

Syntax and Parameter reference

Downloaded 30/08/2010

Standard parametes / attributes for [all widgets](#)

Name in Editor	xml attr	xul	internal name	description of the attribut
Type	type <i>(only for sub-types)</i>	no	type	The type of the widget is stored in the tag itself, some widgets have sub-types (unknow to XUL). In that case the subtype is stored in the type attribut. The Type / tag has to be either the name of a standard eTemplate-widget or of an already existing extension .
Name	id	yes	name	<p>A string to locate the content for the widget in the content array (index) to show the dialog and for the returned content. Can be left blank or be obmitted as xml-attribut if the widget needs no content.</p> <p>The name can contain the following variables, which gets expanded before they are used as array index (for an example see the Tutorial):</p> <p>\$c the column-number (starting with 0, if you have a header, data-cells start at 1)</p> <p>\$col the column-letter: 'A', 'B', 'C', ...</p> <p>\$row the row-number (starting with 0, if you have a header, data-cells start at 1)</p> <p>\$cont the content-array the (sub-)template, on auto-repeated row's this could eg. be used to generate button-names with id-values in it: "del[\$cont[id]]" expands to "del[123]", if \$cont = array('id' => 123)</p> <p>\$row_cont the sub-array indexed by \$row of the content-array, on auto-repeated row's this could eg. be used to generate button-names with id-values in it: "del[\$row_cont[id]]" expands to "del[123]", if \$cont = array('1' => array('id' => 123), '2' => array('id' => 456)) and \$row = 1</p> <p>\$c_ are the respective values of the previous template-inclusion, eg. the column-headers</p>

				<p>\$col_ in the eTemplate-editor are templates itself, to show the column-name in the header you can not use \$col as it will be constant as it is always the same col in the header-template, what you want is the value of the previous template-inclusion.</p> <p>\$row_</p>
Label	label	no	label	<p>The label is displayed by default in front (for radiobuttons behind) each widget (if not empty). If you want to specify a different position, use a '%s' in the label, which gets replaced by the widget itself. Eg. '%s Name' to have the label Name behind a checkbox. The label can contain variables, as descript for name. If the label starts with a '@' it is replaced by the value of the content-array at this index (with the '@'-removed and after expanding the variables).</p> <p>Note: The label gets always translated, if its longer than 1 char! If this is not disired, use a label widget, place the not-to-translate label in the content-array <u>and</u> check NoTranslation or set the xml attribute no_lang.</p>
Help	statustext	yes	help	<p>This text / help-message is displayed in the status-bar of the browser when the widget gets focus (or as tooltip for buttons or general in gtk). If the user has JavaScript switched off, the help-texts get NOT submitted, as this is detected. If the helptext starts with a '@' it is replaced by the value of the content-array at this index (with the '@'-removed and after expanding the variables).</p>
Options	?	?	size	<p>This attribute controls certain aspects of the different widgets. It's meaning and xml / xul-values are document with the widgets. If the options-string starts with a '@' it is replaced by the value of the content-array at this index (with the '@'-removed and after expanding the variables).</p>
NoTranslation	no_lang	no	no_lang	<p>If checked the content of the widget and the label gets NOT translated. The helptext of a widget is always translated.</p>
needed	needed	no	needed	<p>If checked (xml-attr: needed="1") the etemplates will reprompt the user if he left the widget / field empty.</p>
Readonly	readonly	yes	readonly	<p>If checked (xml-attr: readonly="true") the widget will NOT be editable. If it is not supported by the browser, the etemplate-class makes shure that no changes / content is transmitted back to the app. Only applicable to widgets with input capabilities. Readonly Buttons get removed from the dialog, like they where disabled. The readonly attribute could be set on runtime via a readonly-array sublied to the exec or show function of the class, the value on index=name/id has to be true to make a widget readonly.</p>
Disabled	disabled	yes	disabled	<p>If checked (xml-attr: disabled="true") the widget will NOT be shown. For buttons this could be</p>

