Bidirectional Script Support

Hussein Shafie, XMLmind Software <xmleditor-support@xmlmind.com>

June 30, 2026

Abstract

This document explains how to use XMLmind XML Editor as effectively as possible in order to create documents mixing right-to-left (**RTL**; like Arabic and Hebrew) and left-to-right (**LTR**; like English and French) scripts.

Table of Contents

1. Requirements 1
1.1. Install add-on 1
1.1.1. Alternative spell checker add-on 2
1.2. Explicitly declare the directionality of text 2
1.2.1. The "**Set dir**" button 2
2. Features 3
3. Limitations 6
4. Preferences 7
A. Typing Arabic (اللغة العربية) or Hebrew (עברית) when you don't have the corresponding keyboard 9
B. Enabling bidi support in your custom **XXE** configuration 9
1. The `directionalityFinder` configuration element 9
2. The "**Set dir**" button 10

1. Requirements

1.1. Install add-on

The "**Bidi Support**" add-on adds bidirectional script support to XMLmind XML Editor (**XXE**).

The "**Bidi Support**" add-on is not installed by default. If you plan to author documents containing right-to-left scripts (e.g. Arabic – عربي, Hebrew – עברית), you must *really* install the "**Bidi Support**" add-on, as, out of the box, **XXE** has no bidirectional script support whatsoever. Without the "**Bidi Support**" add-on, even the most basic editing features, like the location of the insertion cursor (caret), won't work or would be incorrect.

On the other hand, do *not* install the "**Bidi Support**" add-on unless you have a real need for it. Installing this add-on has a substantial performance penalty on **XXE**, even when authoring documents not containing any right-to-left scripts.

The "**Bidi Support**" add-on may be installed using menu item "**Options → Install Add-ons**" in *XMLmind XML Editor - Online Help*. A sample XHTML document containing English, Arabic and Hebrew, created using **XXE**, is found in *`bidi_support_addon_install_dir`*`/samples/sample1_en_ar_he.html`. Some DocBook and DITA samples are also found in the same directory.

Many thanks to our sponsors!

The development of the new "**Bidi Support**" add-on, a large and complex add-on indeed, has been entirely funded by Université de Caen Normandie and **CNRS** (IR Métopes and Equipex Biblissima). These French education and research public organizations have agreed to make this add-on an integral part of the XMLmind XML Editor product, hence to make it available to all XMLmind XML Editor users, including *free-to-use* Personal Edition users. Many thanks to our generous sponsors!

1.1.1. Alternative spell checker add-on

You may also want to install the "**Hunspell Spell Checker**" add-on. Unlike XMLmind Spell Checker, the spell checker normally used by **XXE**, Hunspell has dictionaries for languages using **RTL** scripts, for example, Arabic and Hebrew.

The "**Hunspell Spell Checker**" add-on may be installed using menu item "**Options → Install Add-ons**"[1]. Its dictionaries are installed using the **Add** button found in the "**Add-on|Hunspell Spell Checker**" preferences sheet of the **Preferences** dialog in *XMLmind XML Editor - Online Help* box (**Options → Preferences** in *XMLmind XML Editor - Online Help*).

1.2. Explicitly declare the directionality of text

Declaring the directionality of the text contained in an element is generally done using the global, “inherited”, `dir` attribute. The `dir` attribute is supported by XHTML, DITA and DocBook[2].

DITA example:

```
<p dir="rtl" xml:lang="ar">السلام عليكم!</p>
```

The language of an element, generally specified using attribute `lang` or `xml:lang`, is *not* used to determine the directionality of the text contained in an element. However, it is strongly recommended to specify both `dir` and `lang` attributes at the same time.

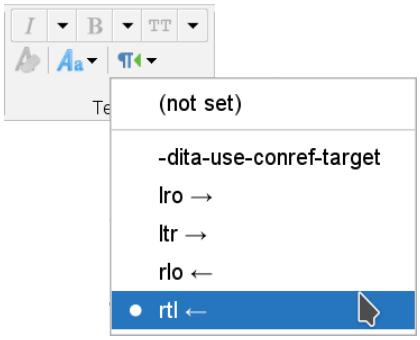
1.2.1. The "Set dir" button

The "**Set dir**" button added to the XHTML, DITA and DocBook toolbars by the "**Bidi Support**" add-on shows you the directionality of the text contained in implicitly or explicitly selected element: left-to-right, right-to-left or unknown.

