

FOX Widget Reference

FOX:Widget:WIDGET REFERENCE

- alive
 - alive+
 - button
 - button+
 - cartographic
 - date
 - date+
 - datetime
 - datetime+
 - file
 - file-new
 - file-upload
 - html
 - html+
 - icon
 - iconprompt
 - input
 - input+
 - input-resizable
 - image
 - link
 - link+
 - mailto
 - mailto-btn
 - obscure
 - password
 - print
 - radio
 - selector
 - selector+
 - static-text
 - textbox
 - timer+
 - url
 - url-btn
-

FOX:Widget:alive

Alive Widget

Description

Shows link for action defined on element if *namespace:ro* and *namespace:run* XPath resolves to true.

Shows menu-out actions as links if *namespace:run* XPath resolves to true.

Examples

Read Only & Run

```
<xs:element name="ALIVE_RO_RUN" alive:ro="." alive:run="." fox:widget="alive" fox:action="action-dummy"/>
```

[Alive Ro Run](#)

Menu Run

```
<fm:action name="menu-alive" alive:run="." alive:widget="alive">
```

[Menu-alive](#)

FOX:Widget:alive+

Alive+ Widget

Description

Shows an active link for action defined on element if *namespace:ro* and *namespace:run* XPath resolves to true.

Shows a non-active link if only *namespace:ro* is true.

Shows menu-out actions as links if *namespace:run* XPath resolves to true.

Examples

Read Only

```
<xs:element name="ALIVE_PLUS_RO" alive:ro="." fox:widget="alive+" fox:action="action-dummy"/>
```

[Alive Plus Ro](#)

Read Only & Run

```
<xs:element name="ALIVE_PLUS_RO_RUN" alive:ro="." alive:run="." fox:widget="alive+" fox:action="action-dummy"/>
```

Alive Plus Ro Run

Menu Run

```
<fm:action name="menu-alive-plus" alive:run="." alive:widget="alive+">
```

Menu-alive-plus

FOX:Widget:button

Button Widget

Description

Allows the user to run an action by clicking an area on the screen.

When used in conjunction with *fox:prompt*, instead of the *action-name* attribute's value appearing within the button's text area, the prompt's value appears instead.

A button will automatically re-size itself to match the length of the text within the button.

Example

```
<fm:action name="action-search" search:run="." fox:widget="button" fox:prompt="Search">  
  <fm:do/>  
</fm:action>
```

The above code will produce this:

When clicked, the FOX engine will execute whatever is in the appropriate *fm:do*.

FOX:Widget:button+

Button+ Widget

Description

Shows an button for the action defined on an element if *namespace:ro* and *namespace:run* XPath resolves to true.

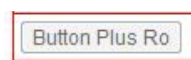
Shows a non-active button if only *namespace:ro* is true.

Shows menu-out actions as buttons if *namespace:run* XPath resolves to true.

Examples

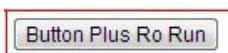
Read Only

```
<xs:element name="BUTTON_PLUS_RO" button:ro="." fox:widget="button+" fox:action="action-dummy"/>
```



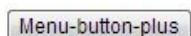
Read Only & Run

```
<xs:element name="BUTTON_PLUS_RO_RUN" button:ro="." button:run="." fox:widget="button+" fox:action="action-dummy"/>
```



Menu Run

```
<fm:action name="menu-button-plus" button:run="." button:widget="button+">  
  <fm:do/>  
</fm:action>
```



FOX:Widget:date

Date Widget

Description



Allows for input of dates. This is the default widget for type="xs:date" elements.

When the item is read-only, this widget will just display as standard text.

Examples

Edit

```
<xs:element name="START_DATE" type="xs:date" fox:edit="." />
```

Start Date  

Read-only

```
<xs:element name="START_DATE" type="xs:date" fox:ro="." />
```

Start Date 04-MAY-2010

FOX:Widget:date+

Date+ Widget

Description



Allows for input of dates. This is the default widget for type="xs:date" elements.

