



Processity Data History

User Guide

Package version 1.161

Table of Contents

Table of Contents	2
A Note on Naming	2
Introduction	3
Observability	3
This Guide	3
Glossary of Terms	4
Installation	4
Install the Application from the Salesforce AppExchange	4
Grant Permissions for Application Setup	6
Verify the Installation	7
Setup	8
Setting Up the Application	8
Object Triggers	12
Asynchronous Apex	15
Lightning Record Page/Flexipage Component	16
Moving Processity Data History Configuration Between Salesforce Environments	20
Configuring Data Management	23
Configuring Data Store	26
Using the Application	29
Permissions Layer Personas	29
Permissions Layer Summary	30
Limitations	30
Known Issues	33
End User Guide	34
Generating Span Events (SEs)	34
Viewing Span Events (SEs)	34
Viewing Processity Data History Data Consumption	42
Data History Data Snapshots	43
Data History Performance Monitoring	48
Get Started with Processity Data History	55
Processity Data History for Process Mining	55

A Note on Naming

You might find references to this application's old name, 'Audicity', in some parts of the application where renaming would be disruptive to existing customers, or in areas that are impossible to rename due to the constraints of Salesforce managed packaging. For upgrading customers, you might continue to see the 'Audicity' application name for non-upgradable artefacts (for example: the application's name in App Launcher) - you can manually update these artefacts to use the 'Processity Data History' name or leave the legacy application name at your discretion.

If you're in any doubt about an application artefact please don't hesitate to contact us via support@processity.ai.

Introduction

Business systems and systems of record don't just track the current state of your business. They also provide valuable historical context of what has happened. As more technologies promise to use data to transform business, the value of this historical data rises. No longer is it the case that just companies with critical standards and regulations require a more full and accurate historical record of their work. Every business stands to benefit from a complete historical data record.

Processity Data History was created to be this historical record. Using principles and designs founded in the open telemetry standard and implemented on the Salesforce platform, with Processity Data History you can ensure that every important data change is recorded permanently. And with precision that tracks to the millisecond, Processity Data History ensures that even the most high velocity business process is recorded accurately.

Processity Data History offers these key features:

- Historical data captured for every Salesforce object which is queryable, updatable and supports Apex triggers.
- Unlimited tracked fields per object including an easy "track all fields" option.
- Changes captured to millisecond precision.
- Transaction traces which can include user-initiated updates, Apex automation, Flow automation.
- Attribution of asynchronous Apex execution to origin transactions.
- Synchronous or asynchronous trace capturing.
- On platform or public cloud trace storage.

Observability

Because of its use of open telemetry as its basis for tracing, Processity Data History can also form an important part of your Salesforce organization's observability story. Observability works well when you complement a logging tool with an all-the-time automatic tracing tool like Processity Data History, which can be especially useful in your observability story when looking for intermittent or sudden errors and performance problems.

This Guide

This document provides information on the steps to configure and use Processity Data History in a Salesforce environment.

It provides guidance for administrators to configure and operate the application, and for end users of the application.

Glossary of Terms

- Processity Data History Action Scheduler (PAS) - scheduled Apex that invokes background tasks that keep Processity Data History running smoothly in your org., and manages scheduled tasks such as Data History Data Management and Data History Data Store.
- Data History Data Management (PDM) - the automated system for managing Data History span event data retention rules and policies.
- Data History Span Events (SEs) - In pure Salesforce terms, these are the custom objects that Processity Data History uses to store data. SEs are the smallest unit of observability work. They may contain information about field changes, metrics, etc. They represent modifications recorded against a Salesforce record, as a result of in-scope user change or system changes initiated by an in-scope user, to such a record. In-scope changes are considered to be a modification of an object record where tracking has been enabled on said object, and specifically a modification of a field on the object record that has been configured by the 'Processity Data History Tracking Administrator' as being subject to tracking.
- 'Trackable' objects - Salesforce objects that are described by Salesforce as supporting Apex Triggers, are *updatable*, and are *queryable*.¹
- 'Trackable' fields - Salesforce fields on 'trackable' objects that are described by Salesforce as *updatable* or is a roll-up summary field.²

Installation

The following steps should be followed to install the application in your Salesforce environment. The installation steps need to be performed by a System Administrator user in your Salesforce environment.

Install the Application from the Salesforce AppExchange

Installation of the application should be initiated from the application's AppExchange listing, via your browser.

Steps:

1. Navigate to the AppExchange Listing:
<https://appexchange.salesforce.com/appxListingDetail?listingId=8ecf5cc2-cc0d-4292-9d22-ff5a73568828&tab=e>
2. Click the **Get It Now** button.

¹ See 'Salesforce Apex Reference Guide - DescribeSObjectResult Class' (https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_methods_system_sobject_describe.htm) for further information.

² See 'Salesforce Apex Reference Guide - DescribeFieldResult Class' (https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_methods_system_fields_describe.htm) for further information.

3. Log in to your **Trailblazer.me** profile.
4. Choose which environment you would like to install the application into:

Where do you want to install this package?

Install in a Production Environment

Install this package in the org where you or your users work, including Developer Edition orgs.

* Connected Salesforce Accounts ⓘ

Don't see your account? [More Info](#)

Install in Production

Install in a Sandbox

Test this package in a copy of a production org.

Install in Sandbox

[Cancel](#)

5. Provide contact details *if you're prompted* and click **Next**.
6. Select the installation level **Install for Admins Only**.

