

Universal Lookup Field Component

LWC version Guide v 3.34



Content

Installation	4
Component Attributes	5
Required attributes	6
object-type	6
selected-record-id	6
Attributes to customize search	8
query-condition	8
If you are using LULC in the Flow (use INPUT params):	8
If you are using LULC in the Custom Lightning component:	9
primary-search-field	10
additional-field	11
fuzzy-search	12
order-by	13
with-sharing	14
query-limit	15
with-advanced-search	15
field-set-name	16
Attributes to customize component UI/UX	18
label / plural-label	18
object-icon-path	19
is-required	19
error-message	20
show-five-recent	20
read-only	21
drop-down-query	22
drop-down-limit	23
help-text	24
place-holder	24
style-class	25
width	26
errorMessage	26
is-dropdown-overflow-available	27
Component Events	28
onblur	28
onfocus	29
onchange	29

Component Global Methods	30
fireChanging	30
Component Exception Handler	31
Lookup Advanced Search	32
Supported Reactive Flow	34



Installation

Notice that for correct Custom Lightning component's work you should deploy it for all Users custom Domain.

 **Install Universal Lookup Field Component**
By Synebo


☐ Install for Admins Only


☒ Install for All Users


☐ Install for Specific Profiles...

Install

Cancel

App Name	Publisher	Version Name	Version Number
Universal Lookup Field Component	Synebo		

[Additional Details](#) [View Components](#)

 **Install Universal Lookup Field Component**
By Synebo

 Installation Complete!

Done

App Name	Publisher	Version Name	Version Number
Universal Lookup Field Component	Synebo		

Component Attributes

Attribute	Type	Default
additional-field	String	-
drop-down-query	String	-
drop-down-limit	Integer	5
with-advanced-search	Boolean	false
field-set-name	String	-
fuzzy-search	Boolean	false
is-required	Boolean	false
label	String	standard
limit	Integer	200
object-type*	String	-
order-by	String	-
plural-label	String	standard
object-icon-path	String	-
primary-search-field	String	standard
query-condition	String	-
read-only	Boolean	false
selected-record-id	Id	-
show-five-recent	Boolean	true
style-class	String	-
width	String	100%
with-sharing	Boolean	false

Required attributes

Attributes that are required for correct component work

object-type

Type

String

Description

Required attribute that defines a type of Object that will be used in Lookup.
It can be Standard or Custom Salesforce Object.
In this attribute a full API Name of Salesforce Object has to be defined.

E.g.:

```
<l_lookup-lookup-cmp object-type="Contact"
    selected-record-id={contactId}></l_lookup-lookup-cmp>
```

```
<l_lookup-lookup-cmp object-type="Customer__c"
    selected-record-id={customerId}></l_lookup-lookup-cmp>
```

selected-record-id

Type

String

Description

String attribute to select record by default by Id.

E.g.:

Component html:

```
<l_lookup-lookup-cmp class="accountLookup" object-type="Account"
    selected-record-id={accountId}></l_lookup-lookup-cmp>

<lightning-button label="Debug log"
    onclick={debugSelectedId}></lightning-button>
```

Component JS:

```
debugSelectedId(event) {
    let lookup = this.template.querySelector('.accountLookup');

    if (lookup) {
        console.log("recordId = ", lookup.selectedRecordId);
    }
}
```

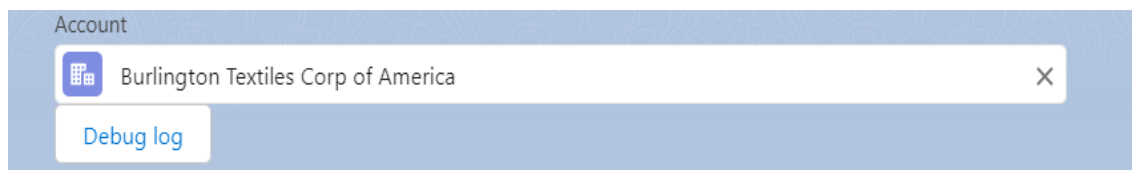
Component html:

```
<l_lookup-lookup-cmp class="accountLookup"
    selected-record-id={accountId}
    object-type="Account"
    onchange={debugSelectedId}>
</l_lookup-lookup-cmp>
```

Component JS:

```
debugSelectedId(event) {
    console.log('Lookup component Id: ' + event.detail.cmpId);
    console.log('Old Id: ' + event.detail.oldValue);
    console.log('New Id: ' + event.detail.newValue);
}
```

UI:



The screenshot shows a user interface with a header 'Account'. Below the header is a lookup field containing the text 'Burlington Textiles Corp of America' and a close button (X). Below the lookup field is a button labeled 'Debug log'.

Console:

Lookup component Id: accountLookup	aura_prod.js:102
Old Id: undefined	aura_prod.js:102
New Id: 0010700000ij1TKAAY	aura_prod.js:102

NOTE:

In console displayed record Id according ObjectType="Account" starts with Object Key Prefixes "001"

Attributes to customize search

query-condition

Type

String

Description

Optional string attribute that allows adding additional conditions to query.
This attribute gives an opportunity to add a filter to lookup.
Be careful with this attribute - if the query is invalid, the component will not work.
You can use hardcoded values or values from attributes for conditions.

If the query-condition is changed, the selected value will be removed. Use attribute *remove-value-on-query-condition-change* to change this behavior.