When the item is read-only, this widget will display as a disabled input field.

Examples

Edit

```
<xs:element name="START_DATE" type="xs:date" fox:widget="date+" fox:edit="." />
```

Start Date  

Read-only

```
<xs:element name="START_DATE" type="xs:date" fox:widget="date+" fox:edit="." />
```

Start Date

FOX:Widget:datetime

Datetime Widget

Description

Allows for input of dates with their full time component show.

Shows editable datetime field when *namespace:edit* resolves to true.

Shows plain text representation of datetime when *namespace:ro* resolves to true.

Examples

Edit

```
<xs:element name="DATE_TIME_EDIT" type="xs:dateTime" default="2006-01-01T19:33:14" date:edit="." fox:widget="datetime"/>
```

Read Only

```
<xs:element name="DATE_TIME_RO" type="xs:dateTime" default="2006-01-01T19:33:14" date:ro="." fox:widget="datetime"/>
```

FOX:Widget:datetime+

Datetime+ Widget

Description

Allows for input of dates with their full time component show.



Shows editable datetime field when *namespace:edit* resolves to true.

Shows non-selectable datetime field when *namespace:ro* resolves to true.

Examples


Edit

```
<xs:element name="DATE_TIME_PLUS_EDIT" type="xs:dateTime" default="2006-01-01T19:33:14" date:edit="." fox:widget="datetime+"/>
```

Read Only

```
<xs:element name="DATE_TIME_PLUS_RO" type="xs:dateTime" default="2006-01-01T19:33:14" date:ro="." fox:widget="datetime+"/>
```



FOX:Widget:file

File Widget

Description

Allows for uploading of a file.

Schema Attributes

- fox:upload-button-flow
 - fox:upload-button-position
 - fox:upload-capture-field1
 - fox:upload-capture-field2
 - fox:upload-capture-field3
 - fox:upload-capture-field4
 - fox:upload-capture-field5
 - fox:upload-choose-prompt
 - fox:upload-display-field1
 - fox:upload-display-field2
 - fox:upload-display-field3
 - fox:upload-display-field4
 - fox:upload-display-field5
 - fox:upload-enable-download
 - fox:upload-enable-view
 - fox:upload-enable-description
 - fox:upload-fail-action
 - fox:upload-field-flow
 - fox:upload-file-type
 - fox:upload-metadata
 - fox:upload-mode
 - fox:upload-prompt
 - fox:upload-prompt-field1
 - fox:upload-prompt-field2
 - fox:upload-prompt-field3
 - fox:upload-prompt-field4
 - fox:upload-prompt-field5
 - fox:upload-save-prompt
 - fox:upload-success-action
 - fox:upload-view-prompt
 - fox:upload-window-type
-

FOX:Widget:html

HTML Widget

Description

Allows for user stylized user input, whereby the user can view and format their text via a text box gui or the raw HTML

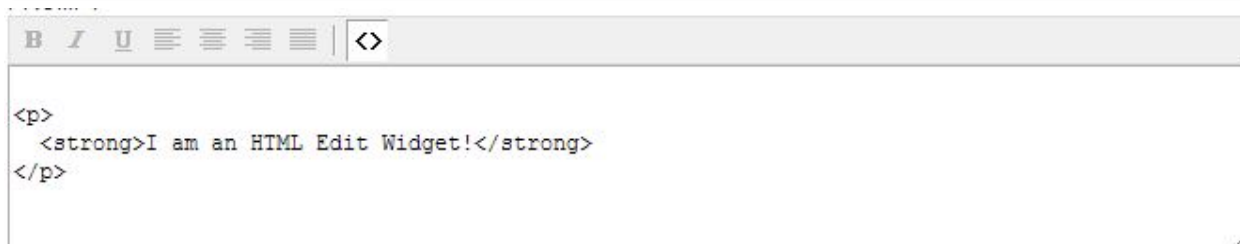
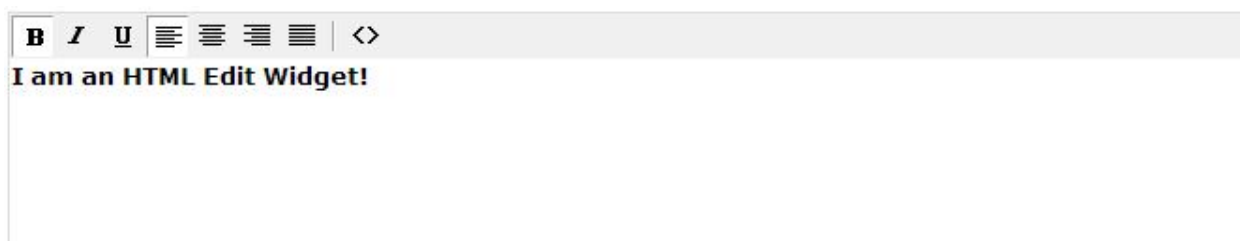
When *namespace:ro* resolves to true, no text area will be displayed.