				archived on runtime via setting them readonly.
onChange	onchange	?	onchange	If checked (xml-attr: onchange="1") and the content of the widget is changed by the user, the form will be submitted. Via xml or by a program other values can be set (not in the editor at the moment and this is not compatible with the phpGTK implementation of the eTemplates).
Span, Class	span class	span: no class: yes	span	In the editor and internally this field contains two comma-separated values: span: how many cells a widget should span (default is one), the special value of 'all' can be used to indicate it should span all remaining cells of a row. This is not supported by xul-grid's at the moment, but is planned to be. class: the CSS class for the widget. If the class-string starts with a '@' it is replaced by the value of the content-array at this index (with the '@'-removed and after expanding the variables).
Align	align	yes	align	Can be set to 'left' (default), 'center' or 'right'.
Width, Disabled column-attr.	width	yes	row[0][#]	Can be set to a percentage (eg. '10%'), a number of pixels or ...
Height, Disabled row-attr.	height	yes	row[0][h#]	Can be set to a percentage (eg. '10%'), a number of pixels or ...
Disabled column-attr. row-attr.	disabled	no	disabled	Syntax: [!]{@name value}[={@name2 value2}] Disables (=dont show it) a row/column if a certain criteria is (not (=!)) met. If no '=...' / 2. value is given, the test is made on the first value being not empty, else the test is made by comparing the two values. Instead of a value you can give a name as an index into the content prefixed by @. Examples: !@data disables row/col if value of data is empty, @val=false disables if value of val is equal to (the string) 'false'
Class, Valign row-attr.	class valign	both: yes	row[0][c#]	In the editor and internally this field contains two comma-separated values: class: the CSS class for the row, there are 4 predefined css-classes: nmh: next-match-header-background, nmr: alternating next-match-row-background which gets replaced by the eTemplate

				<p>class with nmr0 or nmr1.</p> <p>valign: vertical alignment of the widgets in the row: 'top', 'middle' (default) or 'bottom'</p>
blurText	blur	no	blur	<p>This text get displayed if an input-field is empty <u>and</u> does not have the input-focus (blur). It can be used to show a default value or a kind of help-text.</p> <p>If it contains a text (eg. 'Search...'), this text is run through lang(), if it contains a reference to the content array (eg. '@blur_text') it does <u>not</u> get translated.</p>

standard widgets and extensions of the eTemplates

Widget Name in Editor	xml tag	xul	internal name	description of the widget
Label	<description />	yes	label	<p>a textual label The content is taken from the content-array but it can have an own label from the label attribute too.</p> <p>Options has 5 comma-separated fields:</p> <ol style="list-style-type: none"> 1. if it contains a 'i' and/or a 'b' the content (not the label) is rendered in italic and/or bold. 2. link: if set to a menuaction string or an array with get-params (via the content-arry), a link to that methode = app.class.method is put around the label 3. if set URLs in the content get activated 4. name of form-element the label is for: gives focus to that element if the label gets clicked 5. target for the link, eg. _blank 6. widthxheight if a popup should be used for the link, eg. 600x400 7. title for the link
Text	<textbox />	yes	text	<p>a single-line input field for text In the html-UI this is rendered as <input ...></p> <p>Options has 3 comma-separated fields:</p> <p>xml: size: the length in chars of the input-field</p> <p>xml: maxlength: the maximum length of the input</p> <p>xml: validator: perl regular expression to validate the input (kommas are allowed in the expression)</p>