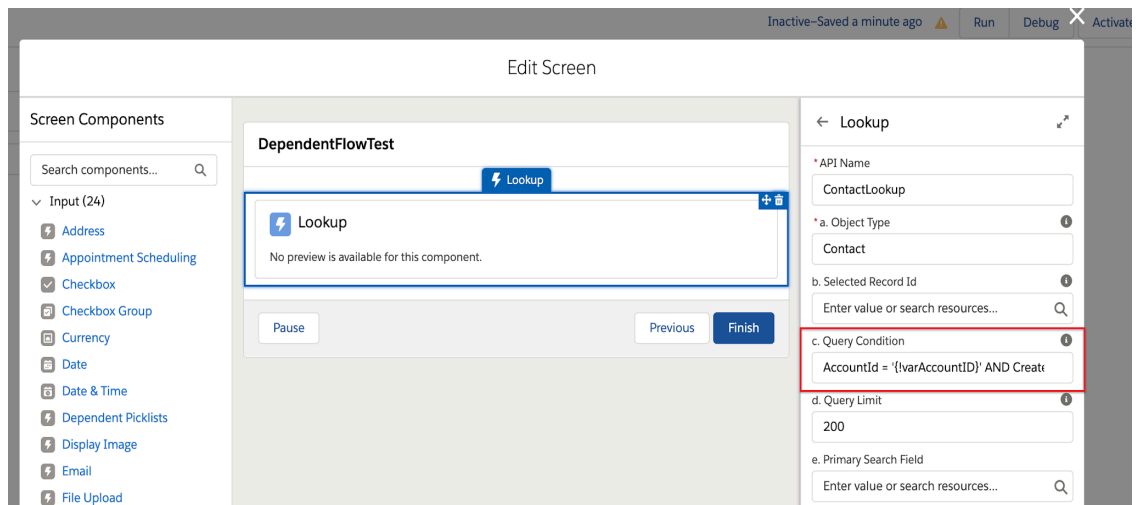
Salesforce documentation: [WHERE conditionExpression](#).

E.g.:

If you are using LULC in the Flow (use INPUT params):

```
AccountId = '{!varAccountID}' AND CreatedDate=Today
```

AccountId - API Name of the field that you want to use in condition
{!varAccountID} - variable from a Flow



If you are using LULC in the Custom Lightning component:

```
// Users with Profile "System Administrator" OR "Account Manager"
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="User"
    query-condition="Profile.Name = 'System Administrator' OR
        Profile.Name = 'Account Manager'">
</l_lookup-lookup-cmp>
```

```
// Contacts with Last Name starts from "S"
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="User"
    query-condition="LastName LIKE 'S%'"></l_lookup-lookup-cmp>
```

```
// Accounts that are created this week
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="Account"
    query-condition="CreatedDate=THIS_WEEK"></l_lookup-lookup-cmp>
```

```
// Example with value from attribute in condition:
// HTML
<l_lookup-lookup-cmp class="uniqueLookupClass"
                    object-type="Contact"
                    query-condition={obtainCondition}></l_lookup-lookup-cmp>

// JS
accountId = "0010a00001LtTmiAAF";

get obtainCondition() {
    return `AccountId='${this.accountId}'`;
}
```

primary-search-field

Type

String

Description

An API Name of the object field that will be primarily used in searching records (instead of standard fields).

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
                    object-type="Account"
                    primary-search-field="Site"
                    additional-field="Name">
</l_lookup-lookup-cmp>
```

Demo:

Account



additional-field

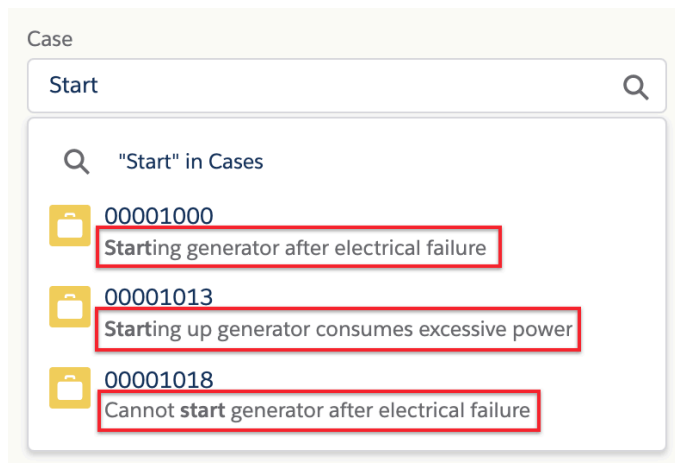
Type

String

Description

Field that is displayed below name or number and used as an additional field for search.
An attribute has to contain the correct API Name of a field that is related to the object that is defined in “**object-type**” attribute.
Works for standard and custom objects.

E.g.:



If user does not define special value, it will contain standard default value in next cases:

Object	additionalField	Display
Case	Subject	Subject
Contact	AccountId	Account.Name
Contract	AccountId	Account.Name
Order	AccountId	Account.Name
Lead	Title	Title
Product2	ProductCode	ProductCode

Field types that are allowed to use in the attribute:

string, email, fax, phone, picklist, reference (will be displayed the Name of record), *url*

fuzzy-search

Type

Boolean

Description

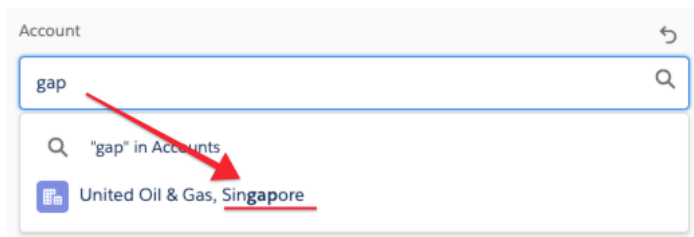
The optional attribute, that enables/disables a search within primary and additional object fields' values. Allow searching in the middle of a word or number starting with two symbols.

The default value is false.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  fuzzy-search="true">
</l_lookup-lookup-cmp>
```

