



USER MANUAL

Universal Polymorphic Lookup

A metadata-driven Lightning Web Component that unifies multiple lookup relationships into a single, searchable UI

VERSION	AUTHOR	API VERSION	LAST UPDATED
1.7	Bart Debruyne	66.0	May 2026

Table of Contents

1 Introduction

2 Key Features

3 Use Cases

4 Component Placement

5 Configuration Properties

6 Custom Metadata Configuration

7 Configuration Examples

8 Hover Card

9 Search Behavior

10 Limitations

11 Disclaimer



1 Introduction

The **Universal Polymorphic Lookup** is a Lightning Web Component (LWC) that consolidates multiple lookup relationships on a single record into one unified search-and-select interface. Instead of exposing individual lookup fields (e.g., Account, Contact, Opportunity) scattered across a page layout, this component presents them as a single "Related To" field where users can search across any of the configured target objects.

All behavior—which objects are searchable, which fields they map to, and what fields are used for search—is driven entirely by a **Custom Metadata Type** record (`Polymorphic_Lookup_Config__mdt`). This means administrators can create and modify configurations without deploying code.

This manual provides a comprehensive guide for configuring and utilizing the Universal Polymorphic Lookup to meet your specific business needs.

Metadata-Driven Configuration

All object mappings, search fields, and multi-select behavior are defined in Custom Metadata, enabling zero-code configuration changes.

Multi-Object Search

Search across multiple Salesforce objects from a single input. An entity selector dropdown lets users switch between configured objects.

Single & Multi-Select Modes

Link one record at a time (single mode) or allow one record per configured object simultaneously (multi-select mode).

Hover Card with Compact Layout

Hovering over a linked record pill displays a popover showing the record's compact layout fields, with a link to open the full record.

Automatic Object Icons & Labels

The component dynamically resolves each object's SLDS icon and translated label, respecting org-level object renames.

Configurable Search Fields

Define which fields to search per object (e.g., **Name**, **Phone**, **Email**). Supports SOSL with automatic SOQL fallback.

Subtitle Display

Search results and linked pills show a subtitle derived from the first non-Name search field, giving users more context at a glance.

Read-Only Mode

Toggle read-only to hide the search bar and remove buttons, displaying linked records as non-interactive pills.

Field Labels on Pills

Optionally display the lookup field label (e.g., "Account", "Asset") above each pill, so users always know which relationship a linked record belongs to.

Duplicate Object Support

Map the same object to multiple lookup fields (e.g., Primary Account and Secondary Account). Each slot is independently selectable, with field labels shown automatically to distinguish them.

Reactive Refresh

When a mapped lookup field changes on the parent record (e.g., via another component), the UI automatically refreshes to reflect the new state.

Flow Output Variables

When used in a Flow Screen, the component exposes selected record IDs, names, and entity values as output variables for downstream flow logic.

Security: All server-side queries use `WITH USER_MODE` to enforce CRUD and Field-Level Security (FLS). Record updates use the Lightning Data Service (`updateRecord`), which also respects user permissions.

The Universal Polymorphic Lookup is ideal for scenarios where a record needs to be associated with one of several possible target objects through separate lookup fields:

Case “Related To” Field

Link a Case to an Account, Contact, Opportunity, or Asset through a single unified search bar instead of separate lookup fields on the page layout.

Activity Logging

Associate a custom Activity or Log record with any of several related objects (Account, Contact, Lead, Opportunity) from one component.

Service Requests

Allow service agents to link a request record to a Product, Asset, or Entitlement using one search interface, streamlining data entry.

Custom Junction Records

When a custom object has lookup fields to multiple parent objects (e.g., `Project__c` with lookups to Account, Contact, and Opportunity), present them as a single polymorphic selector.

Flow-Based Record Creation

Embed the component in a Flow Screen to let users search and select a related record during a guided flow, with the selection available as a flow variable for downstream processing.

Multi-Object Association

Enable multi-select mode to link a record to one Account *and* one Contact simultaneously, each written to its own lookup field.

Component Placement

The component can be added to the following Salesforce page types:

Lightning Record Pages

Lightning Flow Screens

How to add: In Lightning App Builder, open the component panel on the left and search for **sfdcldb Polymorphic Lookup**. Drag it onto the desired region of your page layout. Once placed, configure the properties in the right-hand panel.

Configuration Properties

The following properties are available when adding the component to a Lightning Page or Flow Screen via App Builder or Flow Builder.

5.1 Record Page Properties

When placed on a Lightning Record Page, `recordId` and `objectApiName` are automatically injected by the framework.