Integer	<code><textbox type="int" /></code>	?	int	<p>a input-field to enter an integer In the html-UI this is rendered as <code><input ...></code>. <i>The input-validation is done at the moment only on server-side, clientside validation and input-restriction to only numbers is planed.</i></p> <p>Options has 3 comma-separated fields: xml: min: minimum value, default none, empty values are Ok, as long as needed is not set xml: max: maximum value, default none, empty values are Ok, as long as needed is not set xml: size: the length in chars of the input-field, default 5</p>
Float	<code><textbox type="float"/></code>	?	float	<p>a input-field to enter a float In the html-UI this is rendered as <code><input ...></code>. <i>The input-validation is done at the moment only on server-side, clientside validation and input-restriction to only numbers is planed.</i></p> <p>Options has 4 comma-separated fields: xml: min: minimum value, default none, empty values are Ok, as long as needed is not set xml: max: maximum value, default none, empty values are Ok, as long as needed is not set xml: size: the length in chars of the input-field, default 5 xml: precision: precision of the float number, default maximum</p>
Textarea	<code><textbox multiline="true" /></code>	yes	textarea	<p>a multiline input-field for text In the html-UI this is rendered as <code><textarea ...></code>.</p> <p>Options has 2 comma-separated fields: xml: cols: the width of the field in chars xml: rows: the number of rows</p>
Formatted Text	<code><htmlarea /></code>	no	htmlarea	<p>a multiline input-field for formatted (HTML) text In the html-UI this is rendered as <code><textarea ...></code> and the HTMLArea javascript</p>

(HTML)				<p>editor is used.</p> <p>Options has 5 comma-separated fields: xml: mode: {ascii simple extended advanced} xml: height: height of htmlarea xml: width: width of htmlarea xml: toolbar: {true false} show toolbar xml: base_href: if passed activates the browser for images at the path (relative to the docroot)</p>
Checkbox	<checkbox />	yes	checkbox	<p>a widget that can be checked or unchecked In the html-UI this is rendered as <input type="checkbox" ...>.</p> <p>Multiple checkboxes can have an identical name ending with [], in that case the value will be an array with the set_value's of the checked boxes. You can use a button with a custom javascript onclick action of eg. "toggle_all(this.form,form::name('nm[rows][checkbox][]')); return false;" and a set_value of "\$row_cont[id]" to toggle all checkboxes in the lines of a nextmatch widget. The form::name() function translate the name used in the template into the name used in the form. <i>If the button is an image-button, check needed to render it as button and not as image with link, which has no this.form property!</i></p> <p>Options: [set_value][,unset_value[,ro_true[,ro_false]]] set_value: which value in the content represents the checked state, default=1 unset_value: value in the content representing the unchecked state, default=0 ro_true: what should be displayed for a readonly checked box, default=x ro_false: display for an unchecked box, can be set to 'disable', to not display the widget (incl. label), default is empty</p>
Radiobutton	<radio />	?	radio	<p>a widget in a group of which only one can be checked In the html-UI this is rendered as <input type="radio" ...> Unlike XUL (and like html) the radio-buttons are grouped by giving them the</p>

				<p>same name / id.</p> <p>Options: [set_value][,ro_true[,ro_false]]</p> <p>set_value: which value in the content represents the checked state, default=1</p> <p>ro_true: what should be displayed for a readonly checked box, default=x</p> <p>ro_false: display for an unchecked box, can be set to 'disable', to not display the widget (incl. label), default is empty If the value of the content array at index name/id matches set_value the radiobutton is marked 'checked'.</p>
Submitbutton	<button image="img.gif" ro_image="img-grey.gif" />	yes	button	<p>a button to submit the form / end the dialog</p> <p>In the html-UI this is rendered as <input type="submit" ...>.</p> <p>If a button is set readonly (via setting its id in the \$readonlys array passed to exec) it is not rendered at all (if no ro_image is given), like it would be disabled.</p> <p>needed: if set and the user has JavaScript enabled the button is rendered as a link around the label and a hidden input to set id if the link is clicked.</p> <p>Options xml: image, ro_image: Image to use instead of a Button with a label. There will be no button around the image. If a ro_image is given (separated by a comma in the editors options) it will be used if the button is set readonly (else the button is no rendered at all) . onclick: specify some java-script to be called if the button gets pressed/clicked:</p> <p>a) general javascript: "window.close();"</p> <p>b) confirmation: "return window.confirm();" (message get run through lang()!)</p> <p>c) popup: app.class.func&id=\$cont[id],target(default _blank),width (default 600),height (default 450) You can use \$cont[] or \$row_cont[] (note no quotes!) to pass further information to the popup via the content array.)</p>
Button	<buttononly image="img.gif" ro_image="img-grey.gif" />	no	buttononly	<p>a button</p> <p>Same as Submitbutton but it is rendered as <input type="button" ...> in the html-UI</p>
Horizonatal Rule	<hrule />	no	hrule	<p>a horizontal rule / line</p> <p>In the html-UI this is rendered as <hr ...></p>