Demo:

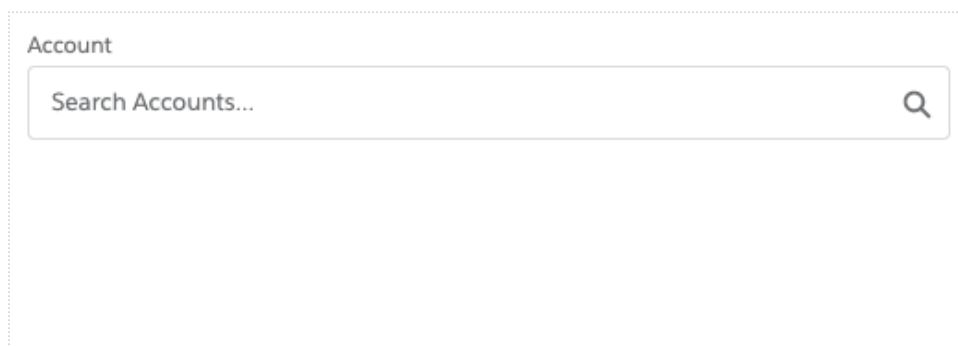


Account

gap

Q "gap" in Accounts

United Oil & Gas, Singapore



Account

Search Accounts...

order-by

Type

String

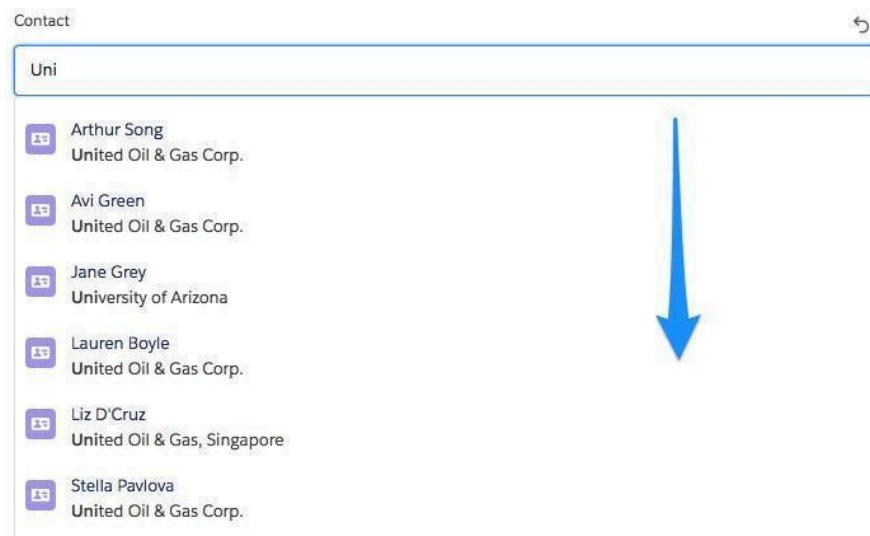
Description

Optional attribute that allows sorting records.

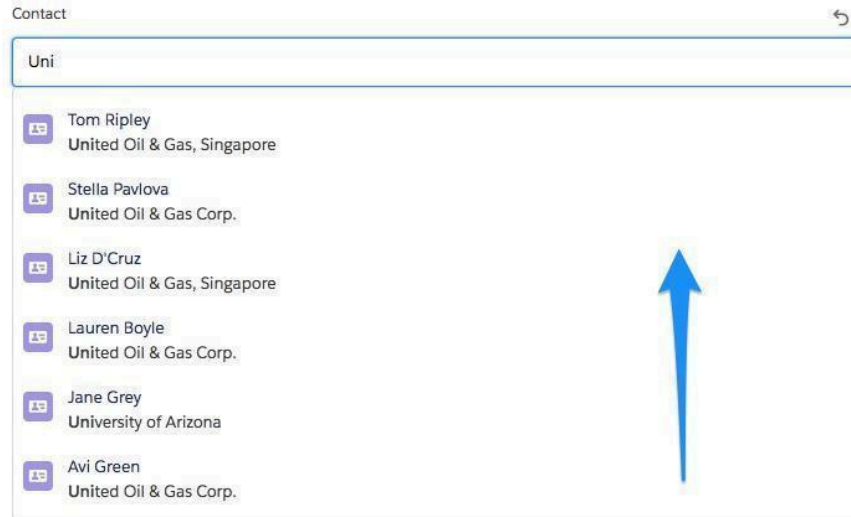
Salesforce documentation: [ORDER BY in SOQL](#)

E.g.:

```
// Order by Contact Name A-Z
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="Contact"
    order-by="Name">
</l_lookup-lookup-cmp>
```



```
// Order by Contact Name Z-A
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="Contact"
    order-by="Name DESC">
</l_lookup-lookup-cmp>
```



Also you can define several fields by comma separator:

```
// Order by Name and Created date
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  order-by="Name DESC, CreatedDate">
</l_lookup-lookup-cmp>
```

with-sharing

Type

Boolean

Description

Optional attribute that controls retrieving records according to user's access level.