NOTE: While Processity Data History will function using any of the installation level options, any choice besides **Install for Admins Only** risks bypassing your object, field, and sharing security settings. Doing so will potentially give access to sensitive data to users who should be restricted. For this reason, we strongly recommend the default option of **Install for Admins Only** for the vast majority of customer use-cases. In this way only System Administrators should be initially enabled for the application.

Installation Level	Profiles Enabled	Processity Data History Permissions Granted	Recommended Installation Level
Install for Admins Only	System Administrator	All	Most installations
Install for All Users	All Profiles		<u>Strongly Not Recommended</u>
Install for Specific Profiles	Profiles you select		Some installations

App Name	Publisher	Version Name	Version Number
Audicity	Processity.ai		

Description
Complete Salesforce Audit Trail and Field History

Additional Details [View Components](#)

7. Initiate the installation by clicking **Install**.
8. The installation process will execute (in some cases, long-running installations will alert you on completion by email)

Grant Permissions for Application Setup

To set up the application, System Administrator users will require a set of application permissions provided by the Permission Set Group *Processity Data History Tracking Administrator*. To add a Permission Set Group assignment for a user, follow these steps:

1. Click the cog icon  in the top right of the browser window and click **Setup** (opens a new window).
2. In the left hand setup menu click on **Users > Users**.
3. Select the desired user from the list of users.
4. Scroll down to the **Permission Set Group Assignments** related list (this may be empty or will contain any pre-existing Permission Set Group Assignments for this user):
5. Click **Edit Assignments**.
6. Select **Processity Data History Tracking Administrator** and move it to the right-hand column:

Available Permission Set Groups

- Processity Data History Trace Viewer
- Processity Data History Trace Writer

Enabled Permission Set Groups

- Processity Data History Tracking Administrator

Add

Remove

Save Cancel

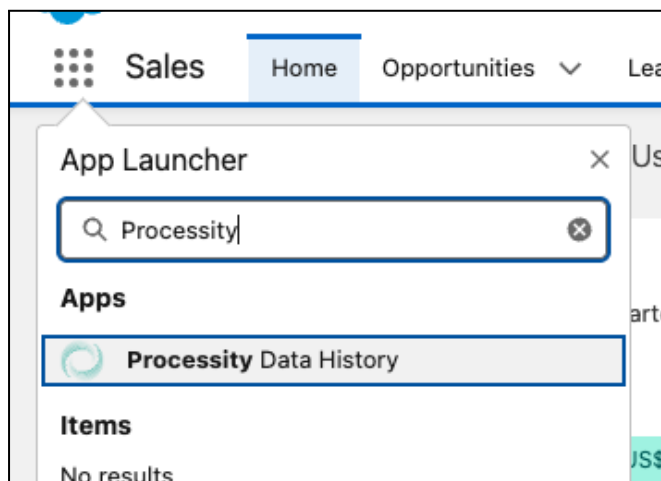
Save Cancel

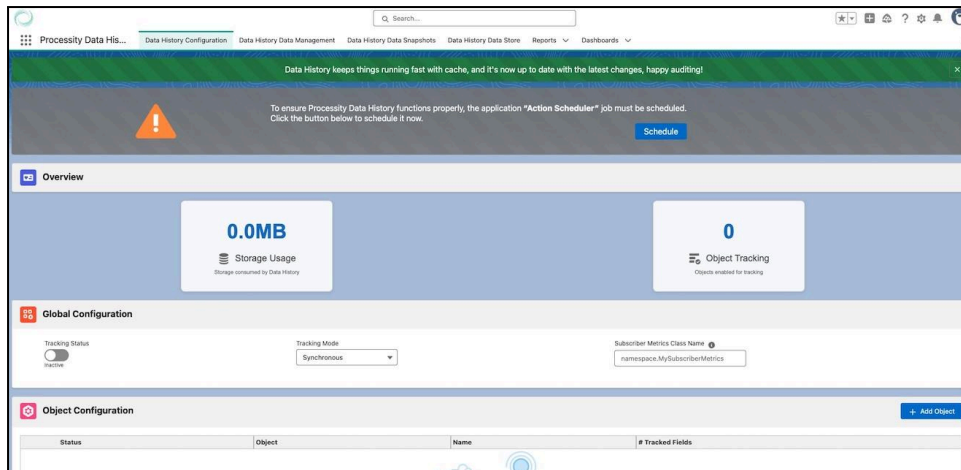
7. Click **Save**.

You should now see the Processity Data History Tracking Administrator Permission Set Group assignment (along with any other pre-existing Permission Set Group assignments) in the user's 'Permission Set Group Assignments' related list:

Verify the Installation

Once the application has been successfully installed and any additional administration users granted the necessary permissions, you can verify the installation and permissions by accessing the **Processity Data History** application in the App Launcher as a user with the *Processity Data History Tracking Administrator* permission set group:





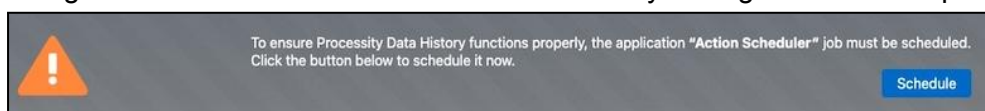
Setup

Setting Up the Application

These steps explain the configuration actions to control the application's core operation and are represented in two levels - Global, and Object configurations. And the Action Scheduler.

Action Scheduler

Processity Data History Action Scheduler (PAS) is responsible for a number of background tasks that keep your installation of Processity Data History running smoothly, reducing redundant storage of Span Events (SEs), and enables the execution of functionality such as Data History Data Management and Data History Data Store. Once PAS is scheduled, these background tasks will execute without further intervention. On installation of the application, or if PAS has been unscheduled in your org., a banner prompt will display on the data History configuration tabs. Click **Schedule** to add PAS to your org's scheduled apex.