When *namespace:edit* resolves to true, an editable text area which can switch to html view will display.

Examples

Edit

```
<xs:element name="MY_HTML_EDIT" fox:edit="." fox:widget="html"/>
```



Read-only

```
<xs:element name="MY_HTML_RO" fox:ro="." fox:widget="html"/>
```

No image output.

FOX:Widget:html+

HTML Plus Widget

Description

Allows for user stylized user input, whereby the user can view and format their text via a text box gui or the raw HTML.

When *namespace:ro* resolves to true, a non-selectable text area will display.

When *namespace:edit* resolves to true, an editable text area which can switch to html view will display.

Examples

Edit

```
<xs:element name="MY_HTML_PLUS_EDIT" fox:edit="." fox:widget="html+"/>
```

The screenshot shows a text area with a toolbar at the top. The toolbar contains icons for Bold (B), Italic (I), Underline (U), Bulleted List, Numbered List, Indent Left, Indent Right, and a source code icon (<>). The text area contains the text "I am an HTML Plus Edit Widget!". Below the text area, the raw HTML code is displayed:

```
<p>
  <strong>I am an HTML Plus Edit Widget!</strong>
</p>
```

Read-only

```
<xs:element name="MY_HTML_PLUS_RO" fox:ro="." fox:widget="html+"/>
```

The screenshot shows a read-only text area with a toolbar at the top. The toolbar contains icons for Bold (B), Italic (I), Underline (U), Bulleted List, Numbered List, Indent Left, Indent Right, and a source code icon (<>). The text area contains the text "I am an HTML Plus Read Only Widget!". The text area is shaded gray, indicating it is read-only.

FOX:Widget:icon

Icon Widget

Description

Deprecated, use *image* widget type instead.

Alternatively, use a button plus with *namespace:run* set to false and the image defined though *imgURL*.

FOX:Widget:iconprompt

Icon Widget

Description

Deprecated along with *icon*.

FOX:Widget:image

Image Widget

Description

Allows for uploading of a image which will then be resized and a thumbnail displayed on the page according to the related schema attributes.

Schema Attributes

- fox:image-display-height
- fox:image-display-size
- fox:image-display-width
- fox:image-enable-preview
- fox:image-enable-rotate
- fox:image-only

Also inherits the schema attributes from the file widget.

FOX:Widget:input

Input Widget

Description

Allows for user input. This is the default widget for most cases (it is superceded if a more precise attribute is specified like type="xs:date" then the date widget is used instead).

When the item is read-only, this widget will just display as standard text.

Examples

Edit

```
<xs:element name="INPUT_EDIT" type="xs:string" fox:fieldWidth="20" fox:fieldHeight="1" fox:edit="."/>
```

Input Edit	<input type="text" value="Input Edit"/>
------------	---

Read-only

```
<xs:element name="INPUT_RO" type="xs:string" fox:fieldWidth="20" fox:fieldHeight="1" fox:ro="."/>
```

Input Ro	Input Ro
----------	----------

FOX:Widget:input+

Input+ Widget

Description

Allows for user input.