The default value is true.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  with-sharing="true">
</l_lookup-lookup-cmp>
```

query-limit

Type

Integer

Description

Optional attribute that controls retrieving records according to user's access level.

The default value is 200.

with-advanced-search

Type

Boolean

Description

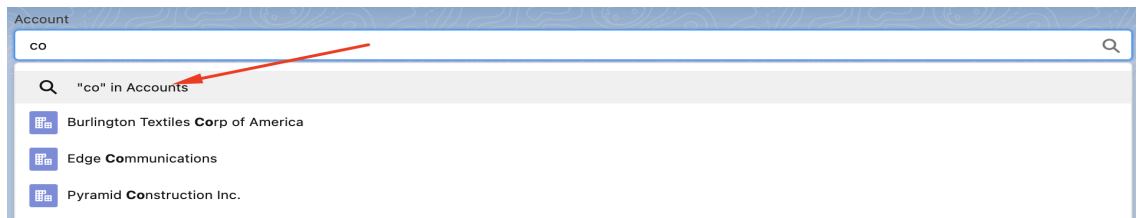
Optional attribute that allows to open advanced search modal by clicking on the row with search icon.

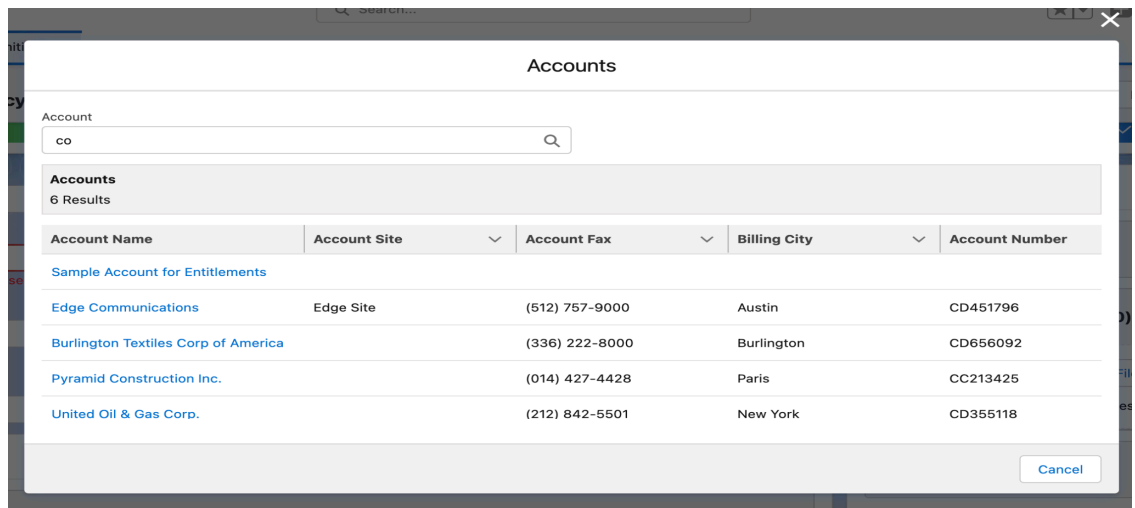
The default value is **false**.

E.g.:

```
<l_lookup-lookup-cmp class="myLookup"
  object-type="Contact"
  field-set-name={fieldSetName}
  with-advanced-search={withAdvancedSearch} >
</l_lookup-lookup-cmp>
```

Controller: @api withAdvancedSearch = false;





field-set-name

Type

String

Description

Attribute that defines a Field Set API Name which is used for Advanced Search. The fieldSetName is required when **withAdvancedSearch** is **true**. The fields from Field Set will be used for search and displayed in the Advanced Search table. If primarySearchField is not in the field set, it will be added and displayed in the advanced search table. If an additionalField is populated and not in the field set, it will be added and displayed in the advanced search table.

The fields maintained in Fields Set for Advanced Search:

- Like / Equals:
Auto Number, Lookup Relationship, Email, Phone, Picklist, Text, URL.
- Only Equals:
Number.
- Only View:
Currency, Roll-Up Summary, Checkbox, Date, Date/Time, Geolocation, Percent, Picklist (Multi-Select), Text (Encrypted), Time.

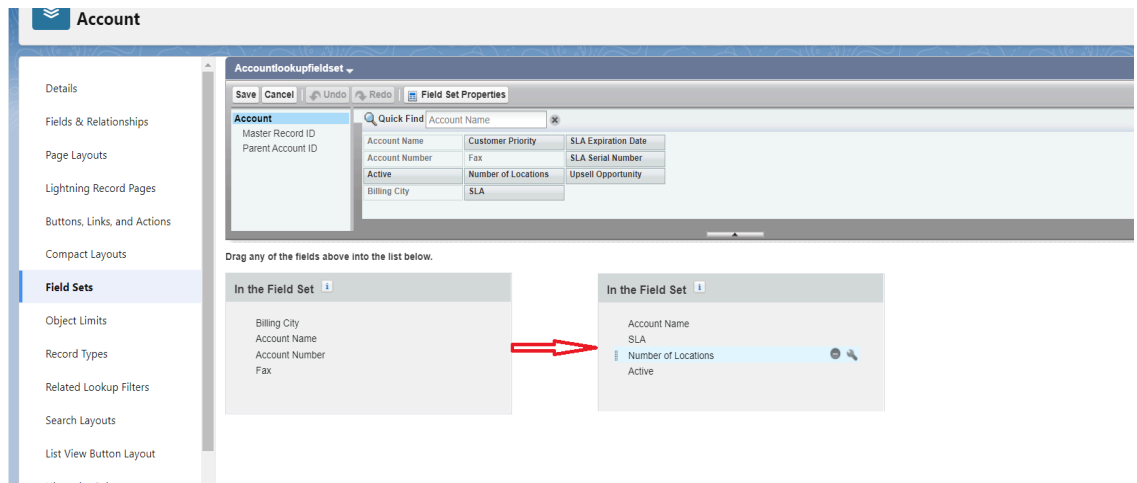
E.g.:

Component:

