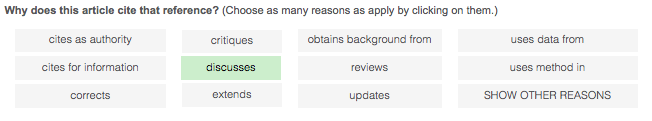
* PLOS Currents
* eLife
* PubMed Central
* ZooKeys
* Our semantically enhanced version of an article by Reis *et al*. in PLoS Neglected Tropical Diseases, [doi:10.1371/journal.pntd.0000228.x001](http://dx.doi.org/10.1371/journal.pntd.0000228.x001).

To examine these examples, and to make CiTO annotations on their references:

1. Download the Zip file from the [CiTO GitHub Repository](https://github.com/tgra/cito/) by clicking the "ZIP" tab on the CiTO GitHub Repository landing page. (Filename cito-master.zip, location <https://github.com/tgra/cito/archive/master.zip>, size 7.7 mBytes.)
2. Unpack the zip file on your local computer by double clicking it.
3. Open the folder "cito-master" that is created, then open the folder "examples". In this, you will see five sub-folders, one for each of the five CiTO-enhanced articles.
4. Double-click the .html file in any one of these folders to open that CiTO-enhanced article in your web browser.
5. Scroll down to the "References" section at the end of the article, where you will see the following CiTO Tool Annotation Box, presenting the eleven most common CiTO citation annotation properties:



1. Hovering with the mouse over one of these properties will cause its button to change to a light blue, and will cause a pop-up box to appear, displaying the definition of this property drawn from the Citation Typing Ontology, as shown:
2. Clicking on one of the CiTO property buttons will cause its appearance to change to green, showing it has been selected. The property definition pop-up will remain for as long as the mouse continues to hover over the property, and the green colour will persist after the mouse has been moved away, as shown in the following figure.



Choosing a property in this manner results in that choice being recorded, as explained below.

1. Re-clicking on a green button that has been selected will "un-choose" that property, reverting the button appearance from green to grey (or light blue while still hovering over it), and will also cause that reversal of choice to be recorded, as explained below.
2. You are free to choose as many CiTO properties for any one reference as you think apply. If none of the eleven displayed choices are appropriate, click the SHOW OTHER REASONS button to display the other 28 CiTO properties, which can be selected in the same manner, as shown here:



Clicking HIDE OTHER REASONS will hide these additional options, but will not negate any selections you have made from among them.

1. Continue making choices for each of the references in the reference list. You may stop your citation annotation activity at any time.
2. If you now save this annotated article as an .html file ("save page as" from the file menu) in the same directory as the original file, but with a different filename, these annotations will be saved with the article and will be visible when the annotated article is re-opened in a browser.

# Usage by publishers

To achieve the CiTO JavaScript Reference Annotation Tool functionality displayed in the examples, the following code needs to be inserted into the XML journal article. For mainstream use, this needs to be done by the publisher before the article is published.

<script type="text/javascript" src="./javascript/cito.js?reftype=REFTYPE"></script>  
 <link rel="stylesheet" type="text/css" href="./javascript/cito.css">  
 <script language="JavaScript" type="text/javascript" src="./javascript/jquery.min.js"></script>

where REFTYPE is replaced with one of the following values, depending on the type of reference list:

* pubmed for articles in PubMed Central and Europe PubMed Central
* elife for articles in the *eLife* journal
* zookeys for articles in the Pensoft *Zookeys* journal
* plos for articles in *PLoS Currents*.

To adapt this CiTO JavaScript Annotation Tool for articles in other journals that use a different DTD, or that use a different method of mapping the NLM-DTD v3.0 to HTML than that used by PubMed Central, it is necessary to add two new functions to the cito.js file, one to identify the HTML for the reference list, and another to extract a DOI, a URL or a textual bibliographic citation for the reference, that would then be used as the object of the bibliographic citation. The code is not complicated, but its modification requires someone with an understanding of JavaScript.

# What happens when a user clicks on a CiTO property?

## CiTO property that was previously unselected

When a user clicks on a CiTO property that was previously unselected, the value is stored in the user’s web browser and also sent to an external web service and saved in our CiTO Tools Annotations Database.

### Value stored in user’s web browser

* A key-value pair is stored in the browser’s web storage facility<http://en.wikipedia.org/wiki/Web_storage> using the following statement: localStorage.setItem(key,value);
  + The **key** is set to a value created by concatenating the browser window’s url and the unique identifier for the HTML that forms the CiTO property ‘button’ in the web page.
  + The **value** is set to ‘1’.