[1] No need to uninstall the "**XMLmind Spell Checker**" add-on, the spell checker normally used by **XXE**, before doing this. When installed the "**Hunspell Spell Checker**" add-on automatically supersedes the "**XMLmind Spell Checker**" add-on.

[2] But not by TEI Lite, which is supported by **XXE** after installing the "**TEI Lite configuration**" add-on. TEI Lite uses the `xml:lang` attribute to determine the the directionality of the text contained in an element. For example, `xml:lang="he"` (Hebrew) implies that the element contains some RTL text.

*Figure 1. The "**Set dir**" button as found in the DITA toolbar*



Clicking this button displays a menu letting you set, modify or remove the corresponding directionality attribute (typically the `dir` attribute).

Removing the directionality attribute from selected element generally[3] implies inheriting the text directionality from ancestor elements. This is done by selecting "**(not set)**" in the menu.

Using the Attributes tool to specify the `dir` attribute

Alternatively, you may use the **Attributes** tool in *XMLmind XML Editor - Online Help* to specify the `dir` attribute.

The `dir` attribute, which is rarely needed, is generally not listed by the **Attributes** tool. In order to display it, you may have to click the "eye icon" in *XMLmind XML Editor - Online Help* found at the left of the header of the **Attributes** table. This icon displays a menu. Select the "**Show 'Localization' Attributes**" item for a DITA document. Select the "**Show 'Other Common' Attributes**" item for a DocBook document.

2. Features

What's explained below is showcased in a short screencast published on YouTube. See https://youtu.be/8n3LSGAKlEQ.

XXE bidirectional script support is best explained by an example. Let's use this short Arabic sentence for that:

← *From right to left*					
.1978	عام	القاهرة	في	طارق	ولد
1978.	*'am*, year	*Al Kahira*, Cairo	*fi*, in	*Tarek*, a common first name	*walada*, was born

*Figure 2. Arabic equivalent of "**Tarek was born in Cairo in 1978.**" as rendered by **XXE***

[3] XHTML elements `bdi` and `bdo` do *not* inherit `dir` from their ancestor elements.

In the above figure, option "**Give characters a distinctive color**" [8] has been turned on to give **RTL** text runs a distinctive dark magenta color.

Also note that while Arabic and Hebrew are written from the right to the left, numbers, whether using western digits or Arabic digits (e.g. "١٩٧٨"), are still written from the left to the right. That's why "`1978`" is not given a dark magenta color in the above figure.

- Setting attribute `dir="rtl"` has immediate effects on the tree and styled views of an element. For example, the text align automatically changes from left to right. Other example, in the styled view, the left and right margin properties are inverted, the left and right padding properties are inverted, etc.

- The insertion cursor (caret) changes shape inside an **RTL** character sequence and also inside a text node containing both **RTL** and **LTR** character sequences.

  It is given a small arrow which indicates the directionality of the character following the caret. Inside an **RTL** character sequence, the caret looks like this: . Inside an **LTR** character sequence, the caret looks like this: .

  Note that the caret is not given any special shape inside text nodes containing only **LTR** characters.

- Inside an **RTL** character sequence, pressing key Left moves the caret to the left, that is, to the following character in the sequence and pressing key Right moves the caret to the right, that is, to the preceding character in the sequence[4].

  This behavior is deemed the most intuitive one but it has important consequences. For example, let's suppose the caret is inside "طارق" (Tarek). Pressing repeatedly key Left to reach the period which ends the sentence will get you “stuck” when the caret reaches "1978".

  When the caret reaches "1978", an **LTR** character sequence, you'll have to press key Right four times to repeatedly move the caret to the following character in the sequence. After doing that, the caret changes its shape and pressing key Left one more time will take you past the period which ends the sentence.

- Inside an **RTL** character sequence, pressing key Backspace deletes the character found at the left of the caret, that is, deletes the following character in the sequence and pressing key Delete[5] deletes the character found at the right of the caret, that is, deletes the preceding character in the sequence.

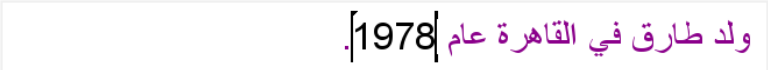
  Similarly, inside an **RTL** character sequence, pressing **Ctrl**+Backspace deletes the word found at the left of the caret. Pressing **Ctrl**+Delete[5] deletes the word found at the right of the caret.