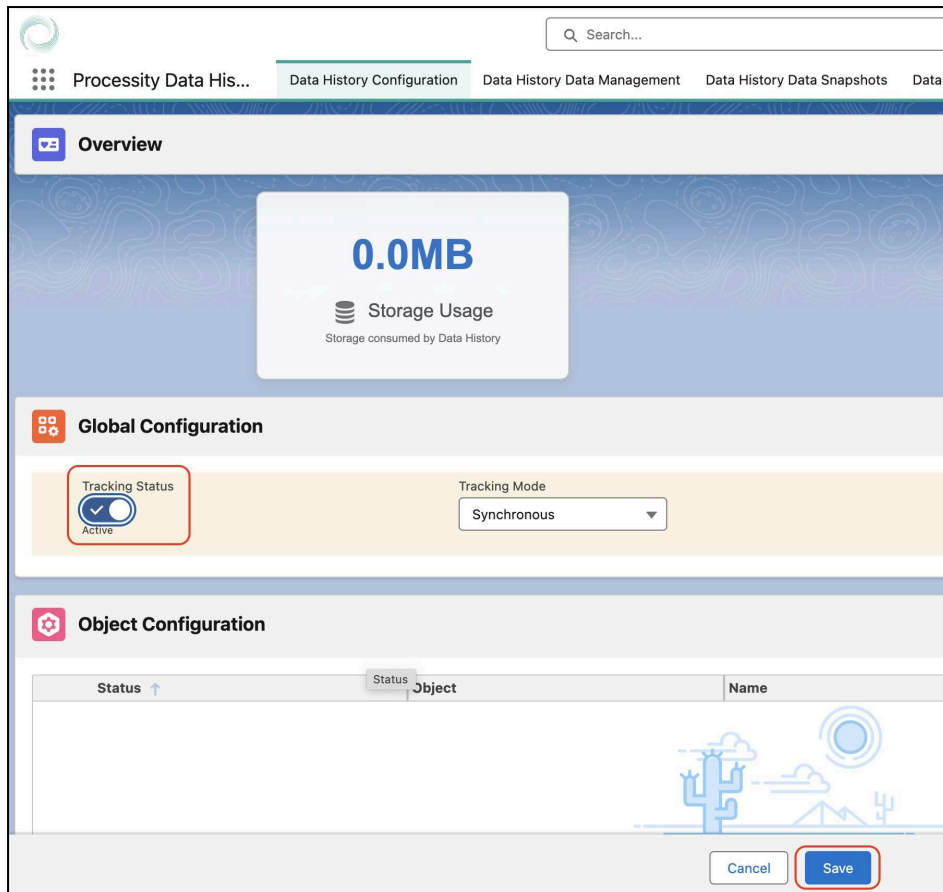


Global Configuration

Tracking Status

The Global Configuration **Tracking Status** can be thought of as the master switch for the application's operation. This configuration overrides all other configurations - if set to Inactive the package will not generate any Data History Span Events (SEs) (See [Glossary of Terms](#) for definition).

To enable Processity Data History - set the **Logging Status** to *Active* and click **Save**:



Tracking Mode

The Global Configuration **Tracking Mode** gives you control over the mechanism used for Processity Data History processing. This setting allows Data History processing (the initial registration of the potential tracked changes and the comparison of before and after values on tracked object fields) to occur in one of three modes:

1. Synchronous
All Data History processing occurs in the same transaction as the changes being tracked (initial registration of the potential tracked changes and the comparative processing to determine and record before and after changes)
2. Queueable
Splits the registering of the potential tracked changes, which occurs in the same transaction as the changes being tracked, and the comparative processing to determine and record before and after changes, which execute in a separate Queueable transaction. This mode is recommended for organisations that have existing issues with governor limit exceptions, or that have very small governor limit margins³ (I.e., your existing transactions execute very close to governor limits and there is not enough spare margin to introduce Data History processing)

³ See Salesforce Developers - Execution Governors and Limits (https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm) for more information.

3. Platform Events

As with Queueable - this option is suitable for organisations with existing issues with governor limit exceptions, or that have very small governor limit margins. Splits the registering of the potential tracked changes, which occurs in the same transaction as the changes being tracked, and the comparative processing to determine and record before and after changes, which execute in a separate platform event transaction.

Recommended for use when Queueable mode is unsuitable due to a high levels of asynchronous Apex method executions⁴ in your organisation.