```
<l_lookup-lookup-cmp class="myLookup"
  object-type="Account"
  field-set-name={fieldSetName}
  with-advanced-search={withAdvancedSearch} >
</l_lookup-lookup-cmp>
```

Controller: @api fieldSetName = "l_lookup__accountlookupfieldset";

Customer has possibility to change the Field Set suite:



Attributes to customize component UI/UX

label / plural-label

Type

String

Description

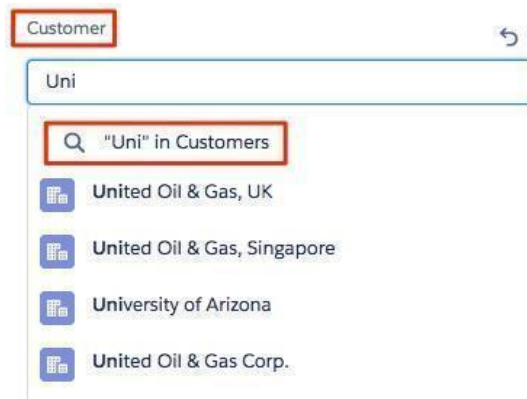
String attribute for Custom Label.

It could be useful if you use component for custom field and want to display not standard label.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Account"
  label="Customer"
  plural-label="Customers">
</l_lookup-lookup-cmp>
```

Demo:



object-icon-path

Type

String

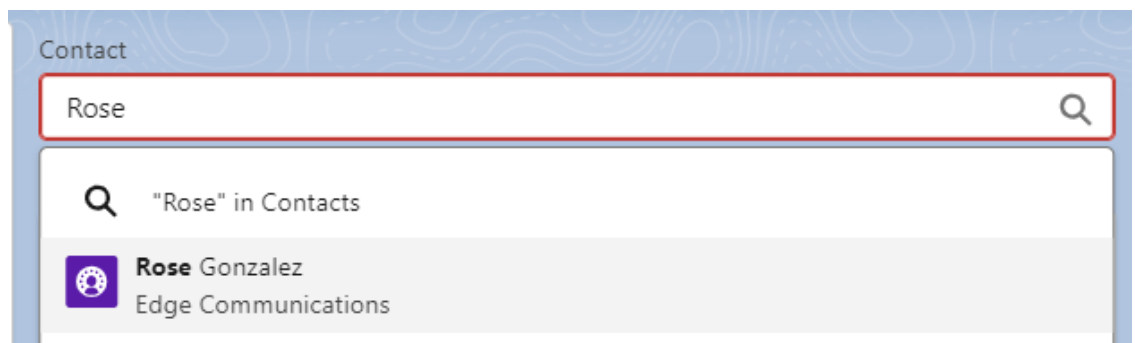
Description

Optional attribute that will be used in searching records to provide alternative visual representations for users.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  object-icon-path="standard:data_cloud"
  additional-field="Name">
</l_lookup-lookup-cmp>
```

Demo:



is-required

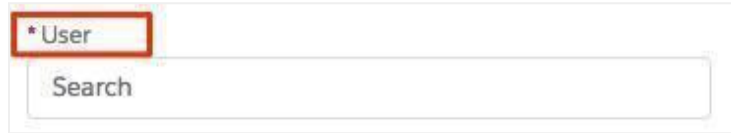
Type

Boolean

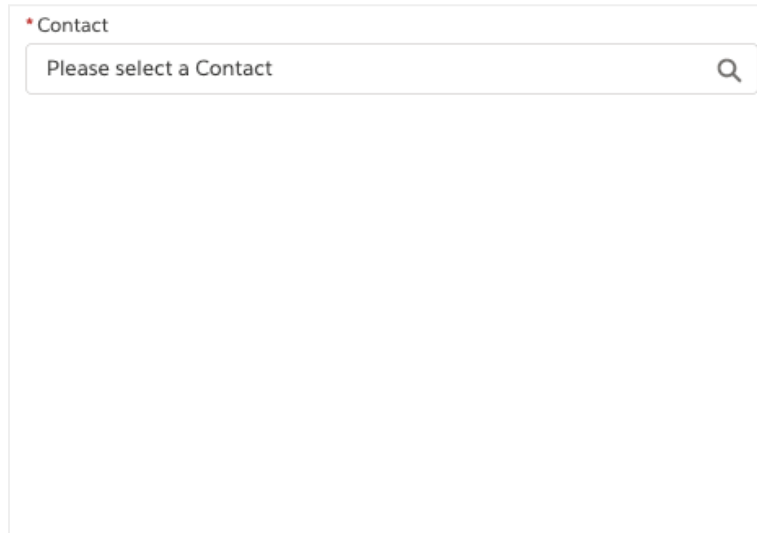
Description

The optional boolean attribute, which is false by default.

If it is set to true, a red asterisk will be displayed next to the Universal Lookup label.



A screenshot of a user lookup field. The label is "* User", where the asterisk is red and the text is in a standard font. Below the label is a search bar with the placeholder text "Search". The entire field is enclosed in a light gray border.



A screenshot of a contact lookup field. The label is "* Contact", where the asterisk is red and the text is in a standard font. Below the label is a search bar with the placeholder text "Please select a Contact" and a magnifying glass icon on the right. The entire field is enclosed in a light gray border.

error-message

Type

String

Description

The optional string attribute, used for validation errors. Default value: 'Complete this field'.

show-five-recent

Type

Boolean

Description

Optional boolean attribute that switches on or switches off Lightning salesforce feature. Standard lookups in Lightning by default show recent viewed records on focus. Maximum

that will be displayed is 5 records.
It is true by default.



read-only

Type

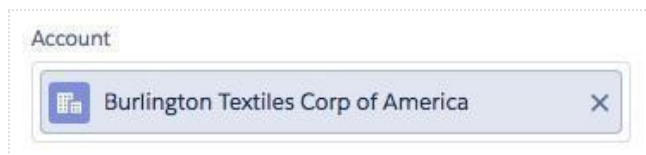
Boolean

Description

Boolean attribute to define if component is read-only.
If you define the lookup component as read-only, you will not allow the user to change it.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Account"
  read-only="true">
</l_lookup-lookup-cmp>
```



