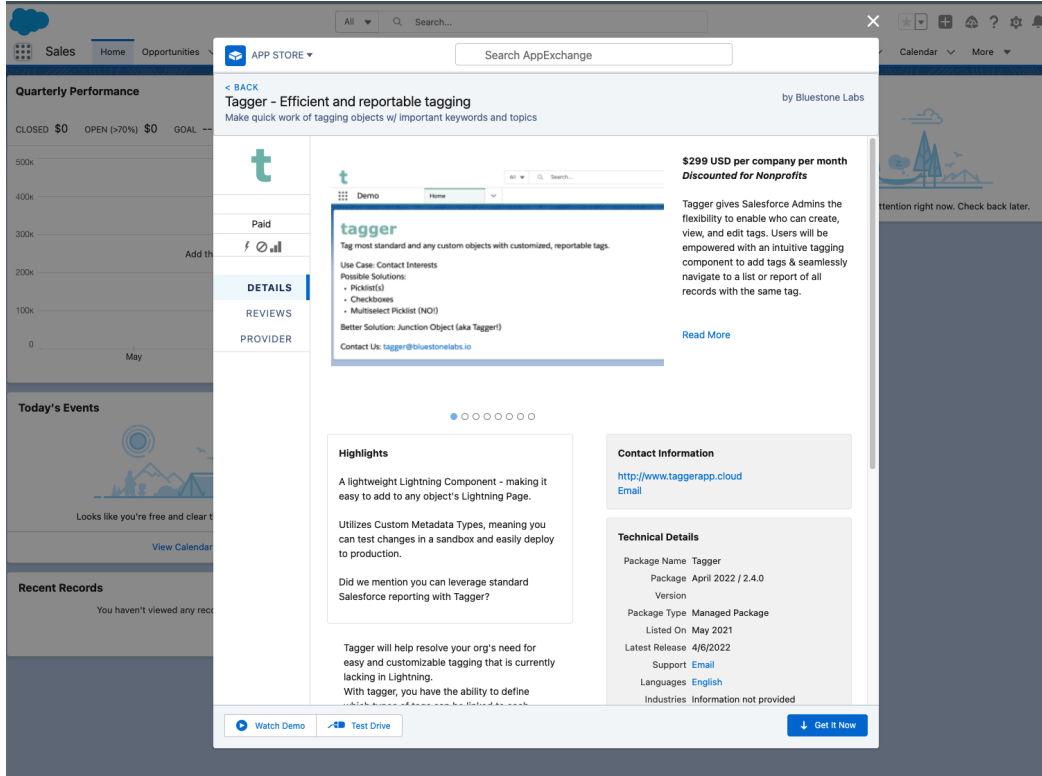
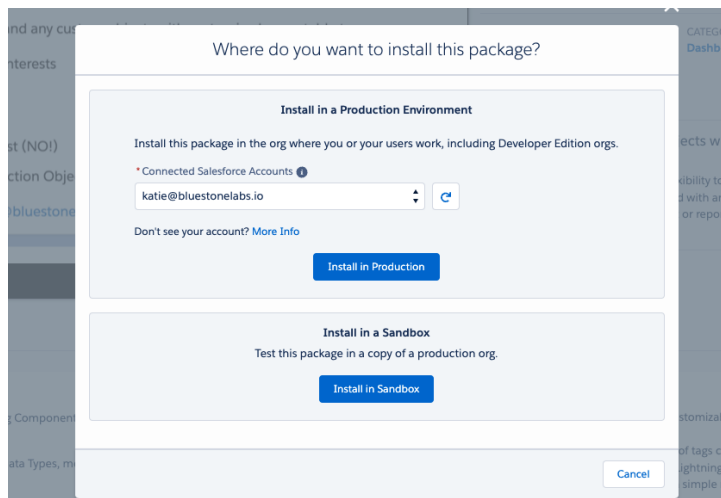


Installation and Configuration of Tagger from the AppExchange

1. Go to the Salesforce App Exchange and search for [Tagger](#), from the App Store click “Get it Now”



2. If you are not logged in to the AppExchange, click to Login and “Allow” access to Salesforce.
3. If you are in the AppStore within your Salesforce org, you will be redirected to the full AppExchange website to complete the install process.
4. Choose if you would like to install the package in your Sandbox for testing, or your Production Org.