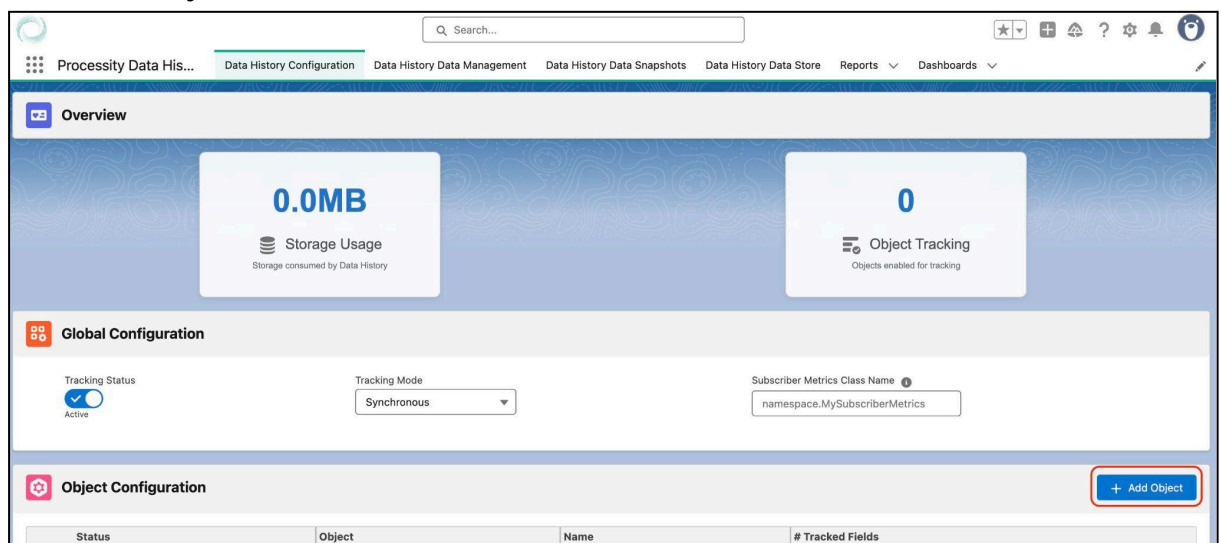
As **Tracking Mode** is part of Processity Data History's Global Configuration, the selected tracking mode applies to all objects enabled for Data History tracking.

Object Configuration

The Object Configuration represents the way the application should generate SEs for specific Salesforce objects. Any Salesforce object that supports Apex triggers, can be queried, and is updatable can be enabled for tracking⁵.

Add a new object configuration for a trackable object from **Data History Configuration**:

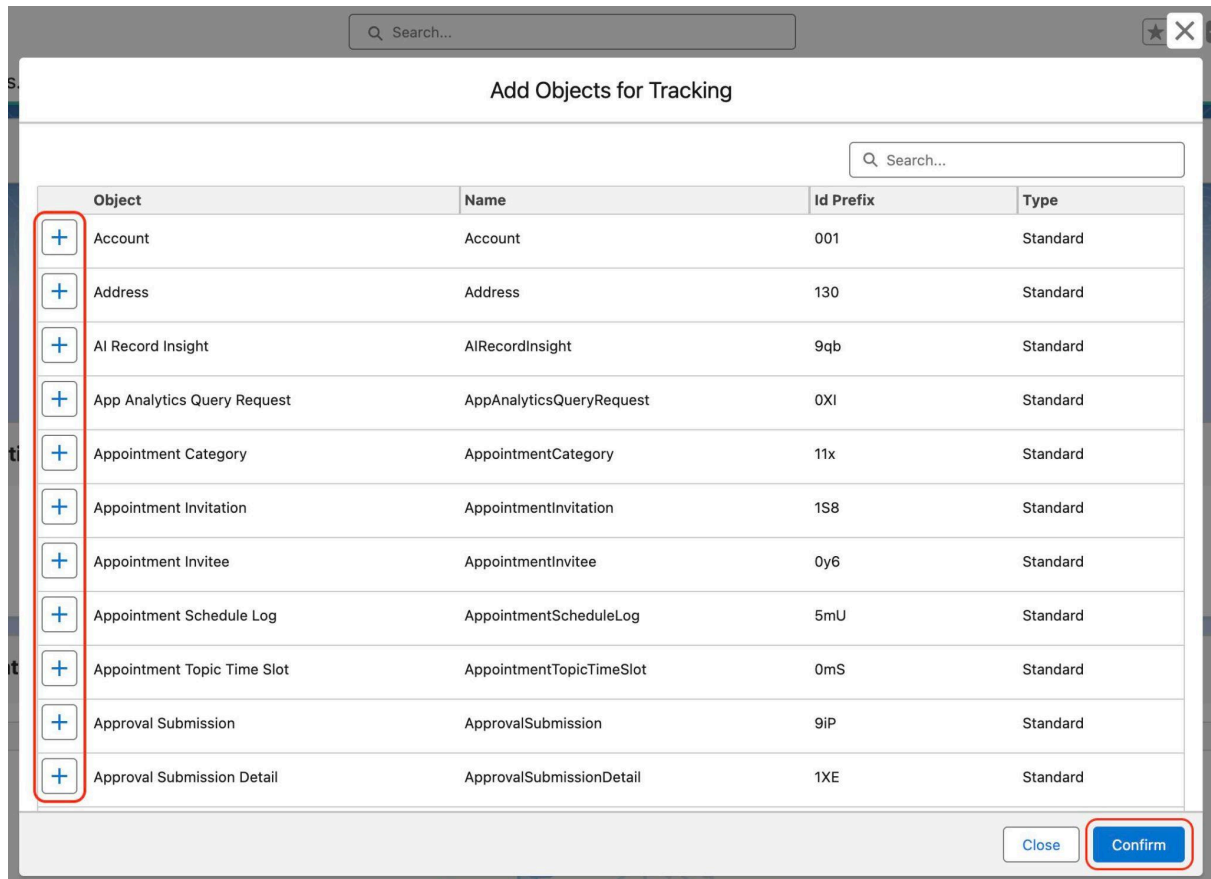
1. Click **Add Object**.



⁴ See 'Lightning Platform Apex Limits' in Salesforce Developers - Execution Governors and Limits (https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm) for more information.