PROPERTY	TYPE	REQUIRED	DESCRIPTION
<code>configDeveloperName</code>	String	Yes	The Developer Name of the <code>Polymorphic_Lookup_Config__mdt</code> record that defines the mapping, search fields, and multi-select behavior. Default: <code>Default_Config</code> .
<code>componentLabel</code>	String	No	The label displayed above the component. Default: <code>Related To</code> .
<code>isReadOnly</code>	Boolean	No	When <code>true</code> , the search bar and remove buttons are hidden. Linked records are displayed as non-interactive pills. Default: <code>false</code> .
<code>allowReadOnlyNavigation</code>	Boolean	No	When <code>true</code> (and <code>isReadOnly</code> is also <code>true</code>), linked record pills become clickable links that navigate the user to the selected record. When <code>false</code> , read-only pills are non-interactive. Default: <code>false</code> .
<code>showFieldLabels</code>	Boolean	No	When <code>true</code> , the lookup field label (e.g., "Account", "Asset") is displayed above each pill so users can identify which relationship it represents. This is automatically enabled when the configuration contains duplicate objects (multiple lookup fields pointing to the same object). Default: <code>false</code> .

5.2 Flow Screen Properties

When placed in a Flow Screen, `recordId` and `objectApiName` must be passed in as input variables. The component also exposes output variables for downstream flow logic.

PROPERTY	DIRECTION	TYPE	DESCRIPTION
<code>recordId</code>	Input	String	The ID of the record to associate lookups with. Required.
<code>objectApiName</code>	Input	String	The API name of the object that <code>recordId</code> belongs to. Required.
<code>configDeveloperName</code>	Input	String	Developer Name of the CMDT config record. Required.
<code>componentLabel</code>	Input	String	Label displayed above the component. Default: <code>Related To</code> .
<code>isReadOnly</code>	Input	Boolean	Hides the search bar and remove buttons when <code>true</code> .
<code>allowReadOnlyNavigation</code>	Input	Boolean	When <code>true</code> and <code>isReadOnly</code> is enabled, record pills become clickable links that navigate to the linked record. Default: <code>false</code> .
<code>showFieldLabels</code>	Input	Boolean	Displays the lookup field label above each pill. Automatically enabled when duplicate objects are configured. Default: <code>false</code> .
<code>selectedRecordId</code>	Output	String	The ID of the selected record (single-select mode).
<code>selectedRecordName</code>	Output	String	The name of the selected record (single-select mode).
<code>selectedRecordIds</code>	Output	String	Comma-separated list of all linked record IDs (multi-select mode).
<code>selectedRecordNames</code>	Output	String	Comma-separated list of all linked record names (multi-select mode).
<code>selectedEntityValues</code>	Output	String	Comma-separated list of Object API Names matching each selected record.

Example — Flow Usage: In a Screen Flow for creating a Case, pass in the new Case's `recordId` and `objectApiName = "Case"`. After the user selects a related Account, the `selectedRecordId` output variable contains the Account ID, which can be used in subsequent flow elements.

Custom Metadata Configuration

The entire behavior of the component is controlled by a single `Polymorphic_Lookup_Config__mdt` Custom Metadata Type record. Each record defines which objects are available, how they map to lookup fields, and which fields are searchable.

6.1 Custom Metadata Fields

FIELD	TYPE	REQUIRED	DESCRIPTION
<code>Mapping_Config__c</code>	Long Text Area	Yes	Comma-separated list of <code>ObjectApiName:LookupFieldApiName</code> pairs. Each entry maps a target object to the lookup field on the parent record that stores the relationship. See format below.
<code>Search_Fields_Config__c</code>	Long Text Area	Yes	Pipe-delimited list of <code>ObjectApiName:Field1,Field2,...</code> entries. Defines which fields to search per object. The first non-Name field is used as the subtitle in search results.
<code>Allow_Multiple_Relationships__c</code>	Checkbox	No	When checked, the component allows one linked record per configured object simultaneously. When unchecked (default), only a single linked record is permitted at a time.

6.2 Mapping Config Format

The `Mapping_Config__c` field uses a simple comma-separated format:

```
ObjectApiName1:LookupField1, ObjectApiName2:LookupField2
```

Example: To allow linking a Contact record to an Account or an Opportunity, where the Contact has `AccountId` and a custom `Opportunity__c` lookup field:

```
Account:AccountId, Opportunity:Opportunity__c
```

Important: Each mapping entry must follow the exact format `ObjectApiName:FieldApiName`. The field must be a valid lookup field on the parent object. Invalid formats will cause a validation error on component load.