- When installing Tagger from the AppExchange, it's recommended to choose: **“Install for All Users”** so that Apex Class access for the Lightning Components does not have to be manually assigned through the “Tagger User” Permission Set

App Name	Publisher	Version Name	Version Number
Tagger	Bluestone Labs Tagger	January 2022	2.3

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

****OR Manually Grant Access to Tagger End Users via Permission Sets**

The only permissions needed for a front-end User to use Tagger are the Apex Class permissions for the Lightning Components. The Apex Classes are automatically added to all Profiles if you choose **“Install for All Users”**. If you did not choose **“Install for All Users”** and want to assign permission manually, you will need to add the “Tagger User” Permission Set to the necessary Users:

- Setup -> Permission Sets -> Tagger User -> Manage Assignments -> Add Assignments

****OR Manually Grant Access to Tagger Setup via Permission Sets**

If you choose “Install for All Users”, the **Tagger Setup** tab and associated Apex Classes are visible to all Profiles. However, any User without the “Customize Application” Permission (permission needed to create Custom Objects and Fields) will receive an error message when they click on the **Tagger Setup** tab. If you did not choose Install for All Users and want to assign permission manually, you will need to add the “Tagger User” Permission Set to the necessary Users:

- Setup -> Permission Sets -> Tagger User -> Manage Assignments -> Add Assignments

- The last step is to assign Tagger Licenses to each User that needs to use the app (whether front-end Tagger Lightning Component or backend **Tagger Setup** tab):
 - Setup -> Installed Packages -> Manage Licenses (on left of Tagger row)

SETUP **Package Manager**

Package Details [Help for this Page](#) ?

Tagger
[Back to Previous Page](#)

Enable for Platform Integrations

Package Name	Tagger	Publisher	Bluestone Labs Tagger
Status	Active	Allowed Licenses	25
Expiration Date	Does not Expire	Used Licenses	2
Enabled for Platform Integrations	<input type="checkbox"/>		

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | **All**

Licensed Users

Action	Full Name ↑	Role	Active	Profile
Remove	Evatt, Katie		✓	System Administrator
Remove	Luehrs, Benjamin		✓	System Administrator

CONGRATULATIONS, you have successfully installed the Tagger app - the app that will make quick work of tagging any* (*most!*) standard and custom object records with important tags, that you can report on!

Next, let's learn more about Tagger, and go over some of the solution design so you can surprise and delight your end users with how easily they can now tag any record in Salesforce.

Technical Description:

Tagger is a lightweight Lightning Component that can be placed on Lightning Record Pages (using the Lightning App Builder) to easily add and remove Tags (junction object records) between the current Source Object record and Target Object records. Tagger is completely admin-customizable, with no development required. And because the Tags are stored as junction objects, they are easily reportable through standard Salesforce reporting (unlike other common solutions like multi-select picklists or standard Salesforce solutions to tagging).

Technical Architecture:

- A. There are 2 Custom Metadata Types (CMDT) included when you install Tagger. These CMDT records will help you setup the admin-defined customization of Tagger. After installing the package into your Salesforce Org, navigate to Setup > Custom Metadata Types. (You can type the text "meta" into the Admin Menu Quick Find as a shortcut!) Go ahead and navigate to the Custom Metadata Types page in Setup, so you can view these as we discuss what they do.



Search Setup



Setup Home Object Manager

meta

Custom Code

Custom Metadata Types

Didn't find what you're looking for? Try using Global Search.



SETUP

Custom Metadata Types

All Custom Metadata Types

[Help for this Page](#)

Custom metadata types enable you to create your own setup objects whose records are metadata rather than data. These are typically used to define application configuration that need to be migrated from one environment to another, or packaged and installed.

Rather than building apps from data records in custom objects or custom settings, you can create custom metadata types and add metadata records, with all the manageability that comes with metadata: package, deploy, and upgrade. Querying custom metadata records doesn't count against SQL limits.