drop-down-query

Type

String

Description

Optional attribute that can contain query condition that will populate a drop-down with a list of suggestions instead of standard last viewed records.

It will allow for example to show a top list of records that are available for selection and users will not have to type anything.

If the attribute is empty, a standard process will work that will show 5 recently viewed records if *show-five-recent* is not false.

If you also use the `queryCondition` attribute, it will be added to drop-down-query criteria.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  drop-down-query="IsActive__c = true">
</l_lookup-lookup-cmp>
```

Demo:

Contact

Search Contacts...



drop-down-limit

Type

Integer

Description

The optional attribute, which can be used in combination with the drop-down-query attribute. It defines a number of records that will be displayed in drop-down by drop-down-query attribute.

The value is 5 by default.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  drop-down-query="IsActive__c = true"
  drop-down-limit="3">
</l_lookup-lookup-cmp>
```

Demo:

Contact

Search Contacts...

help-text

Type

String

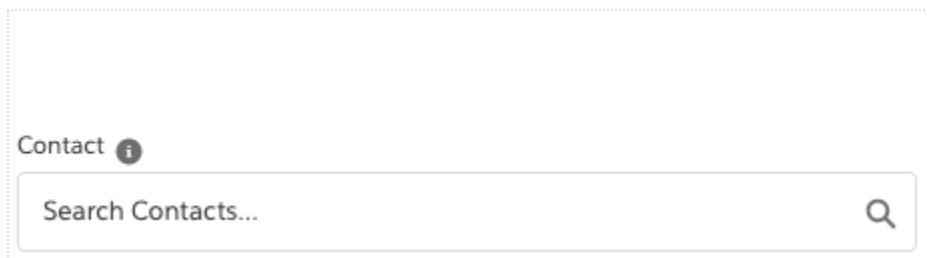
Description

Optional string attribute that adds help icon with defined text..

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  help-text="This contact will be used as primary">
</l_lookup-lookup-cmp>
```

Demo:

A screenshot of a web form element. At the top left, the word "Contact" is displayed next to a small circular help icon (an 'i' inside a circle). Below this is a search input field with the placeholder text "Search Contacts...". To the right of the input field is a magnifying glass icon.A second screenshot of the same web form element, showing the "Contact" label, help icon, search input field with "Search Contacts..." placeholder, and the magnifying glass icon.

place-holder

Type

String

Description

Optional string attribute that allows to replace standard input placeholder.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="Contact"
    place-holder="Please select a Contact">
</l_lookup-lookup-cmp>
```

Demo:

Contact

style-class

Type

String

Description

Optional string attribute that adds your own style classes to Lookup.

As LWC uses Shadow DOM, the styles provided in LWC CSS and added to the **style-class** will not affect the lookup. To resolve it, can be used one of the next variants:

1. Save styles in CSS file and add to static resource. Then in the component with lookup or in the component that uses components with lookup import this static resource and use **loadStyle** to load this CSS.
2. Use aura component as a wrapper for the main LWC component, that uses lookup or components with lookup. Add styles classes to aura CSS.
3. On the first renderedCallback add a style node with style classes to the component with lookup.

All these variants will add style classes to the DOM so it will affect the LWC component with Shadow DOM. The first variant is recommended if the aura wrapper component can't be used for any reason.

E.g. for the third variant:

HTML:

```
<l_lookup-lookup-cmp class="accountLookup"
                      selected-record-id={accountId}
                      object-type="Account"
                      style-class="red-border"
>
</l_lookup-lookup-cmp>
```

CSS:

```
.red-border {
    border : 2px solid #c23934 !important;
}
```

UI:



width

Type

String

Description

Optional string attribute that controls the width of the component. By default it's equal to "100%".