6.3 Search Fields Config Format

The `Search_Fields_Config__c` field uses a pipe-delimited format, with comma-separated fields per object:

```
ObjectApiName1:Field1,Field2 | ObjectApiName2:Field1,Field2,Field3
```

Example: To search Accounts by Name, Phone, and Website, and Contacts by LastName and Email:

```
Account:Name,Phone,Website | Contact:LastName,Email
```

Subtitle Logic: The first field in the search fields list that is *not* the display field (typically `Name`) and not `Id` is automatically used as the subtitle in search results and on linked pills. In the example above, `Phone` would be the subtitle for Account results, and `Email` for Contact results.

6.4 Single vs. Multi-Select Mode

The `Allow_Multiple_Relationships__c` checkbox controls how many records can be linked simultaneously:

MODE	CHECKBOX	BEHAVIOR
Single Select	Unchecked	Only one linked record at a time. Selecting a new record replaces the existing one. If the new record maps to a different lookup field, the old field is automatically cleared.
Multi-Select	Checked	One record per configured object is allowed simultaneously. For example, one Account <i>and</i> one Contact can be linked at the same time. The entity selector only shows objects that have not been linked yet. Once all objects are linked, the search row is hidden.

6.5 Field Labels & Duplicate Object Mappings

Starting with version 1.7, the component supports mapping the **same object to multiple lookup fields**. This is useful when a record has separate lookups pointing to the same object type—for example, a `Primary_Account__c` and a `Secondary_Account__c` field both referencing the Account object.

Mapping Format for Duplicate Objects

List each field as a separate entry in the `Mapping_Config__c` field, even though they share the same object API name:

```
Account:Primary_Account__c, Account:Secondary_Account__c, Contact:ContactId
```

How the Component Handles Duplicates

- When the component detects that the same object appears in more than one mapping entry, it automatically enables **field labels on pills** so the user can tell which lookup field each linked record belongs to.
- The entity selector dropdown shows the **translated field label** (e.g., "Primary Account", "Secondary Account") instead of the object label, preventing confusion.
- Internally, each configured **lookup field** is treated as an independent slot. Selecting a record for "Primary Account" does not affect the "Secondary Account" slot.

Tip: Even when duplicate objects are not configured, you can enable field labels manually by setting the `showFieldLabels` property to `true` in App Builder. This can improve clarity on configurations with many different objects.

6.6 How to Create a Configuration

- 1 In Salesforce Setup, navigate to **Custom Metadata Types**.
- 2 Find **Polymorphic Lookup Config** and click **Manage Records**.
- 3 Click **New** and enter a **Label** and **Developer Name** (e.g., `Case_Related_To`).
- 4 Fill in **Object to Field Mapping** with the comma-separated `Object:Field` pairs.
- 5 Fill in **Search Fields Configuration** with the pipe-delimited field lists per object.
- 6 Check **Allow Multiple Relationships** if multi-select is desired.
- 7 Click **Save**. Use the Developer Name as the `configDeveloperName` property on the component.

Complete Example — Case Page: Create a CMDT record with Developer Name

`Case_Polymorphic` , Mapping Config = `Account:AccountId, Contact:ContactId, Asset:AssetId` , Search Fields = `Account:Name,Phone | Contact:LastName,Email,Phone | Asset:Name,SerialNumber` , and Allow Multiple = `true` . Then add the component to the Case record page with `configDeveloperName = "Case_Polymorphic"` .

7 Configuration Examples

Below are complete, ready-to-use examples of `Polymorphic_Lookup_Config__mdt` records for common business scenarios. Each example shows the exact values to enter in the Custom Metadata fields and the matching App Builder property.

7.1 Case “Related To” — Multi-Select

Link a Case to an Account, Contact, and/or Asset simultaneously. The agent sees all three relationships in a single component.

FIELD	VALUE
Label	Case Related To
Developer Name	<code>Case_Related_To</code>
Mapping Config	<code>Account:AccountId, Contact:ContactId, Asset:AssetId</code>
Search Fields Config	<code>Account:Name,Phone,Website Contact:LastName,Email,Phone Asset:Name,SerialNumber</code>
Allow Multiple Relationships	Checked

App Builder property: `configDeveloperName = "Case_Related_To"`

Result: The component shows an entity selector with Account, Contact, and Asset. In multi-select mode, the user can link one Account *and* one Contact *and* one Asset at the same time. Each selection is saved to its respective lookup field on the Case.