New Custom Metadata Type

Action	Label	Installed Package	Namespace Prefix	Visibility	API Name	Record Size	Description
Manage Records	Tagger Component	Tagger	Tagger	Public	Tagger__Tagger__mdt	506	Defines the Source Object on which the Tagger resides and the general configuration of the Tagger.
Manage Records	Tagger Target Filter	Tagger	Tagger	Public	Tagger__Tagger_Object__mdt	1857	Defines the Target Object the Tag Object connects to from the Source Object and the customization of the Target Object search and Tag Object creation.

1. **Tagger Target Filter:** This Custom Metadata Type will capture details on :
 - a. The Source Object
 - i. What object will the Tagger be placed on?
 - b. The Tag Object
 - i. This is where your Tags will be saved
 - ii. This object is the junction object (or joiner object) that should be created by an Admin in your org using Tagger Setup or using standard Admin Setup. It will have Lookup Relationship or Master-Detail Relationship fields connecting to both the Source Object and Target Object
 - c. The Target Object
 - i. This is the object in Salesforce which will be searched for and to which the Tag records will connect the Source record.
 - d. Query Options
 - i. SOSL vs. SOQL Search
 - ii. Target Filter
 - iii. Suggested Target Records - Last Modified, Last Viewed, or Custom
 - iv. Target Search Limits
 - v. Target Order By

Tagger Target Filter

[Edit Layout](#) | [Help for this Page](#)

[← Back to List: Tagger Target Filters](#)

Tagger Target Filter Detail		Edit	Delete	Clone
▼ Information				
Label	Contact Tag	Protected Component		<input type="checkbox"/>
Tagger Target Filter Name	Contact_Tag	Namespace Prefix		
Tagger Component	Contact			
▼ Source Object				
Source Object	Contact			
▼ Tag Object				
Tag Object	Tag_Link	Source Lookup	Contact	
Tag Record Type Name	Tag	Target Lookup	Tag	
Title Override	Tag	Tag Icon Name	custom:custom45	
Tag Order	1	Allow Tag Duplicates?	<input type="checkbox"/>	
▼ Target Object				
Target Object	Tag	Tag Label	Tag	
Target Record Type Name	Tag	Tag Subtitle 1	Category	
Create New Target Records?	After No Results Found	Tag Subtitle 2	Number	
		Tag Subtitle 3	Description	
▼ Query Options				
Target Query Type	sosl	Target Search Limit	Tag	
Target Filter	Tag	Target Order By	Tag	
Suggested Target Records	viewed	Custom Suggested Target Records	Tag	

- ❖ One **Tagger Component** can hold multiple **Tagger Target Filters** for the same Target Object (e.g. different Record Types, different Target Objects).

2. **Tagger Component:** This Custom Metadata Type will define the Source Object on which the Tagger resides and the general configuration of the Tagger.
 - a. *Title* - This will be the title displayed on the Tagger Component when you initially drag and drop the Component onto the record page in Lightning App Builder
 - b. *Source Object* - The object where the component will be placed, and the object record page that end users will be adding tags to.
 - c. *Display Type*: pill vs. tile
 - d. *Save Mode*: immediate vs. confirm (with save)
 - e. *Icon Name*: select which SLDS icon you want to display next to the component title, or you may leave as “None”.
 - f. *Hide Option* - If only one Tagger Target Filter will be leveraged on a particular Tagger Component, you can select this checkbox to hide the “dropdown selection” icon from the component itself (since only one Tagger Target Filter is available to pick from).
 - g. *Target Object Navigable?* - This setting determines whether Users can navigate to the Target Object record by clicking on the tag pill or tag tiles.
 - h. *Read Only?* - This setting will dictate if Tags should be Read Only, meaning end users will not be able to add or delete the tags on a record, only view them.

SETUP
Custom Metadata Types

Tagger Component Help for this Page ?