⁵ 'Salesforce Tooling API Reference - EntityDefinition' https://developer.salesforce.com/docs/atlas.en-us.api_tooling.meta/api_tooling/tooling_api_objects_entitydefinition.htm

Select one or more objects using the *plus* icon in the left-most column:



2. Click **Confirm** once all your selections have been made

A trackable object can be enabled for tracking and configured with a number of parameters, as follows:

1. Status - the activation status of an object for tracking:
 - a. **Active** - *will* generate SEs for this object.
 - b. **Inactive** - *will not* generate SEs for this object.
2. Tracked Fields - the specific fields on an object that should be considered in-scope for tracking, can be set to **Log All Fields** or specific fields selected. Fields that are considered updatable by Salesforce⁶ can be enabled for tracking.
 - a. **Log All Fields** - if enabled, all trackable fields will be considered 'in-scope' and tracked on this object. **Any** modification to a trackable field, performed by a Processity Data History Trace Writer user (See [Processity Data History Trace Writer](#) for more) will generate SEs.

Note that this also applies to fields added to the object after Data History is configured. When **Log All Fields** is enabled, those new fields will also be

⁶ 'Salesforce Apex Reference Guide'

https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_methods_system_fields_describe.htm#apex_Schema_DescribeFieldResult_isUpdateable

tracked automatically.

OR

- b. Specific field selection - a selection of fields to be considered in-scope and tracked on this object (E.g., the Name and Owner fields on the Account Object) - *only* modifications to these fields, performed by a Processity Data History Trace Writer user (See [Processity Data History Trace Writer](#) for more), will generate SEs.

1. Status: Active

2a. Track All Fields: No

2b.

Field Label	Field Name	Data Type
+ Account Description	Description	textarea
+ Account Fax	Fax	phone
+ Account Name	Name	string
+ Account Number	AccountNumber	string
+ Account Phone	Phone	phone
+ Account Rating	Rating	picklist
+ Account Site	Site	string
+ Account Source	AccountSource	picklist
+ Account Type	Type	picklist

Once all your object and field selections have been made, be sure to click 'Save' to persist your configuration.

Object Triggers

As part of the enablement of an object for Processity Data History tracking, a trigger or trigger handler call must be implemented to enter Data History tracking for an object. This can be done with any of the standard tools, e.g. VS Code or Code Builder.

To get accurate information, it is important that Processity Data History is called as the first before-save trigger and the last after-save trigger. Any existing triggers should run between those two calls to Processity Data History.

Trigger

If your Salesforce environment does not implement a trigger handler framework, then Salesforce does not guarantee the order in which triggers will be called. In that case, it will not be possible to ensure that Processity Data History runs first and last. The result will be that all changes during a save are recorded, but the Trace Explorer may show inaccurate relationships about how one change leads to another.

The following trigger snippets can be used to create the required Apex artifacts. It should only be used for objects that have no other Apex triggers, or used as a guide to add Processity Data History to existing triggers.

This trigger snippet shows the pattern of entering Processity Data History tracking for a Salesforce object. A sample test class for test coverage is also provided (if required)⁷.

Trigger

```
None
trigger AccountDataHistory on Account(before insert, before update, before
delete, after insert, after update, after delete, after undelete) {

    mantra.AudicityApex.track();

}
```

Test Class Snippet

```
None
@Test
public class [ObjectName]DataHistoryTest {
    @IsTest
    public static void testInsert(){
        Test.startTest();
        [ObjectName] obj = new [ObjectName]([Set required fields]);
        insert obj;
        Test.stopTest();
        Assert.IsNotNull(obj.Id, 'The object Id should be set when the object is
inserted');
    }
}
```

Trigger Handler Calls

If your organisation implements a Trigger Handler Framework, entry to Processity Data History tracking for each enabled object can be made by implementing the following calls within the handler framework:

At the beginning of the before trigger context (i.e., the first operation in the before context):

⁷ See 'Salesforce Developers - Apex Developer Guide' for more:
([https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_code_coverage_in
tro.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_code_coverage_intro.htm))

None

```
mantra.AudicityApex.track();  
//All other calls/logic
```

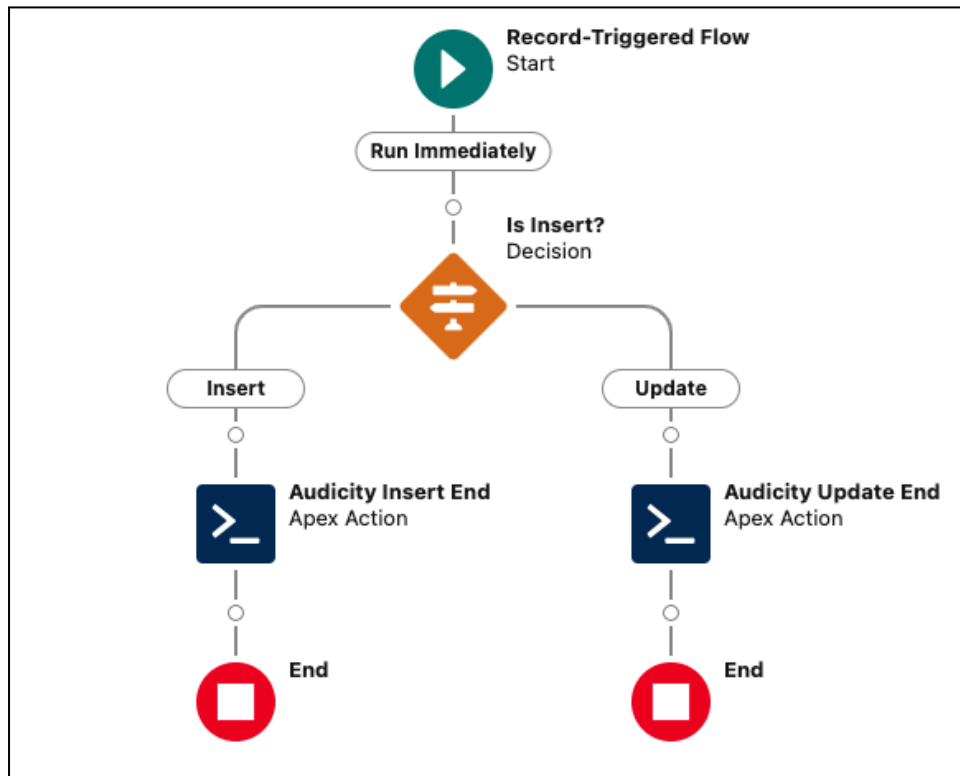
At the end of the after trigger context (i.e., the last operation in the after context):