The rationale behind this behavior of Backspace is that this keyboard key generally looks like *an arrow pointing to the left* and containing an `X`: ⌫.

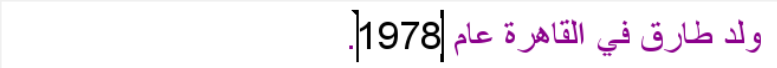
[4]Inside an **RTL** character sequence, pressing Shift+Left (respectively, **Ctrl**+Shift+Left) extends the text selection by one character (respectively, by one word) at the left of the caret. Pressing Shift+Right (respectively, **Ctrl**+Shift+Right) extends the text selection by one character (respectively, by one word) at the right of the caret.
[5]Some Mac computers, like the Macbook, don't have a Delete key. These have just a Backspace key. When this is the case, pressing **Fn**+Backspace is equivalent to pressing the Delete key.

If you find this default behavior non-intuitive then please revert to customary "Backspace deletes preceding character" using the **Preferences** dialog box. More information below [8].

- The above experiment [4] will show you an unexpected screen artifact as soon as the caret reaches "1978" after pressing key Left a number of times.

  After pressing key Right four times to go past "1978", you'll see a slightly different screen artifact.

  These artifacts are *secondary insertion cursors*.

  When relevant, **XXE** displays a secondary insertion cursor (looking like a “flattened” square bracket) in addition to the actual caret (looking like a little flag):

  - When the actual caret is before "1978", typing any **LTR** character (e.g. "0") inserts this character before the "1". The secondary insertion cursor, looking like a `[`, is a hint indicating that typing any **RTL** character (e.g. "و") inserts this character after the space following "عام" (year).
  - When the actual caret is before the ending period, typing any **RTL** character (e.g. "م") inserts this character before the period. The secondary insertion cursor, looking like a `]`, is a hint indicating that typing any **LTR** character (e.g. "A") inserts this character after the "8".

- An Arabic diacritic (تشكيل, *tashkil*) or a Hebrew diacritic (נקוד, *niqqud*) must be typed after the consonant letter to which the diacritic is to be added. For example, in order to add a فتحة (*fathah*; sounds like a short /a/) to letter "ط", you must first move the caret after letter "ط" and then type the *fathah*.

  Other example, in order to add a كسرة (*Kasrah*; sounds like a short /i/) to letter "ر", you must first move the caret after letter "ر": and then type the *Kasrah*: .

- An Arabic diacritic or an Hebrew diacritic is rendered as if it has been combined with the consonant letter bearing it, but in fact, this is not the case. Example: while typing "o" and then typing "^" inserts a single character "ô" into the document being edited, typing "ط" and then typing a *fathah* inserts *two distinct characters*, "ط" and the *fathah*, into the document being edited.

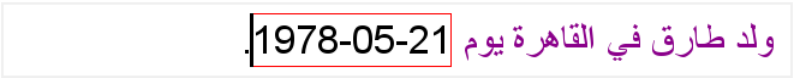
  This could pose a usability problem because this implies that, for example, in order to delete "ط" and its *fathah*, the author would have to press key Delete twice. Fortunately, **XXE** considers that a letter and all its diacritics have been combined and now form a single editable entity. Therefore pressing key Left or key Right skips the letter and all its diacritics, and pressing key Backspace or key Delete deletes the letter and all its diacritics.

- Now let's replace "1978 عام" (*'am*, year 1978) by "1978-05-21 يوم" (*yom*, day 1978-05-21). By simply typing "`1978-05-21`", we'll get this, which is not what we want:

This is normal because character "-" which separates day from month from year is given by the Unicode Bidirectional Algorithm a **RTL** directionality[6], hence character "-" "breaks" the desired "`1978-05-21`" left-to-right character sequence.