7.2 Opportunity “Related Record” — Single-Select

Allow users to associate an Opportunity with either a Campaign or a Contract, but only one at a time.

FIELD	VALUE
Label	Opportunity Related Record
Developer Name	<code>Opportunity_Related_Record</code>
Mapping Config	<code>Campaign:CampaignId, Contract:ContractId</code>

FIELD	VALUE
Search Fields Config	Campaign:Name,Type,Status Contract:ContractNumber,Status
Allow Multiple Relationships	Unchecked

App Builder property: `configDeveloperName = "Opportunity_Related_Record"`

Result: The user can search and link either a Campaign or a Contract. Selecting a new record automatically replaces the previous one. If the user switches from Campaign to Contract, the `CampaignId` field is cleared and the `ContractId` field is populated.

7.3 Custom Object with Duplicate Account Lookups

A custom `Service_Request__c` object has two Account lookups: `Requesting_Account__c` (the account requesting the service) and `Servicing_Account__c` (the account performing the service), plus a Contact lookup.

FIELD	VALUE
Label	Service Request Parties
Developer Name	<code>Service_Request_Parties</code>
Mapping Config	<code>Account:Requesting_Account__c, Account:Servicing_Account__c, Contact:ContactId</code>
Search Fields Config	<code>Account:Name,Phone,BillingCity Contact:LastName,Email,Phone</code>
Allow Multiple Relationships	Checked

App Builder properties: `configDeveloperName = "Service_Request_Parties"` , `showFieldLabels = true`

Result: Because Account appears twice, the entity selector automatically shows the **field labels** "Requesting Account" and "Servicing Account" instead of just "Account". Field labels are also displayed above the pills. The user can link one record per slot: a Requesting Account, a Servicing Account, and a Contact—all independently.

7.4 Flow Screen — Lead Qualification

Use the component inside a Screen Flow to let an agent link a newly created Lead to an existing Account or Opportunity during a qualification flow.

FIELD	VALUE
Label	Lead Qualification Lookup
Developer Name	Lead_Qualification
Mapping Config	Account:Company__c, Opportunity:Related_Opportunity__c
Search Fields Config	Account:Name,Industry,Phone Opportunity:Name,StageName,Amount
Allow Multiple Relationships	Checked

Flow Screen input variables:

- `recordId` = the Lead's record ID
- `objectApiName` = "Lead"
- `configDeveloperName` = "Lead_Qualification"

Result: During the flow, the agent can search for and link an Account and/or Opportunity. After the screen, use the `selectedRecordId` and `selectedEntityValues` output variables to route the flow or update fields on the Lead.

7.5 Read-Only View with Navigation

Display all linked relationships on a record page in read-only mode, where clicking a pill navigates to the related record.

FIELD	VALUE
Label	Project Relationships
Developer Name	Project_Relationships
Mapping Config	Account:Account__c, Contact:Primary_Contact__c, Opportunity:Opportunity__c
Search Fields Config	Account:Name,Phone Contact:LastName,Email Opportunity:Name,StageName
Allow Multiple Relationships	Checked

App Builder properties: `configDeveloperName = "Project_Relationships"`, `isReadOnly = true`, `allowReadOnlyNavigation = true`, `showFieldLabels = true`

Result: The component displays all linked records as clickable pills with field labels. No search bar or remove buttons are shown. Clicking a pill navigates the user to that record. This is ideal for a "Related Parties" summary card.

When a user hovers over a linked record pill, a popover card appears displaying the record's compact layout fields. This provides a quick preview without navigating away from the current page.

8.1 How It Works

- 1 Hover over any linked record pill. After a short delay (300ms), the popover appears above the pill.
- 2 The popover header shows the record name, the object's SLDS icon, and a link icon that opens the full record in a new tab.
- 3 The body displays the record's compact layout fields in a two-column grid using `lightning-record-form` in readonly mode.
- 4 A spinner is shown while the compact layout fields load for the first time. Subsequent hovers on the same record render instantly from cache.
- 5 Moving the cursor to the popover keeps it open. Moving away from both the pill and the popover closes it after a short delay (200ms).

Compact Layout: The hover card automatically uses the compact layout configured for the target object in Setup. To control which fields appear in the hover card, customize the object's compact layout in Salesforce Setup > Object Manager > [Object] > Compact Layouts.

Note: The hover card respects Field-Level Security. Fields in the compact layout that the current user cannot read are automatically hidden by `lightning-record-form`.

9 Search Behavior

The component provides a two-part search interface: an entity selector dropdown on the left and a text search input on the right.