When the item is read-only, this widget will display as a disabled input field.

Examples

Edit

```
<xs:element name="FORENAME" type="xs:string" fox:widget="input+" fox:fieldWidth="20" fox:fieldHeight="1" fox:edit="."/>
```

Input Plus Edit	<input type="text" value="Input Plus Edit"/>
-----------------	--

Read-only

```
<xs:element name="FORENAME" type="xs:string" fox:widget="input+" fox:fieldWidth="20" fox:fieldHeight="1" fox:ro="." />
```

Input Plus Ro	Input Plus Ro
---------------	---------------

FOX:Widget:input-resizable

Input-Resizable Widget

Description

Works in much the same way as to the *input* Widget but with the added feature of a resizable text area.

When *namespace:ro* resolves to true, the text contents are displayed.

When *namespace:edit* resolves to true, a resizable and editable text area is displayed.

Examples

Edit

```
<xs:element name="INPUT_RESIZABLE_RO" fox:ro="." fox:widget="input-resizable"/>
```

Read Only

Read-only

```
<xs:element name="INPUT_RESIZABLE_EDIT" fox:edit="." fox:widget="input-resizable"/>
```

Edit

FOX:Widget:link

Link Widget

Description

Displays a link. This is the default widget for menu-out.

When the item is not runnable, this widget will display nothing.

Note: In a form set-out no prompt is displayed.

Examples

Runnable

```
<xs:element name="LINK_RO_RUN" link:ro="." link:run="." fox:widget="link" fox:action="action-dummy"/>
```

[Link Ro Run](#)

Not Runnable

```
<xs:element name="LINK_RO" link:ro="." fox:widget="link" fox:action="action-dummy"/>
```

No Output

FOX:Widget:link+

Link+ Widget

Description

Displays a link.

When the item is not runnable, this widget will just display normal text.

Note: In a form set-out no prompt is displayed.

Examples

Runnable

```
<xs:element name="LINK_PLUS_RO_RUN" link:ro="." link:run="." fox:widget="link+" fox:action="action-dummy"/>
```

[Link Plus Ro Run](#)

Not Runnable

```
<xs:element name="LINK_PLUS_RO_RUN" link:ro="." link:run="." fox:widget="link+" fox:action="action-dummy"/>
```

Link Plus Ro

FOX:Widget:mailto

Mailto Widget

Description

Displays a mailto link which will open up an email window to the address specified in the local email client.

When *namespace:run* resolves to true, a mailto link will be displayed.

When *namespace:run* resolves to false, a text representation of the link will display.

Examples

Runnable

```
<xs:element name="MAILTO_RUN" fox:ro="." fox:run="." fox:widget="mailto" fox:email="joe@bloggs.com" fox:prompt="Mailto Me!"/>
```

[Mailto Me!](mailto:joe@bloggs.com)

Not Runnable

```
<xs:element name="MAILTO_NOT_RUN" fox:ro="." fox:run="not(.)" fox:widget="mailto" fox:email="joe@bloggs.com" fox:prompt="Don't Mailto Me!"/>
```

Don't Mailto Me!

FOX:Widget:mailto-btn

Mailto-btn Widget

Description

Displays a mailto button which will open up an email window to the address specified in the local email client.

When *namespace:run* resolves to true, a mailto button will be displayed.

When *namespace:run* resolves to false, a text representation of the link will display.

Examples

Runnable

```
<xs:element name="MAILTO_RUN" fox:ro="." fox:run="." fox:widget="mailto-btn" fox:email="joe@bloggs.com" fox:prompt="Mailto Me!"/>
```



Not Runnable

```
<xs:element name="MAILTO_NOT_RUN" fox:ro="." fox:run="not(.)" fox:widget="mailto-btn" fox:email="joe@bloggs.com" fox:prompt="Don't Mailto Me!"/>
```



FOX:Widget:obscure

Obscure Widget

Description

Displays the text content of the element as an obscured image (similar to a Captcha), this works best with 6 characters.