				Options can contain a width of the rule, default is 100%
Template	<template id="app.name" content="subarr" />	yes	template	a separate eTemplate to be loaded into this cell Name xml: id : the name of the etemplate to load Options xml: content : if set, the template uses an own sub-array of the content array indexed by the value of this field (if not the full content-array is used). Variables like \$row can be used as descript for the general attribute Name.
Image	<image src="foo.gif" label="Get a foo" options="app.class.method" >	yes	image	shows an image Label xml: label : the label is shown as tooltip (like html-title) Name xml: src : the name of the image to load, the image is search in the apps template-dirs Options xml: options : up to 4 comma-separated values: 1. link to a methode = app.class.method for the image 2. target for the link, eg. _blank 3. imagemap 4. widthxheight if a popup should be used for the link, eg. 600x400
Selectbox	<menulist> <menupopup id="name" options="Select one" /> </menulist> <i>multiselect: options > 1</i> <listbox rows="#" /> <i>Examples for predefined selectboxes:</i> <listbox type="select-cat" rows="5" /> <menulist>	yes	select	shows a selectbox The content of the selectbox / the options have to be in an array which can be in 2 locations: 1. in \$content["options-\$name"] 2. or in an separate array only for select-box-options under the index name, this array is passed to the exec or show function of the etemplate-class Options in the editor : if set and > 1 the selectbox is a multiselection with options number of lines xml: rows : <i>only for <listbox></i> : number of rows to show xml options : <i>only for <menupopup/></i> : textual label for a first Row, e.g. 'All'

	<pre><menupopup type="select-account" options="All,both,2"/> </menulist></pre>			<p>or 'None' (id will be ""), additional attr see sub-types</p> <p>xml: type: can be set to get several predefined select-contents, in that case you dont need to set the content as described above (if set it too its in front of the predefined rows):</p> <p>select-cat: Select an eGW category, determined by the options-field: ,{no_global_cats},{extra_style_multiselect},{cat_app(default:current app)}</p> <p>select-account: Select a user and/or group, determined by the options-field: ,{accounts(default) groups both},{"(phpgw-default) 0(only lid) 1(only names) 2(both)}</p> <p>select-percent, select-priority, select-access, select-country, select-state: as you expect by the name</p> <p>select-year, select-month, select-day: options for year: ,start,end (start and end can be a number of years from now or if > 100 a absolut year)</p> <p>select-number: Select a number out of a range specified by the options-field: ,{start (default=1)},{end (incl., default=10)},{decrement (default={padding zeros}1)},{suffix}.</p> <p>Example with padding zeros: options=',0,59,05' will give values: 00, 05, 10, ..., 55 (like you would use it for minutes in a time-field). The suffix get's added to the label of each option.</p> <p>select-dow: Select one or multiple weekdays, keys are as defined in MCAL_M_... (1=Sun, 2=Mon, 4=Tue, ...)</p> <p>select-app: Select an application, available options: ,{"=user enabled(default) installed all}</p>
FileUpload	<file id="name"/>	no	file	<p>Input and Button to select a file for uploading Returns the file-name of the uploaded file in the servers tmp-dir (the</p>