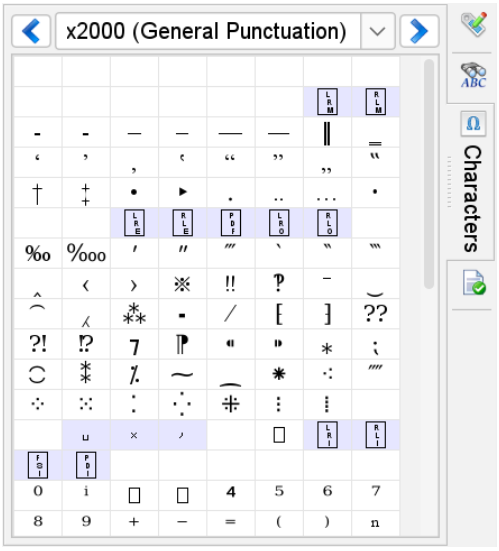
This problem occurs quite often because many commonly used characters: quotes, parentheses, etc, behave just like character "-". There are two ways to solve this problem:

1. The simple way. Select text displayed as "`21-05-1978`"[7] and use the **Edit** tool in *XMLmind XML Editor - Online Help* to convert it to a span element of some sort (i.e. XHTML `bdi` or `span`, DITA `ph`, DocBook `phrase`) having an **LTR** direction.



2. The hard way. Insert an **LRE** character[8] before "`1978`" and a **PDF** character[9] after "`21`".

Such "*Explicit Directional Overrides*" characters are found in the **Characters** tool in *XMLmind XML Editor - Online Help*:



3. Limitations

- In a right-to-left table, the order of columns is not inverted. For a right-to-left table, column zero should be on the right side and not at the left side like for left-to-right tables.

- Only the tree view and styled (with or without visible tags) view have bidirectional script support. The "**XML source**" view in *XMLmind XML Editor - Online Help* has no bidirectional script support whatsoever.

- You should expect poor results if you attempt to convert documents containing some **RTL** text or a mix of **RTL** and **LTR** text, using any of the menu items of the "**Convert Document**" sub-menus (for

[6] Notice its dark magenta color.
[7] Using the mouse to select alternating left-to-right and right-to-left character sequences is not easy. Prefer to use Shift+Click, which extends the selection, to do that.
[8] `U+202D`, LEFT-TO-RIGHT OVERRIDE, force following characters to be treated as strong left-to-right characters.
[9] `U+202C`, POP DIRECTIONAL FORMATTING, end the scope of the last **LRE**, **RLE**, **RLO**, or **LRO**.

example, **DocBook → Convert Document** in *XMLmind XML Editor - DocBook Support*). The reasons for this limitation are:
- The XSLT stylesheets invoked by these menu items have limited (e.g. the DocBook XSL stylesheets) or no bidirectional script support (e.g. our own DITA and XHTML XSL stylesheets).
- The XSL-FO processors invoked by some of these menu items have buggy (e.g. Apache FOP) or no bidirectional script support (e.g. our own XMLmind XSL-FO Converter).

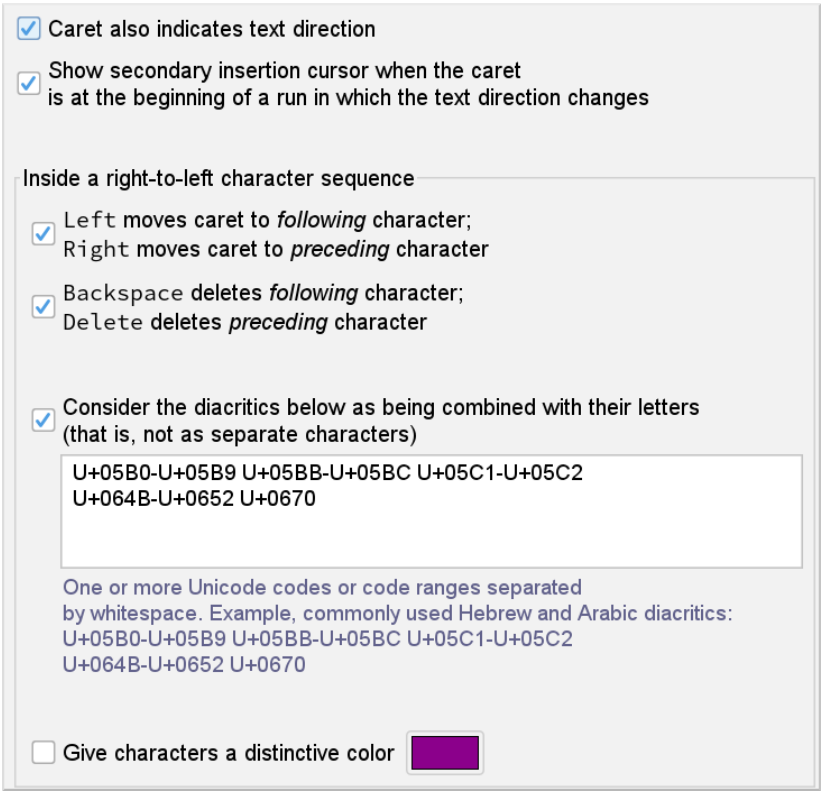
- XHTML `bdo` is rendered on screen by **XXE** just like `bdi`, that is, `bdo` will not be shown overriding the inherent directionality of characters. For example, `<bdo dir="rtl">1978</bdo>` is *not* rendered on screen as "`8791`", as it should be.