None

```
//All other calls/logic  
mantra.AudicityApex.track();
```

Processity Data History After Save Flow [OPTIONAL]

If the object you are enabling for Processity Data History tracking has existing after save record triggered flows OR your organisation has other managed packages that implement triggers on the object, an after-save flow, at the end of any existing chain of after-save flows, can be implemented to capture these changes on the object by entering Processity Data History tracking in the insert and update contexts by calling Processity Data History invocable Apex methods:



For more information on Flows, see Salesforce Help - Flow Builder (<https://help.salesforce.com/s/articleView?id=sf.flow.htm&type=5>).

Asynchronous Apex

Processity Data History can track the relationship between asynchronous Apex jobs so that a single Trace in Processity Data History can include data from multiple Salesforce transactions. To do this, Processity Data History must be notified when a Queueable is enqueued and when the Queueable is executed.

Queueable

When enqueueing a Queueable job, call Processity Data History with the AsyncApexJob Id to record that the job originates from this transaction:

```
None
Id jobId = System.enqueueJob(myQueueable);
mantra.AudicityAsync.track(jobId);
```

Then, at the start of the execute method, call Processity Data History with the QueueableContext:

```
None
public void execute(QueueableContext context) {
    mantra.AudicityAsync.track(context);
    // ...
}
```

By making these two calls, Processity Data History is able to relate any events that happen within the Queueable execution to the original Trace generated in the first transaction.

If you do not add these calls to your code, Processity Data History will still track changes to tracked objects that are made by the Queueable class, but it will not relate them to the original Trace.

Lightning Record Page/Flexipage Component

To provide access to a set of SEs in the context of their traced records, a Lightning Web Component should be added to object Lightning Record Pages/Flexipages, as follows:

1. Navigate to the Lightning Record Page/Flexipage for an object to modify.
 - a. Navigate to an object record, click the cog  in the top right of the browser window, and click **Edit Page**.