				<p>webservice needs to have a writable tmp-dir) plus, if javascript is enabled, the local filename of the client as "\${name}_path".</p>
Date	<pre><date options="Y-m-d,1"/> <date type="date-time"/> <date type="date-timeonly" options="H:i"/> <date type="date-houronly"/> <date type="date-duration"/></pre>	no	date	<p>Date-/Time-input via selectboxes or a field for the year The order of the input-fields is determined by the prefs of the user. Options: [datetime-storage-format] [,&1=year-no-selectbox &2=today-button &4=one-min-steps &8=ro-suppress-0h0] datetime-storage-format is the format, in which the date is stored in the variable: empty means an unix-timestamp (in GMT), or a string containing the letters Y, m, d, H, i plus separators, eg. 'Y-m-d': 2002-12-31. The storage format for times is always 24h or timestamp with date 1.1.1970 (if no date used). (This has nothing to do with the format of the display, which is only determined by the users preferences.) year-no-selectbox if set (&1) an int-widget (input-field) is used instead of a select-year widget. today-button: if set (&2) a [Today] button is displayed which sets the fields to the up-to-date date (via javascript) one-min-steps: if set (&4) the minute-selectbox uses one minutes steps, default 5min steps ro-suppress-0h0: if set (&8) the time is suppressed for readonly and a time of 0h0 day-of-week-prefix: if set (&16) readonly dates get prefixed with the day of week week-number-prefix: if set (&32) readonly dates get prefixed with lang('Wk') & weeknumber Sub-widgets: date-time: a date and a time and date-timeonly or date-houronly: only a time / hour These widgets allow the input of times too or only, they use 12h am/pm or 24h format as specified in the user prefs. <i>If readonly is set, this widget can be used to display a date, without the need to convert it.</i></p> <p>Duration a floating point input with an optional selectbox for the unit (hours</p>

				<p>or days)</p> <p>Options: [duration-storage-format] [, [duration-display] [, hours_per_day]]</p> <p>duration-storage-format: 'h' = hours (float), 'd' = days (float), default minutes (integer)</p> <p>duration-display: 'd' = days, 'h' = hours, 'dh' = days or hours with selectbox, optional '%' allows to enter a percentage</p> <p>hours_per_day: conversation between hours and (working) day, default 8</p> <p>hours_per_day: conversation between hours and (working) day, default 8</p>
VBox, HBox, Box	<pre><vbox> <widget ...> <widget ...> </vbox> <hbox span="all"> <widget ...> <widget ...> </hbox> <box orient="horizontal"> <widget ...> <widget ...> </box></pre>	yes	vbox, hbox, box	<p>vertical or horizontal container for child widgets. This is useful if one needs more widgets or widgets outside the column- / row-order of a grid. HBox or VBox is rendered as Grid/html:table with only one row or colum. Box is rendered as a html:div containing all child-widgets. Disabled child-cells are completely left out (no empty cells or rows get generated).</p> <p>Options in the editor: the number of cells in the box (does NOT need to be set in xml).</p> <p>orient: horizontal, vertical or none (means h/vbox as expected and no table for boxes) cellpadding,cellspacing: known table-options keepEmpty: if true, empty cells (lines or rows) are kept, otherwise they are completely removed</p>
GroupBox	<pre><groupbox> <caption label="Legend"/> <widget ...> <widget ...> </groupbox></pre>	yes	groupbox	<p>container to visually group other widgets by putting a border around them. The upper line may contain a legend. The widgets are ordered vertical, like a VBox. Disabled child-cells are completely left out (no empty cells or rows get generated).</p> <p>Options in the editor: the number of cells in the box (does NOT need to be set in xml).</p> <p>orient: horizontal, vertical or none (defaults to vertical) options: cellpadding,cellspacing of the table</p>

