

# Technical Disclosure Commons

---

Defensive Publications Series

---

November 10, 2016

## Right-to-Left Conversion Tool

Devin Mancuso

Follow this and additional works at: [http://www.tdcommons.org/dpubs\\_series](http://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Mancuso, Devin, "Right-to-Left Conversion Tool", Technical Disclosure Commons, (November 10, 2016)  
[http://www.tdcommons.org/dpubs\\_series/314](http://www.tdcommons.org/dpubs_series/314)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## **Right-to-Left Conversion Tool**

### BACKGROUND

Some content presented on the Internet is in a language that is normally written right-to-left (RTL), such as Hebrew, Arabic, or Persian. Existing design tools used for designing content can often produce content that is subsequently rendered incorrectly. This can happen, for example, because some content design tools don't support a proper conversion of content to an RTL layout. As a result, content normally presented left-to-right (LTR), when rendered, can appear, to audiences viewing the content, as illegible. Unless a content designer is fluent in a particular RTL language in which content is to be presented, mistakes that may be made during the design of content can be difficult to spot. Additionally, most design tools do not provide the ability to preview content in RTL format. As such, current content designers are forced to manually create RTL versions of each of their designs in order to support rendering of RTL layouts.

### DESCRIPTION OF DRAWINGS

Figure 1 shows an example of LTR content.

Figure 2 shows an example of RTL content created from the LTR content using the LTR-to-RTL conversion option provided by the plugin.

### DETAILED DESCRIPTION

A system is described for automatically creating, e.g., within a content design tool, RTL versions of LTR content. RTL versions of content can be created, for example, using a

plugin (e.g., written in JavaScript or some other language) that is included in design tool software. The plugin can be used, for example, to speed up the process of creating RTL designs and to improve RTL consistency across products. The plugin can be used in content design tools and software, as well as in other applications, such as photo editing and drawing applications.

As an example, an LTR-to-RTL plugin can take a designer's design that is stored within a design tool canvas, such as an artboard, and duplicate the artboard, performing a translation on each layer of the design to create an RTL layout alternative of the original LTR design. In one example technique of performing the translation, the plugin can calculate a distance between a design element of the content and the leftmost edge of the artboard. The distance can then be used, for example, to move the design element's X position to the right until the element is positioned at the same distance from the rightmost edge of the artboard.

When the plugin encounters font layers, for example, within the design, the plugin can perform a similar translation, but can also change the text's alignment to right alignment. In this way, an RTL font can be easily substituted in.

When the plugin encounters a scalable vector graphics (SVG) icon or icon-font formatted with a specific layer name, the plugin can perform a look-up against a list of known icons that should be mirrored when presented in RTL mode. If the plugin finds a match between the icon in the design and one of the known icons, the plugin can mirror the element, e.g., performing a horizontal flip. Other types of content elements can be handled on a case-by-case basis, such as if the plugin decides whether or not to mirror the content element or leave the content element unchanged.

Other types of content elements can be handled in special ways. For example, to ensure that numbers are still read left-to-right like they do in LTR languages, the plugin can

determine if a string contains only numeric characters. The plugin can ignore special characters, such as brackets, parentheses, and/or dashes in phone numbers, while identifying strings as being numeric if the strings contain decimal points or other numeric delimiters, including currency signs for currency amounts. A numeric field can therefore retain its left alignment during repositioning among other content elements, as opposed to changing to right alignment.

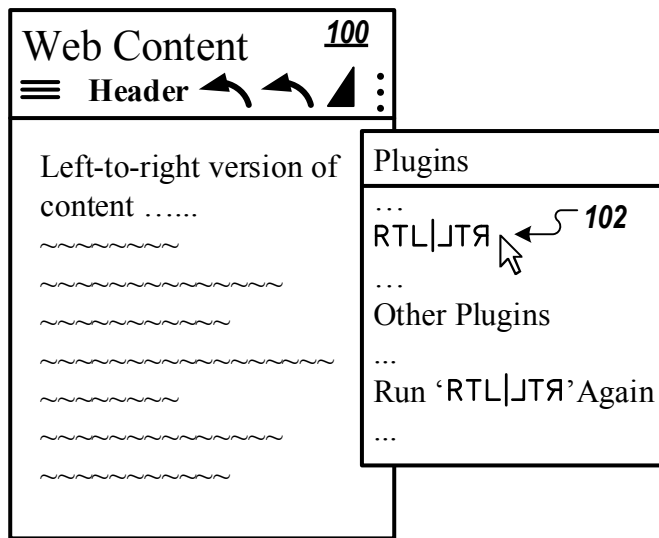


Figure 1

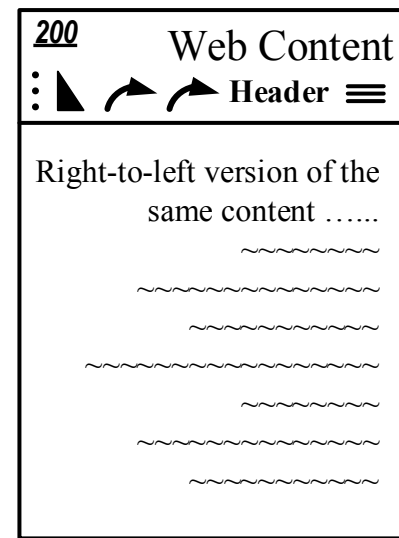


Figure 2

Figure 1 shows an example of LTR content 100. The content 100 can be an article, for example, that is being created by a content designer for presentation on the Internet, such as on a web page. Other types of content are possible. In order to create a RTL version of the same content that is in the content 100, the content designer can use a LTR-to-RTL conversion option 102, e.g., selected from a pull-down menu of plugin options and/or other tools. Other ways of initiating the generation of the RTL version are possible.

Figure 2 shows an example of RTL content 200 created from the LTR content 100 using the LTR-to-RTL conversion option 102 provided by the plugin.

In the example provided above with reference to Figure 1, an entire artboard is shown as being translated from LTR to RTL. However, it is also possible that the designer can choose to translate one or more individual elements within an artboard canvas without applying the translation to the entire artboard itself. This can allow the designer to select individual content elements, while excluding images, charts or other elements for which the LTR orientation is to be preserved in the RTL version.

## ABSTRACT OF THE DISCLOSURE

A system is described for automatically creating, e.g., within a content design tool, a right-to-left (RTL) version of left-to-right (LTR) content. RTL presentation of content can be used, for example, for content presented in a language that is normally written right-to-left, such as Hebrew, Arabic, or Persian. RTL versions of content can be created, for example, using a plugin (e.g., written in JavaScript) included in design tool software. Plugins can speed up the process of creating RTL designs and improve RTL consistency across products. Plugins can also be used in other applications, such as photo editing and drawing applications.