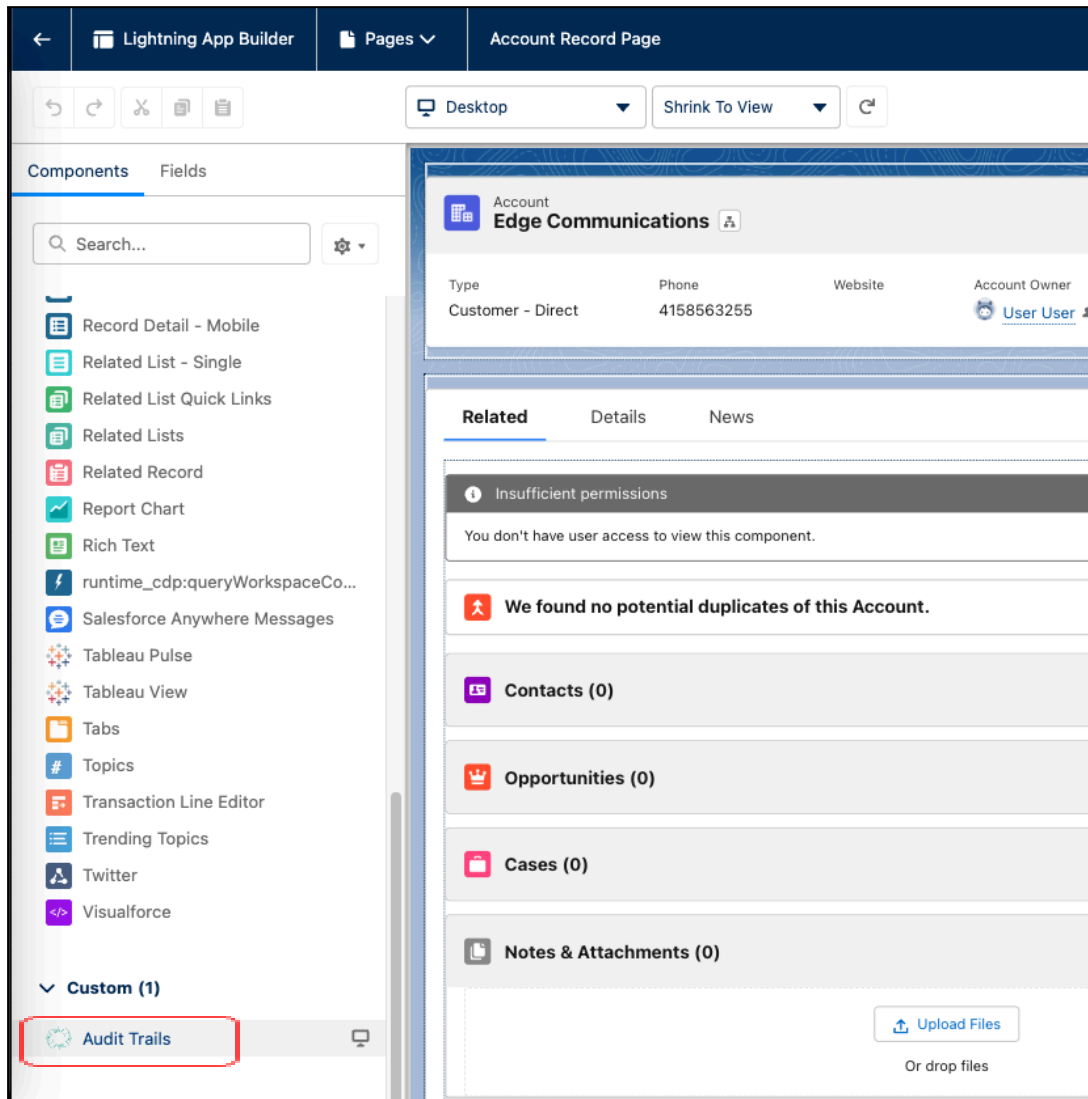
OR to edit a custom record page

- b. Click the cog  in the top right of the browser window, click **Setup**, click **Object Manager**, click on the object in question, and then click into Lightning Record Pages and finally, click into the Lightning Record Page/Flexipage to amend.

OR

Click the cog  in the top right of the browser window, click **Setup**, click into the **Lightning App Builder** via the Setup menu on the left of the browser window via the menu **User Interface > Lightning App Builder** menu and then finally, click into the Lightning Record Page/Flexipage to amend.

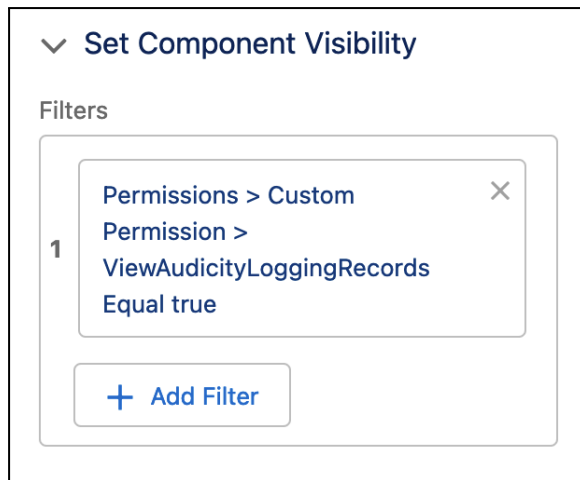
- On the Lightning Record Page/Flexipage, using the navigation on the left of the browser window, select the **Audit Trail** component under the **Custom** sub-menu.



- Add the **Audit Trails** component to one of the user interface slots on the page.
- [OPTIONAL] You can set a number of parameters for this component that impact the way it behaves:
 - Default Record Count** - the number of records to initially load, defaults to 10.
 - Records to Load More - the number of records to load when the end user clicks Load More in the [SE Field-by-field View](#), defaults to 20.
 - Fields to Exclude** - any specific fields to exclude from presenting as SEs in the component, for this specific Salesforce object, as a comma-separated list. E.g., 'StandardField, Custom_Field__c'.
The custom permission *View Processivity Data History Excluded Fields* can be used to override these exclusions, for specific users.

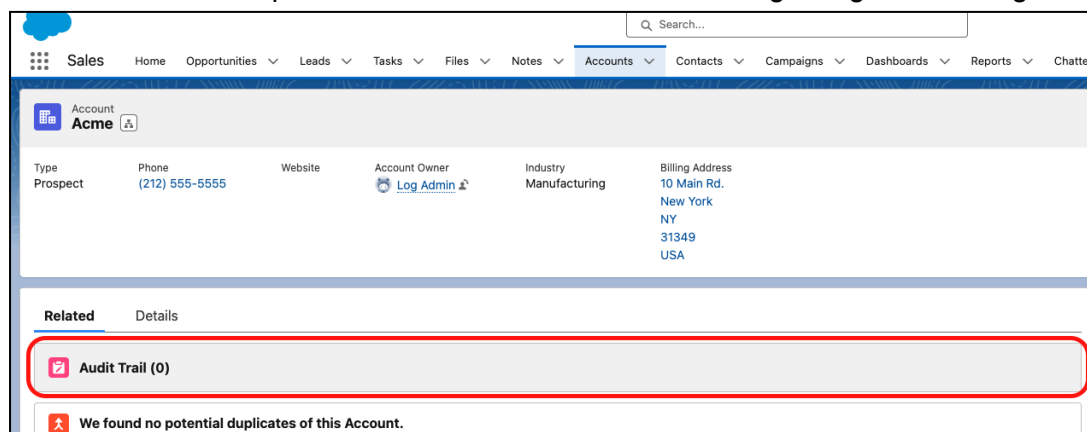
This is a convenience feature for fields end users need FLS read access to, but would otherwise not find the presentation of tracking history useful for said fields. This can be thought of as a soft exclusion of fields from just the presentation layer. (See [Viewing Processity Data History Span Events \(SEs\)](#) for more information)

5. [OPTIONAL] Set Component Visibility⁸. Users will only be able to view the Audit Trails component if they have the **View Processity Data History Traces** custom permission (provided by the 'Processity Data History Trace Viewer' and 'Processity Data History Administrator' Permission Set Groups). So, it is recommended to configure component visibility as below:



6. Click **Save** and click through any activation steps.

The Audit Trails component should now be visible on the Lightning Record Page⁹.



The Audit Trails component can also support objects that do not have their own detail or Flexipage - you can add multiple Audit Trails components to a parent object Flexipage, one component returning SEs related to the parent record, with one or more further components returning those SEs related to child records. Using 'Campaign' and 'Campaign Member' as an example where the 'Campaign Member' object does not have a record detail page, two Audit Trails components can be added to the Campaign Flexipage:

⁸ See 'Salesforce Help - Visibility Rules on Lightning Pages' for more:
https://help.salesforce.com/s/articleView?id=sf.lightning_page_components_visibility.htm&type=5

⁹ Depending on any 'Set Component Visibility' constraints you have applied.