### Value stored in external database

* An AJAX request is sent that inserts a record into our CiTO Tools Annotations Database hosted at www.miidi.org with the following fields:
  + unique id for database record (auto-increment)
  + userid - unique identifier for user - stored in browser’s web storage facility
  + timestamp - when action was taken
  + action = ‘add’
  + subject - url for the journal article == window.location.href
  + predicate - CiTO property
  + object - url (or citation text if url not available) parsed from reference item

record example:

|294|KDYXFJ4IM2RIAUBYRYUWPWO37BLNSD|Fri, 04 Jan 2013 17:25:34 GMT|add|<http://elife.elifesciences.org/content/1/e00248>|<http://purl.org/spar/cito/obtainsBackgroundFrom>|<http://dx.doi.org/10.1073/pnas.0700629104>

## CiTO property that was previously selected

When a user clicks on a CiTO property that was previously selected, the value is stored in the user’s web browser and also sent to an external web service and saved in our CiTO Tools Annotations Database.

### Value stored in user’s web browser

* A key-value pair is stored in the browser’s web storage facility<http://en.wikipedia.org/wiki/Web_storage> using the following statement: localStorage.setItem(key,value);
  + The **key** is set to a unique value created by concatenating the browser window’s url and the unique identifier for the HTML that forms the CiTO property ‘button’ in the web page.
  + The **value** is set to ‘0’.

### Value stored in external database

* An AJAX request is sent that inserts a record into a database hosted at [www.miidi.org](http://www.miidi.org) with the following fields:
  + userid - unique identifier for user - stored in browser’s web storage facility
  + timestamp - when action was taken
  + action = ‘remove’
  + subject - url for the journal article == window.location.href
  + predicate - CiTO property
  + object - url (or citation text if url not available) parsed from reference item

## 

# Web Services

## CiTO Triples

A web service is available to view all the CiTO annotation triples submitted using the JavaScript and Chrome functions by all users on all annotated papers. To view, visit <http://www.miidi.org/metaquery/cito>. This downloads a file called cito containing the accumulated annotation records in the following format:

UniqueOpaqueUserID|DateTime|operation|CitingPaper|CiTOProperty|CitedPaper

For example:

K29GS7D0C1P7SLD17XFC353M1L986U|Fri, 11 Jan 2013 10:01:51 GMT|add|<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808925/>|<http://purl.org/spar/cito/obtainsBackgroundFrom>|<http://www.ncbi.nlm.nih.gov/pubmed/17420237>

The reference in the reference list that is the object of this annotation, as well as the cited paper that is referenced, are both defined by the CitedPaper URI.

The last three items in each record are easily transformed into an RDF triple (in Turtle format):

<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808925/>

<http://purl.org/spar/cito/obtainsBackgroundFrom>

<http://www.ncbi.nlm.nih.gov/pubmed/17420237> .

# 

# Browser support

With the browser-independent javascript file, the issue of browser support raises itself. There are two aspects of the code where this is relevant:

* the use of HTML5 Local Storage
* the use of XMLHttpRequest to make AJAX calls

### XMLHttpRequest

The code instantiates the XMLHttpRequest object with the following line:

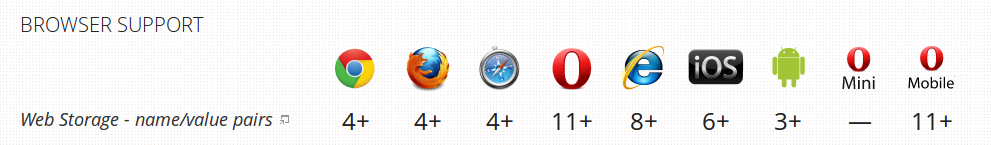
var xhr = new XMLHttpRequest();

This is supported by all modern web browsers including Chrome, Firefox, Safari, Opera and Internet Explorer 7+. It is not supported by IE6 and below.

### HTML5 Local Storage

The use of HTML5 Local Storage to store local name value pairs is supported by the latest versions of the most commonly-used web browsers, as listed at<http://www.html5rocks.com/en/features/storage>

These two features of the CiTO Javascript file mean that the web browser versions listed below are the oldest that are supported:



An alternative demonstrator implementation of this same functionality, by means of that uses the CiTO Chrome Extension, is available free from the Chrome WebStore here:

<https://chrome.google.com/webstore/detail/annotate-pubmedelifeplos/geajighoohelnjnhfmhbcaddbcgcbphn>.

This works only for the [Chrome Browser](https://www.google.com/intl/en/chrome/browser/), and enables annotations only of papers from PubMed Central, eLife and PLoS Currents, **without** them having to be modified by the publisher as described above.

The software for this CiTO Chrome Extension is stored in the chrome-extension folder of the [CiTO GitHub Repository](https://github.com/tgra/cito/).

We welcome the involvement of developers in the community who would be interested working with our code to create and maintain similar extensions / add-ons / plug-ins for other browsers.