You can define it in percentage or pixels.

errorMessage

Type

String

Description

The optional string attribute, used for validation errors. Default value: 'Complete this field'.

E.g.:

```
<aura:attribute name="contactId" type="String" default="" />

<l_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    selectedRecordId="{!v.contactId}"
    isRequired="true"
    errorMessage="Contact must be selected" />
```

is-dropdown-overflow-available

Type

Boolean

Description

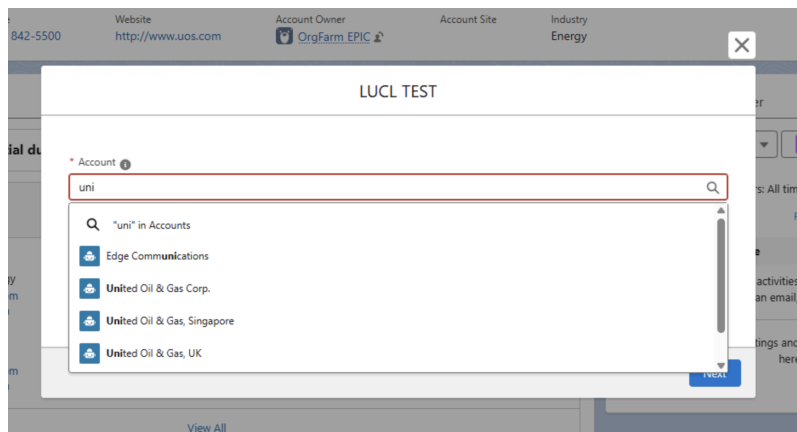
The optional attribute that adds styles to the dropdown so that it is not cut off by the modal footer.

The default value is false.

E.g.:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
    object-type="Contact"
    is-dropdown-overflow-available="true">
</l_lookup-lookup-cmp>
```

Demo:



Component Events

onblur

Description

The event is fired when the user moves off from the component.

Code to handle the event:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  onblur={handleBlur}>
</l_lookup-lookup-cmp>
```

In action you have to define your code that will be executed at onblur event.

If you have several lookups, you have to define a **class** in lookup components and you will get this as an event parameter called **cmpId**.

E.g.:

```
handleBlur(event) {
  console.log('Lookup component Id: ' + event.detail.cmpId);
}
```

```
}
```

onfocus

Description

The event is fired when the user focuses on the component.

Code to handle the event:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  onfocus={handleFocus}>
</l_lookup-lookup-cmp>
```

In action you have to define your code that will be executed at onfocus event.

If you have several lookups, you have to define a **class** in lookup components and you will get this as an event parameter called **cmpId**.

E.g.:

```
handleFocus(event) {
  console.log('Lookup component Id: ' + event.detail.cmpId);
}
```

onchange

Description

The event is fired when the user changes the content of the input.

Code to handle the event:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  onchange={handleChange}>
</l_lookup-lookup-cmp>
```

In action you have to define your code that will be executed at onchange event.
If you have several lookups, you have to define a class in each of the lookup components and you will get this as an event parameter called **cmpId**.

Also you could obtain previous value from **oldValue** and current value from **newValue** event parameters.

E.g.:

```
handleChange(event) {  
  console.log('Lookup component Id: ' + event.detail.cmpId);  
  console.log('Old Id: ' + event.detail.oldValue);  
  console.log('New Id: ' + event.detail.newValue);  
}
```

Component Global Methods

fireChanging

- * To clear the selection you have to pass *null* as a parameter.

E.g.:

Component html:

```
<l_lookup-lookup-cmp class="lookup"  
  object-type="Account"  
  selected-record-id={selectedId}  
  onchange={updateSelected}>  
</l_lookup-lookup-cmp>  
<lightning-button label="Change"  
  onclick={changeIt} >  
</lightning-button>
```

Component JS to change record Id:

```
selectedId = '0011n00002DezzVAAR';  
newValue = '0011n00002DezzoAAB';
```

```

changeIt(event)
  let lookup = this.template.querySelector('.lookup');
  if (lookup) {
    lookup.fireChanging(this.newValue);
  }
}

updateSelected(event) {
  this.selectedId = event.detail.newValue;
}

```

Notice that:

- value will not be set if **readOnly** is true
- value will not be set if new value is invalid
- value will not be set if you use **queryCondition** attribute and new value does not match criteria

Component Exception Handler

For an easier development process we have added an Exception handler.

If you specified an incorrect value in the attribute or missed the required one, you could go to Browser developer tools and read the error description.

E.g.:

Missed required **object-type** attribute:

```

<l_lookup-lookup-cmp class="lookup"
  selected-record-id={selectedId}
  onchange={updateSelected}
  onlookuperror={onLookupError}>
</l_lookup-lookup-cmp>