When either *namespace:ro* or *namespace:edit* resolves to true, a scrambled image of the text will appear.

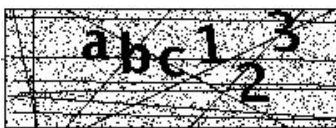
Examples

```
<xs:element name="OBSCURE" fox:ro="." fox:widget="obscure" fox:prompt="TEST"/>
```

With init of

```
<fm:assign initTarget=":{theme}/OBSCURE" textValue="abc123"/>
```

Will display



FOX:Widget:password

Password Widget

Description

Displays a password input field.

When *namespace:ro* resolves to true, no input field will appear.

When *namespace:edit* resolves to true, a password field will display.

Examples

Edit

```
<xs:element name="PASSWORD_EDIT" fox:edit="." fox:widget="password"/>
```

A screenshot of a password input field. The field is a horizontal rectangle with a thin border. Inside the field, there are four dots (••••) representing masked characters. The field is positioned on a light gray background.

Read-only

```
<xs:element name="PASSWORD_RO" fox:ro="." fox:widget="password"/>
```

No output

FOX:Widget:print

Print Widget

A print widget is a command button which invokes a client-side printing function. It does not submit the page, so any action which uses a print widget should have an empty `<fm:do>` block.

FOX:Widget:radio

Radio Widget

Description

The radio widget displays radio values options of which one selection can be chosen. The behavior of a radio widget depends on the contents the widget is displaying (boolean, mapset or enumeration).

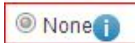
namespace:ro will display the current selected value (or null key) as a non-selectable radio box.

namespace:edit allows for selection of a single radio box.

Examples

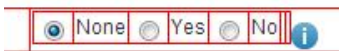
Boolean Read Only

```
<xs:element name="RADIO_BOOL_RO" type="xs:boolean" selector:ro="." fox:widget="radio"/>
```



Boolean Edit

```
<xs:element name="RADIO_BOOL_EDIT" type="xs:boolean" selector:edit="." fox:widget="radio"/>
```



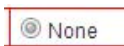
Boolean Edit Down

```
<xs:element name="RADIO_BOOL_EDIT_DOWN" type="xs:boolean" selector:edit="." fox:widget="radio" fox:flow="down"/>
```



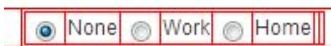
Mapset Read Only

```
<xs:element name="RADIO_MAPSET_RO" type="xs:string" selector:ro="." fox:widget="radio" fox:map-set="ENV_MAPSETS_PERSONAL_EMAIL_TYPE"/>
```



Mapset Edit

```
<xs:element name="RADIO_MAPSET_EDIT" type="xs:string" selector:edit="." fox:widget="radio" fox:map-set="ENV_MAPSETS_PERSONAL_EMAIL_TYPE"/>
```




Mapset Edit Down

```
<xs:element name="RADIO_MAPSET_EDIT_DOWN" type="xs:string" selector:edit="." fox:widget="radio" fox:map-set="ENV_MAPSETS_PERSONAL_EMAIL_TYPE" fox:flow="down"/>
```

<input checked="" type="radio"/>	None
<input type="radio"/>	Work
<input type="radio"/>	Home


Enumeration Read Only

```
<xs:element name="RADIO_MULTI_ENUMERATION_RO" selector:ro="." fox:widget="radio">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONE"/>
      <xs:enumeration value="TWO"/>
      <xs:enumeration value="THREE"/>
      <xs:enumeration value="FOUR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

<input checked="" type="radio"/>	None	
----------------------------------	------	---

Enumeration Edit

```
<xs:element name="RADIO_MULTI_ENUMERATION_EDIT" selector:edit="." fox:widget="radio">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONE"/>
      <xs:enumeration value="TWO"/>
      <xs:enumeration value="THREE"/>
      <xs:enumeration value="FOUR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

<input checked="" type="radio"/>	None	<input type="radio"/>	ONE	
<input type="radio"/>	TWO	<input type="radio"/>	THREE	
<input type="radio"/>	FOUR			

FOX:Widget:selector

Selector Widget

Description

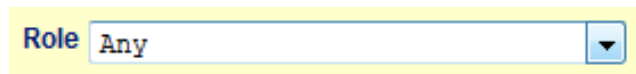
Allows for user input using a drop-down list. This is the default widget for elements linked to a mapset.