Tabs	<pre><tabbox id="name"> <tabs> <tab label="Tab 1" statustext="Help"/> ... </tabs> <tabpanels> <grid id="app.name.tab1"/> ... </tabpanels> </tabbox></pre>	yes	tab	<p>shows a tab-widget The tab-widget is implemented as an extension, as html does not have a tab-widget.</p> <p>The following fields / attributes are in the Editor and internaly in the class separated by ' ', in the xml/xul-file the are attributes of each tab- or grid-tag: Label xml: label: the labels of the tabs eg. 'Tab 1 Tab 2 Tab 3' Help xml: statustext: of the tabs Name xml: id: the names/ids of the eTemplates/grid's to fill the bodies of the tabs, if the name contains no '.', it will be prefixed with the name of the template the widget is in plus a '.'</p>
Manual	<pre><manual></pre>	no	manual	<p>open the online help: displays a small manual icon.</p> <p>Name xml: id: optional name of the manual page (as index into \$content or direct). If no manual page is given, the link included the referer as _GET param.</p>
Custom fields	<pre><custom_fields></pre>	no	custom_fields	<p>display custom fields: the fields can be configured with admin.customfields.edit&appname={ app }</p> <p>The indexes of the custom fields in content are prefixed with a hash (#).</p>
NextMatch	<pre><nextmatch options="notes.index.rows" id="nm"/></pre>	yes	tab	<p>shows a table with some selectboxes, a search-field and arrows to scroll the table The nextmatch-widget is implemented as an extension.</p> <p>Options xml: options: name of the template to display the rows Name xml: id: index into the content-array, it need to be pre-set with some information for the nextmatch widget and it returns its content with it:</p>
Nextmatch-SortHeader	<pre><nextmatch type="nextmatch- sortheader" id="col-name" options="DESC"></pre>	no	nextmatch-sortheader	<p>These widget are an optional part of the nextmatch widget.</p> <p>nextmatch-sortheader</p>

<p>Nextmatch-FilterHeader</p> <p>Nextmatch-Custom FilterHeader</p> <p>Nextmatch-AccountFilter</p>	<pre>label="ColLabel"/> <nextmatch type="nextmatch-filterheader" id="col-name"/> <nextmatch type="nextmatch-customfilter" id="col-name" options="select-present"/> <nextmatch type="nextmatch-accountfilter" id="col-name"/></pre>	<p>nextmatch-filterheader</p> <p>nextmatch-customfilter</p> <p>nextmatch-accountfilter</p> <p>nextmatch-header</p>	<p>Widget to be placed as a column-header in the headerline of a nextmatch-template. It allows, by clicking on it, to order the lines of the nextmatch after a certain column. The column-name is given as name (xml:id) the label is show as a link of button (no javascript). One can specify a default sorting: options={DESC ASC} (default=ASC), to be used when the header is clicked for the first time. Consecutive click on the header change the sorting direction, indicated by a little up- or down-arrow. As a second comma-separated parameter one can specify an extra label for the column-selection.</p> <p>nextmatch-filterheader Widget to allow to show only certain row, which match a selected filter-value. The column-name is given as name (xml:id), the options of the displayed selectbox need to be set as for an ordinary selectbox (eg. in the options parameter to the uietemplate::exec function). If no extra-label is given in options, lang('all') will be used for the empty value, which means no filter activ. An (optional) label can be given and is also used for the column-selection.</p> <p>nextmatch-customfilter The custom filterheader allows to use other (select-)widgets to filter by them. They have to be specified as the first parameter in the comma-separated options attribute. In all other aspects it is identical to the filterheader.</p> <p>nextmatch-accountfilter The Accountfilter allows to select users (via the preferred user-selection-method) to filter by them. It's identical to a nextmatch-customfilter with options="select-account".</p> <p>nextmatch-header Just a header-label for a nextmatch column. It names the column for the column for the column-selection (in difference to the label). The name is used to hide the column (with a 'no_' prefix) and as the name for the preference. A</p>
--	---	--	---