  Similarly, DITA and DocBook `dir="rlo"` is rendered just like `dir="rtl"` and `dir="lro"` is rendered just like `dir="ltr"`.

4. Preferences

Installing the "**Bidi Support**" add-on adds a preferences sheet to the **Preferences** dialog in *XMLmind XML Editor - Online Help* box (**Options → Preferences** in *XMLmind XML Editor - Online Help*).

*Figure 3. The "**Bidi Support**" preference sheet*



IMPORTANT: the following options have an effect only inside an **RTL** character sequence or a text node containing both **RTL** and **LTR** character sequences.

Caret also indicates text direction

If this option is turned on, the insertion cursor (caret) changes its shape and is given a small arrow which indicates the directionality of the character following the caret. More information [4].

Default: option turned on.

Show secondary insertion cursor

Ignored unless option "**Caret also indicates text direction**" is turned on. If this option is turned on, when relevant, **XXE** displays a secondary insertion cursor in addition to the actual caret. More information [5].

Default: option turned on.

IMPORTANT: the following options have an effect only inside an **RTL** character sequence.

Left moves caret to following character

If this option is turned on, pressing key Left key moves the caret to the left, that is, to the following character in the **RTL** character sequence and pressing key Right moves the caret to the right, that is, to the preceding character in the **RTL** character sequence. More information [4].

Default: option turned on.

Backspace deletes following character

If this option is turned on, pressing key Backspace deletes the character found at the left of the caret, that is, deletes the following character in the **RTL** character sequence and pressing key Delete[5] deletes the character found at the right of the caret, that is, deletes the preceding character in the **RTL** character sequence. More information [4].

Default: option turned on.

Consider the diacritics below as being combined with their letters

If this option is turned on, **XXE** considers that a letter and all its diacritics have been combined and now form a single editable entity. Therefore pressing key Left or key Right skips the letter and all its diacritics, and pressing key Backspace or key Delete deletes the letter and all its diacritics. More information [5].

Diacritics to be considered as combined with their letters are specified in the text field below the check-box. Diacritics must be specified as the Unicode value of a single character, example: `U+0670` (ألف خنجرية, dagger alif), or as a character range, example: `U+064B-U+0652`.

Default: option turned on. Default specification of the diacritics: the most common Hebrew diacritics: "`U+05B0-U+05B9 U+05BB-U+05BC U+05C1-U+05C2`" and the most common Arabic diacritics: "`U+064B-U+0652 U+0670`".

Give characters a distinctive color

If this option is turned on, give **RTL** text runs a distinctive color. This distinctive color is chosen using the color chooser displayed by clicking the “color button” found at the right of this check-box.

This feature is especially useful to determine whether a character having a neutral directionality (punctuation, no-break-space, etc) is considered by **XXE** as being **RTL** or on the contrary, as being **LTR**. More information [4].

Default: option turned off.

A. Typing Arabic (اللغة العربية) or Hebrew (עברית) when you don't have the corresponding keyboard

Example A.1. Typing Arabic on a Windows computer

- Click **Start** Menu. Click **Settings**. Click "**Time & Language**". Click **Language**.
- Click "**Add a language**". Choose the language to be installed, for example: "العربية (مصر), Arabic (Egypt)".
- Click **Next**. Turn off the **Text-to-speech** option. Click **Install**.
- After the language pack is downloaded and installed, press the Windows logo key+**Ctrl**+**O** to turn the On-Screen Keyboard on or off. (This is documented in **Settings**, "**Ease of Access**", **Keyboard**, "**Use the On-Screen Keyboard**".)