When the item is read-only, this widget will just display as standard text.

Examples

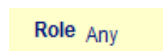
Edit

```
<xs:element name="ROLE" type="xs:string" fox:map-set="ms-DASH008L-webRoles" fox:key-null="Any" fox:edit="." />
```

A screenshot of a web form element. It consists of a label 'Role' followed by a text input field containing the word 'Any'. To the right of the input field is a small blue downward-pointing arrow icon, indicating a dropdown menu. The entire element is highlighted with a yellow background.

Read-only

```
<xs:element name="ROLE" type="xs:string" fox:map-set="ms-DASH008L-webRoles" fox:key-null="Any" fox:ro="." />
```

A screenshot of a web form element. It consists of a label 'Role' followed by the word 'Any'. The entire element is highlighted with a yellow background.

FOX:Widget:selector+

Selector+ Widget

Description

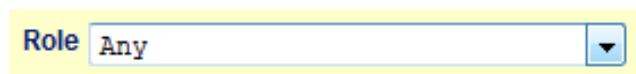
Allows for user input using a drop-down list.

When the item is read-only, this widget will just display as a greyed out, uneditable drop-down selector.

Examples

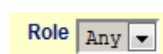
Edit

```
<xs:element name="ROLE" type="xs:string" fox:widget="selector+" fox:map-set="ms-DASH008L-webRoles" fox:key-null="Any" fox:edit="." />
```

A screenshot of a web form element. It consists of a label 'Role' followed by a text input field containing the word 'Any'. To the right of the input field is a small blue downward-pointing arrow icon, indicating a dropdown menu. The entire element is highlighted with a yellow background.

Read-only

```
<xs:element name="ROLE" type="xs:string" fox:widget="selector+" fox:map-set="ms-DASH008L-webRoles" fox:key-null="Any" fox:ro="." />
```

A screenshot of a web form element. It consists of a label 'Role' followed by a greyed-out dropdown menu containing the word 'Any'. The entire element is highlighted with a yellow background.

FOX:Widget:static-text

Static-Text Widget

Description

Displays a prompt on a phantom element when the phantoms root element is initialised and set out.

When *namespace:ro* or *namespace:edit* resolves to true, text from the phantom's prompt will display.

Examples

```
<xs:element name="phantom-static" fox:ro="." type="phantom" fox:widget="static-text" fox:prompt="Static Text." />
```

Given this schema.

```
<xs:element name="STATIC_ROOT" fox:ro=".">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EMPTY_ONE" fox:ro="." />
      <xs:element name="EMPTY_TWO" fox:ro="." />
      <xs:element name="EMPTY_THREE" fox:ro="." />
      <xs:element name="phantom-static" fox:ro="." type="phantom" fox:widget="static-text" fox:prompt="Static Text." />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

This init

```
<fm:init target=":{theme}/STATIC_ROOT" for-schema="*/*" />
```

This set-out

```
<fm:set-out fox:mode="." match=":{theme}/STATIC_ROOT" />
```

This will be displayed. **Static Text.**

FOX:Widget:checkbox

Checkbox Widget

Description

The checkbox widget displays an array of checkboxes of which the user can select any combination of. The behavior of a checkbox depends on the contents the widget is displaying (boolean, mapset or enumeration).

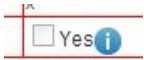
namespace:ro will display the current selected value (or null key) as a non-selectable checkbox.

namespace:edit allows for selection of any number of checkboxes.

Examples

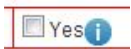
Boolean Read Only