[Tagger Target Filters \(1\)](#)

Tagger Component Detail Edit Delete Clone

Label	Contact	Protected Component	<input type="checkbox"/>
Tagger Component Name	Contact	Namespace Prefix	

▼ Tagger Details

Title	Tagger	Display Type	tile
Source Object	Contact	Save Mode	immediate
Default Tagger Target Filter	Contact_Tag	Icon Name	
Hide Option?	<input checked="" type="checkbox"/>	Read Only?	<input type="checkbox"/>
Target Object Navigable?	<input checked="" type="checkbox"/>		

▼ System Information

Created By [Katie Evatt](#), 9/15/2021, 10:35 PM Last Modified By [Katie Evatt](#), 9/20/2021, 2:13 PM

Edit Delete Clone

Tagger Target Filters

Action	Label	Tagger Target Filter Name	Target Object	Tag Object	Tag Label	Tag Order
Edit Del	Contact_Tag	Contact_Tag	Tag	Tag_Link	Tag	1

❖ Multiple Tagger Components can be defined for the same Source Object.

B. There is 1 Lightning Component called **“Tagger”** which is found in the “Custom - Managed” section of the Lightning Components list within the Lightning App Builder (look near the bottom of the list). The Tagger Lightning Component can be placed anywhere on a Lightning Record Page. We like placing it somewhere easily visible and intuitive for end-users to utilize the tagging functionality you are building for them.

- ❖ **Tagger** Lightning Component displays 1 simple picklist of your CMTD **“Tagger Component”** records that match for the Source Object of the Lightning Page you are editing. (E.g. - If you are adding Tagger to your Contact Lightning Layout, you will need a CMTD Tagger Component where the Source Object = Contact)
- ❖ All other customizations for the Tagger are controlled within the CMTD **“Tagger Component”** and **“Tagger Target Filter”** records.

C. There is 1 Tab and Lightning Component called **“Tagger Setup”** which can be found by navigating to the Tagger Setup tab. Tagger Setup walks Admins through creating and/or finding the architecture needed to take advantage of Tagger. This includes creating and connecting the needed Source Object, Target Object, Tag Object, Target and Source Lookup Fields on Tag Object, and the Tagger Component and Tagger Target Filter Custom Metadata Records.

Tagger Setup

This process guides you through setting up an Object to be used in Tagger, which can involve creating:

- a new Target Object with optional Tab*
- a new Tag Object with optional Tab and with new Source and Target Lookup Fields*
- new Tagger Component and Tagger Target Filter to allow use of Tagger with just created Objects

The first two steps can also be completed manually by using the Object Manager in Setup. The last step can also be completed manually: Go to 'Setup' -> 'Custom Metadata Types' -> 'Tagger Component - Manage Records' to create or adjust your Tagger Component and/or 'Tagger Target Filter - Manage Records' to create or update a Tagger Target Filter.

Object and Field Setup

Source Object Next

* Source Object
Select an Option

Target Object ?

Tag Object ?

Source Lookup Field on Tag Object ?

Target Lookup Field on Tag Object ?

Custom Metadata ?

Save Reset

- ❖ **Note on Standard Profiles:** For any Custom Object created through it, Tagger Setup automatically adds Create, Read, Update, and Delete permissions to System Administrator and Custom Profiles. However, since Standard Profiles cannot be granted permissions to Custom Objects, if you use Standard Profiles, you will need to add the Create, Read, Update, and Delete permissions along with Read and Edit Field Permissions for the new Custom Objects to an existing or new Permission Set and assign it to those desired Users with Standard Profiles.

- ❖ Note on Tabs: If you chose to automatically create Tab(s) for a new Target Object and/or Tag Object, those “Default On” Visibility was added for those Tabs to the System Administrator, the User’s Profile creating the objects through Tagger Setup (if Custom Profile), and the first 8 Custom Profiles ordered oldest to newest. Because of a Salesforce quirk that we hope is fixed soon, Tab Visibility set to “Default On” through the Metadata API is not properly visible, and Admins will need to navigate to each Profile where the Tab(s) should be visible, go to “Object Settings”, go to each Object, click “Edit” and then immediately “Save” with no changes (leaving Visibility as “Default On”). The Tab(s) should now be visible.