Example A.2. Typing Hebrew on a Mac

- Go to the **Apple** menu. Select "**System Preferences**". Click **Keyboard**.
- Select the "**Input Sources**" tab, click the "+" button.
- Select **א Hebrew** from the list. Then Select **א Hebrew** from the list on the right. Click **Add**.
- Check the box next to "**Show Input menu in menu bar**".
- After selecting **א Hebrew** from the menu bar, select "**Show Keyboard Viewer**" to display the on-screen keyboard.

B. Enabling bidi support in your custom XXE configuration

Normal users are not supposed to do this. The intended audience for this appendix is consultants and “local gurus”.

1. The `directionalityFinder` configuration element

You have written a custom **XXE** configuration in *XMLmind XML Editor - Configuration and Deployment* in order to teach **XXE** about your custom schema. Enabling bidi support in your custom configuration simply consists in adding a `directionalityFinder` configuration element in *XMLmind XML Editor - Configuration and Deployment* to your configuration.

DocBook example:

```
<directionalityFinder>
  <class>com.xmlmind.xmledit.edit.HTMLDirectionalityFinder</class>
  <property name="options" type="String" value="dir ltr rtl lro rlo" />
</directionalityFinder>
```

TEI Lite example:

```
<!-- Full TEI also has style="direction: rtl; unicode-bidi: embed" -->
<directionalityFinder>
  <class>com.xmlmind.xmledit.edit.HTMLDirectionalityFinder</class>
  <property name="options" type="String" value="xml:lang" />
</directionalityFinder>
```

The `directionalityFinder` configuration element is documented in "*XMLmind XML Editor - Configuration and Deployment*".

2. The "Set dir" button

If your custom schema has a text directionality attribute similar to XHTML, DITA or DocBook, global, “inherited”, `dir` attribute then it may be useful to add the "**Set dir**" button to your custom toolbar.

This is done by adding the following `button` configuration element to your `toolBar` configuration element in *XMLmind XML Editor - Configuration and Deployment*. DITA/DocBook example:

```
<button icon="xxe-config:common/icons/cancel.png">❶
  <class>com.xmlmind.xmleditext.bidi.DirTool</class>❷
  <property name="options" type="String"
            value="ltr=ltr|lro rtl=rtl|rlo dirRoots=-" />❸
  <command name="pass" />❹
</button>
```

❶❹ Attribute `icon` and child element `command` are needed only for a successful validation of the configuration file. These specifications are not actually used by `DirTool`.

❷ Implementation of the `DirTool` custom control in *XMLmind XML Editor - Configuration and Deployment*.

❸ The options property is used to parametrize the `DirTool` custom control. These options are used to determine what `DirTool` is to display for implicitly or explicitly selected element. Is it left-to-right, right-to-left or unknown ?

The default value of options reflects the specificities of the `dir` attribute in XHTML documents:

```
dir=dir ltr=ltr rtl=rtl¬
dirRoots={http://www.w3.org/1999/xhtml}bdi|{http://www.w3.org/1999/xhtml}bdo
```

In the case of the DITA or DocBook `dir` attribute, some, but not all, of the above default options must be overridden:

```
ltr=ltr|lro rtl=rtl|rlo dirRoots=-
```

Options are:

`dir` [= *`attribute_name`*]?
: The name of a text directionality attribute similar to XHTML, DITA or DocBook, global, “inherited”, `dir` attribute. Default option: `dir=dir`, or equivalently just `dir`.

`ltr` [= *`attribute_value`* [| *`attribute_value`*]*]?
: One or more attribute values meaning left-to-right. Multiples values are separated by character '`|`'. Default option: `ltr=ltr`, or equivalently just `ltr`.

`rtl` [= *`attribute_value`* [| *`attribute_value`*]*]?
: One or more attribute values meaning right-to-left. Multiples values are separated by character '`|`'. Default option: `rtl=rtl`, or equivalently just `rtl`.

`dirRoots` [= *`element_name`* [| *`element_name`*]*]?

Names of one or more elements for which the text directionality attribute is *not* inherited. Multiples names are separated by character '`|`'. Special value "`-`" may be used to specify: no such elements. Default option: `dirRoots={http://www.w3.org/1999/xhtml}bdi|{http://www.w3.org/1999/xhtml}bdo`, or equivalently just `dirRoots`.

Note that it is not possible to use namespace prefixes for the attribute and element names specified in the options property. Notation `{`*`namespace_URI`*`}`*`local_name`* —the so-called James Clark's notation— must be used instead.