```
<xs:element name="TICKBOX_BOOL_RO" type="xs:boolean" selector:ro="." fox:widget="checkbox"/>
```



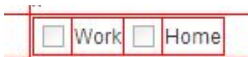
Boolean Edit

```
<xs:element name="TICKBOX_BOOL_EDIT" type="xs:boolean" selector:edit="." fox:widget="checkbox"/>
```



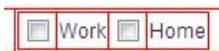
Mapset Read Only

```
<xs:element name="TICKBOX_MAPSET_RO" type="xs:string" selector:ro="." fox:widget="checkbox" fox:map-set="ENV_MAPSETS_PERSONAL_EMAIL_TYPE"/>
```



Mapset Edit

```
<xs:element name="TICKBOX_MAPSET_EDIT" type="xs:string" selector:edit="." fox:widget="checkbox" fox:map-set="ENV_MAPSETS_PERSONAL_EMAIL_TYPE"/>
```



Enumeration Read Only

```
<xs:element name="TICKBOX_MULTI_ENUMERATION_RO" selector:ro="." fox:widget="checkbox" fox:formMaxCols="2">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONE"/>
      <xs:enumeration value="TWO"/>
      <xs:enumeration value="THREE"/>
      <xs:enumeration value="FOUR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

<input type="checkbox"/>	ONE	<input type="checkbox"/>	TWO
<input type="checkbox"/>	THREE	<input type="checkbox"/>	FOUR

Enumeration Edit

```
<xs:element name="TICKBOX_MULTI_ENUMERATION_EDIT" selector:edit="." fox:widget="checkbox" fox:formMaxCols="2">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="ONE"/>
      <xs:enumeration value="TWO"/>
      <xs:enumeration value="THREE"/>
      <xs:enumeration value="FOUR"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

<input type="checkbox"/>	ONE	<input type="checkbox"/>	TWO
<input type="checkbox"/>	THREE	<input type="checkbox"/>	FOUR

FOX:Widget:timer+

Timer+ Widget

Description

Calls a specified action after a pre-set amount of time has elapsed.

Remember to ensure that any database data varying the timer amount is stored in the format "MM:SS"

Example

Timer set to 30 seconds, calling action-do-something

```
<xs:element name="My_Timer" type="xs:string" default="0:30" fox:widget="timer+" fox:run="." fox:action="action-do-something" fox:fieldHeight="1" fox:fieldWidth="10" fox:prompt="Time Remaining" test:row="1">
```

FOX:Widget:url

URL Widget

Description

Displays a link which will open up a new windows with the href supplied. (effectively ``)

When *namespace:run* resolves to true, a link will be displayed.

When *namespace:run* resolves to false, a text representation of the link will display.

Examples

Runnable

```
<xs:element name="URL_RUN" fox:ro="." fox:run="." fox:widget="url" fox:href="http://www.google.com" fox:prompt="Click Me!"/>
```

[Click Me!](#)

Not Runnable

```
<xs:element name="URL_NOT_RUN" fox:ro="." fox:run="not(.)" fox:widget="url" fox:href="http://www.google.com" fox:prompt="Can't Click Me!"/>
```

Cant Click Me!

FOX:Widget:url-btn

URL-btn Widget

Description

Displays a button which will open up a new windows with the href supplied. (effectively ``)

When *namespace:run* resolves to true, a button will be displayed.

When *namespace:run* resolves to false, a non-selectable button will display.

Examples

Runnable

```
<xs:element name="URL_RUN" fox:ro="." fox:run="." fox:widget="url-btn" fox:href="http://www.google.com" fox:prompt="Click Me!"/>
```


Not Runnable

```
<xs:element name="URL_NOT_RUN" fox:ro="." fox:run="not(.)" fox:widget="url-btn" fox:href="http://www.google.com" fox:prompt="Can't Click Me!"/>
```

Can't Click Me!