9.1 Entity Selector

The dropdown on the left side of the search bar lists all configured target objects. Each option displays the object's translated label (e.g., "Customers" instead of "Account" if the object has been renamed in the org). When the configuration contains duplicate objects (the same object mapped to multiple lookup fields), the dropdown shows the **translated field label** instead (e.g., "Primary Account", "Secondary Account") to distinguish between the fields. Selecting an entity switches the search context to that object.

9.2 Search Input

Typing in the search field triggers a server-side search after a 300ms debounce. A minimum of **2 characters** is required before a search executes.

9.3 SOSL / SOQL Strategy

The search uses a two-tier strategy for maximum compatibility:

1. **SOSL First:** A **FIND** query searches across all fields of the target object. SOSL supports multi-word search terms and is the preferred method.
2. **SOQL Fallback:** If SOSL fails (e.g., the object does not support search), the component falls back to a **LIKE**-based SOQL query using only the configured search fields. Field type compatibility is validated automatically—non-text fields (e.g., dates, numbers) are excluded from the **LIKE** clause.

Result Limit: Both SOSL and SOQL queries return a maximum of **20 results**. Results display the record's primary display field as the title and the first applicable search field as a subtitle.

9.4 Display Field Resolution

The component automatically detects the correct display field for each object. Objects without a standard **Name** field are handled with built-in fallbacks:

OBJECT	DISPLAY FIELD
Most objects	Name

OBJECT	DISPLAY FIELD
WorkOrder	WorkOrderNumber
Case	CaseNumber
Contract	ContractNumber
Order	OrderNumber
Task / Event	Subject
Objects with none of the above	Id

9.5 Selecting a Record

- 1 Choose the target object from the entity selector dropdown.
- 2 Type at least 2 characters in the search input. Results appear in a dropdown below.
- 3 Click a result to link it. The record is saved to the parent record's lookup field immediately.
- 4 A pill appears showing the linked record's name and object icon.

9.6 Removing a Record

Click the ✕ button on a linked record pill to unlink it. The corresponding lookup field on the parent record is set to `null` and a confirmation toast is shown.

Important: Removing a linked record clears the lookup field on the parent record in the database immediately. This action cannot be undone from within the component.

One Record Per Lookup Field in Multi-Select **1 per field**

In multi-select mode, only one record can be linked per configured lookup field. If the same object is mapped to multiple fields (e.g., `Primary_Account__c` and `Secondary_Account__c`), each field can hold one record independently. Otherwise, each object slot accepts one record.

Search Fields Per Object **15 fields max**

Each object in the `Search_Fields_Config__c` supports a maximum of 15 search fields. Exceeding this limit will cause a validation error on component load.

Search Results **20 results max**

Both SOSL and SOQL queries are capped at 20 results per search. Users should refine their search terms if the desired record is not visible.

Minimum Search Characters **2 characters**

The search will not execute until at least 2 characters have been typed. Single-character searches are not supported.

SOSL Not Available on All Objects

Some custom objects or objects with search disabled will not return results via SOSL. The component automatically falls back to SOQL in these cases, but SOQL `LIKE` queries only match configured search fields, not all fields.

Hover Card Requires Compact Layout

The hover card popover renders the target object's compact layout. If no compact layout is configured, the default system compact layout is used. Fields hidden by FLS will not appear in the hover card.

Non-Text Fields in SOQL Fallback

When the SOQL fallback is used, only text-compatible fields (String, Text Area, Email, Phone, URL, Picklist, Multi-Picklist, Combobox, Reference) support `LIKE` matching. Date, Number, and Checkbox fields are excluded from the `WHERE` clause.

Lookup Field Must Be Editable

The component writes to the parent record's lookup field using `updateRecord` . If the lookup field is not editable by the running user (due to FLS, page layout, or validation rules), the save will fail with an error toast.

No Cross-Object Relationship Fields in Search

The search fields configuration only supports direct fields on the target object. Relationship fields (e.g., `Account.Name`) are not supported as search fields.

Special Characters Sanitized in SOSL

For SOSL searches, special characters are stripped from the search term. Only alphanumeric characters and spaces are passed to the `FIND` clause. The SOQL fallback preserves the original term for `LIKE` matching.

This component was developed by **Bart Debruyne**. It is provided "as is" without warranty of any kind, express or implied. The author disclaims all liability for any damages, errors, or issues arising from its use. Use at your own risk.



© 2026 Bart Debruyne. All rights reserved.

Universal Polymorphic Lookup • Version 1.7 • User Manual