```

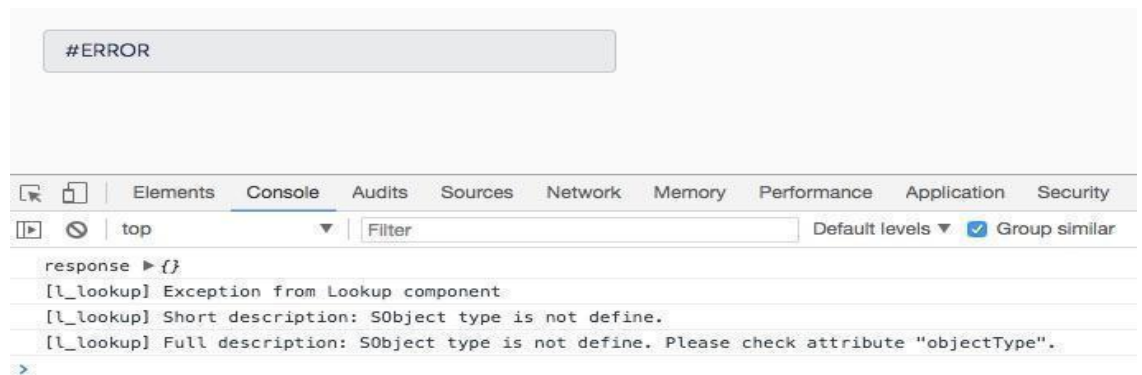
onLookupError:

```

onLookupError(event) {

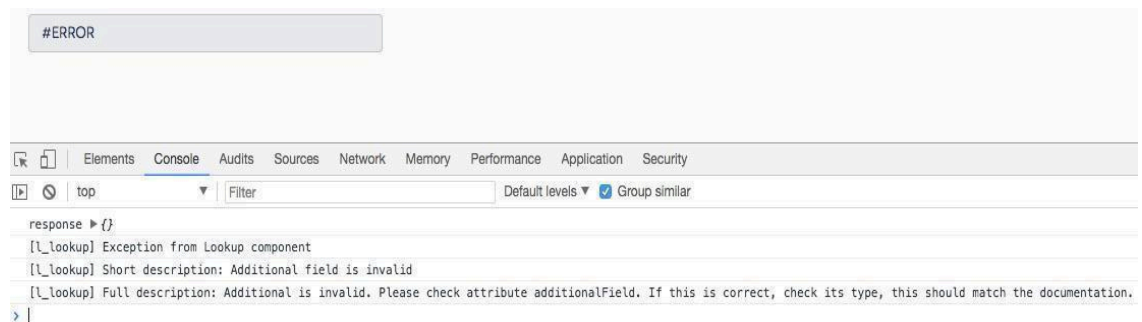
```

```
console.error(event.detail.message);
}
```



Invalid value in additionalField attribute:

```
<l_lookup-lookup-cmp class="uniqueLookupClass"
  object-type="Contact"
  additional-field="Email">
</l_lookup-lookup-cmp>
```



Lookup Advanced Search

Introduction

Lookup Advanced Search allows searching by the list of fields with the same additional condition like were provided in the lookup.

It appears by clicking on the row with the search icon in the search results drop down on the lookup component.

The Advanced Search result will be displayed on the modal window in a table. The table has a lazy loading functionality to load by 50 records.

The Lookup Advanced Search displays all fields specified in the Fields Set. Specified fields columns are sortable.

If the user clicks on the record - the result will be displayed in the input field of the page where the lookup component is located.

The Advanced Search functionality has the possibility to do searching using the input field above the table. It allows searching starting with two symbols.

For Advanced Search were added two new parameters: fieldSetName and withAdvancedSearch. When withAdvancedSearch is 'true' the fieldSetName is required.

The Advanced Search result is displayed on the modal window in a table:

Accounts

Account



Accounts

4 Results

Account Name	Billing City	Account Number	Account Fax
Edge Communications	Austin	CD451796	(512) 757-9000
Burlington Textiles Corp of America	Burlington	CD656092	(336) 222-8000
Pyramid Construction Inc.	Paris	CC213425	(014) 427-4428
Express Logistics and Transport	Portland	CC947211	(503) 421-7801

Cancel

Clicking on the record - the result will be displayed in the input field of the page where the lookup component is located :

Accounts

Account

Accounts

4 Results

Account Name	Billing City	Account Number	Account Fax
Edge Communications	Austin	CD451796	(512) 757-9000
Burlington Textiles Corp of America	Burlington	CD656092	(336) 222-8000
Pyramid Construction Inc.	Paris	CC213425	(014) 427-4428
Express Logistics and Transport	Portland	CC947211	(503) 421-7801

Cancel

Result displayed in the input field:

 Opportunity
Burlington Textiles Weaving Plant Generator

Account Name	Close Date	Amount	Opportunity Owner
Burlington Textiles Corp of America	3/11/2018	USD 235,000.00	 Maria Kropinova

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Account

Supported Reactive Flow

Reactivity empowers supported flow screen components and formulas to respond to changes on the same screen instantly. This feature enables the creation of screens that mimic single-page applications, streamlining user tasks and minimizing navigation between screens.

Available for api version not less than 59.0

Demo result:

The screenshot shows the 'Edit Screen' for a component named 'ReactiveFlowTest'. On the left, a 'Components' sidebar lists various input types like Address, Call Script, Checkbox, etc. The main area displays a 'Lookup LWC' component with a 'Record ID' field containing the formula '{!Custom_Lookup.selectedRecordId}'. A 'Text' component is also visible. On the right, a configuration panel for the 'Text' component shows fields for 'Label' (Record ID), 'API Name' (Recordid), and 'Default Value' (also containing the formula '{!Custom_Lookup.selectedRecordId}', which is highlighted with a red box). Navigation buttons like 'Pause', 'Previous', 'Finish', 'Cancel', and 'Done' are present.

ReactiveFlowTest

Contact

 Andy Young ✕

Record ID

0030600002DplCmAAJ

Finish