				<p>different label for the column-selection preference can be specified via the option field.</p> <p><i>Note:</i> All four widgets interoperate with the nextmatch-widget which passes the set values as part of the query-parameter to its get_rows function, they are <u>not</u> returned in the rows sub-array.</p>
LinkWidgets	<pre><link type="link-to" id="name"/> <link type="link-list" id="name"/> <link type="link-string" id="name"/></pre>	no	<pre>link-to link-list link-string</pre>	<p>These widgets are the UI-part of the link-class (<code>{bo so}link</code>) in the API.</p> <p>eGroupWare has a linking system that lets you link two records from different apps together.</p> <p>For example, you can link the addressbook entry of the person you're meeting with to the meeting on your calendar, or an infolog entry for the phone call you made to postpone the meeting.</p> <p>To display links in your own application, you should use the LinkList group of widgets.</p> <p>link-list Widget to shows the links to an entry and a Unlink Button for each link.</p> <p>It needs an array with two entries. If you name the LinkList widget 'links', you need:</p> <pre>\$data['links']['to_app'] = 'myapp'; \$data['links']['to_id'] = \$record_id;</pre> <p>This will display links where the \$record_id record of myapp is one side of the link. Make sure that both variables are properly defined before the form template gets executed.</p> <p>link-to</p>

Widget to make a links to other entries of link-aware apps and to attach files.

Note: Both Widgets can be used on the same template with the same name. They share the content of the variable with that name, which contains just the id of the entry in the current app.

link-string

Comma-separated list of link-titles with a link to its view-method, value is like the return of `bolink::get_links()`.

Before you can use a link to your application, you need to specify some information in a 'search_link' hook.

In your `setup.inc.php`, you need to point `$setup_info['myapp']['hooks']['search_link']` to a function that will return an array:

```
return array(
'query' =>      'myapp.bo_myapp.link_query',
                // A function that takes a search string
                // and returns a list of matching records
'title' =>      'myapp.bo_myapp.link_title',
                // A function that takes an id from one side
                // of a link and returns a string for that entry
'view'  =>
array('menuaction'=>'myapp.ui_myapp.link_view'),
        // Function to view a link, may be an existing view
function
'view_id'      =>      'link_id',
                // name of the id variable provided to the view
function above
'add'   =>      array('menuaction' =>
'myapp.ui_myapp.new_entry'),
        // Function to add a new entry
```

			<pre>);</pre> <p>Also, make sure that the declared methods are implemented and methods from the UI class are listed in its \$public_methods attribute:</p> <pre>class ui_myapp { var \$public_methods = array('view' => true, 'add' => true); ... }</pre>
Ajax Select		ajax_select	<p>The Ajax Select is a Combo Box. It lets the user type anything they want, and choose from a list of options that are presented below. The user is not limited to the choices, but there is some checking done. If what they type returns several results, and they don't choose one, for example. You can reject any values you don't like in your UI code. It is best used where you might normally want to use a selectbox but your list of data is too large. You can have several on one page, but the name must be different for each.</p> <p>Options can be found under the "AJAX Select options" section of the pop-up.</p> <p>Data Source: the list options, can be any function that can provide data for a nextmatch widget.</p> <p>Title Source: When an option from the list is selected, the text in the search function is replaced with the result of this function. The ID Field is passed. link_title() functions work well.</p> <p>ID Field: Data Source is expected to return an array as for a nextmatch, with several columns. This is the key of the column you actually want returned for</p>

a value.

Result row template: (Optional) You can provide a custom eTemplate to use for the list options. It should be constructed similarly to a row template for a nextmatch, and will be repeated for each option.

Link: (Optional) If the field is read-only, and Link is provided, the widget will turn into a link. Link should look like:
perp_ap.ui_perp_supplier.edit&supplier_id=\${cont[supplier_id]} where ID Field is supplier_id.

Icon: (Optional) An icon placed to the left of the search box, to help indicate what the user is searching (addresses, suppliers, etc.). It will be automatically resized.