The screenshot displays the 'Campaign Record Page' for a campaign named 'Sample Campaign'. The page is divided into several sections. On the left, there's a 'Related' section with two components highlighted by red boxes and numbered 1 and 2. Component 1 is 'Audit Trail (Campaign) (1)' and Component 2 is 'Audit Trail (Campaign Members) (1)'. Both components show a table of audit trail records. The right sidebar contains configuration options for the audit trails, including 'Title', 'Show Related Object Audit Trails', 'Default Record Count', 'Records to Load More', 'Grouped By Transaction', and 'Set Component Visibility'.

This close-up screenshot focuses on the 'Show Related Object Audit Trails' section in the right sidebar. It shows a search bar with the text 'CampaignMember (CampaignMembers)' entered, and a magnifying glass icon to the right. The section is highlighted with a red box.

The second component is scoped to return 'CampaignMember' records.

From a Campaign record, both components are now visible on the Flexipage, with the first returning SEs related to this Campaign, and the second returning SEs concerning Campaign Members related to this campaign, as shown below:

Type	Status	Start Date	End Date
Conference	Planned	14/03/2025	14/03/2026

Trace	Date	User	Summary
01e12681-471f-440c-8c39-a602f8cc47db	14/03/2025, 04:04 pm	User User	Record created with Name as Sample Campaign and Type as Conference and Status as Planned ... See More

Trace	Related Entity	Date	User	Summary
12425e5d-0eaa-45ae-a62a-8d06dbc6ae44	CampaignMember	14/03/2025, 04:27 pm	User User	Record created with Campaign ID as Sample Campaign and Lead ID as Joe Bloggs and Status as Sent
8f1dfdab-ae1c-4984-acbe-ef8ce968a11a	CampaignMember	14/03/2025, 04:10 pm	User User	Record created with Campaign ID as Sample Campaign and Lead ID as Jane Doe and Status as Sent ... See More

Moving Processity Data History Configuration Between Salesforce Environments

We recommend evaluating Processity Data History configurations in suitable test environments - it's important to evaluate how Processity Data History interacts with your existing environment configuration and customizations, as well as assessing the configuration against production-like data volumes in terms of normal usage and any bulk data processing your organisation performs in your Production environment.

Processity Data History Configuration Artifacts

The following artifacts should be considered when promoting configuration between environments:

- Custom Metadata Type - `mantra__AudicityConfig__mdt`
Represents the Processity Data History Global and Object Configuration records and should be promoted when:
 - Changing Global Processity Data History settings
 - Adding a new object for tracking
 - Changing a Processity Data History Object Configuration's Status
 - Changing a Processity Data History Object Configuration's field tracking
 - Adding or updating Processity Data History's Data Management Configuration (Global, Object or Hybrid configurations (see [Configuring Data Management](#) for more information))
- Apex artifacts
Apex Triggers or Classes added or modified to enable Processity Data History tracking
- Flow artifacts - see [Processity Data History After Save Flow](#) for more.

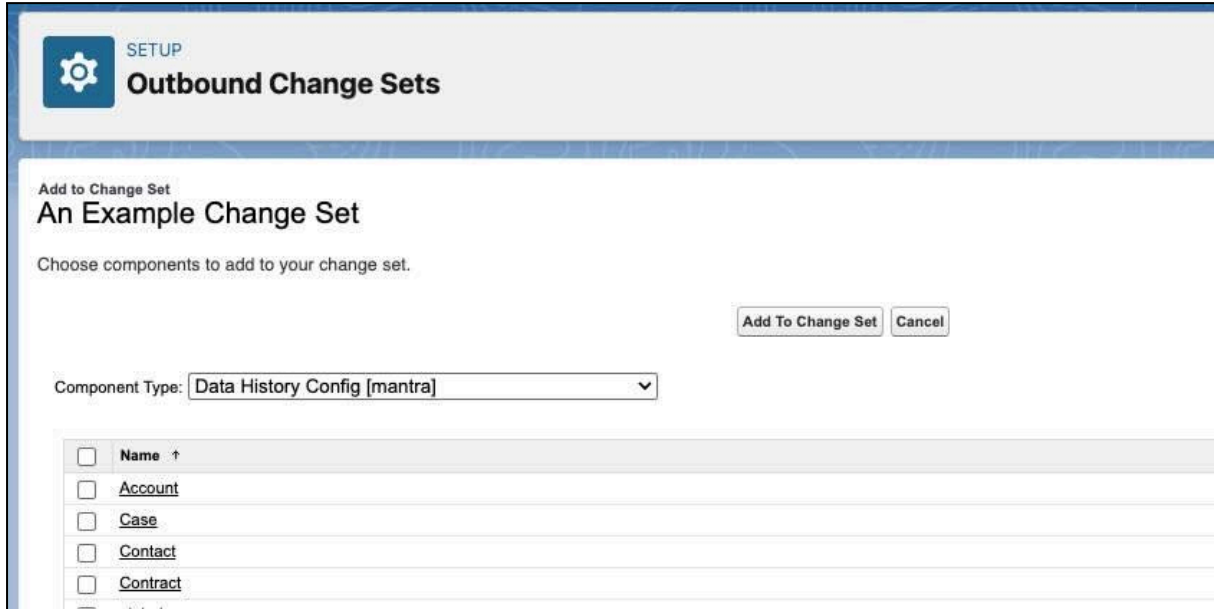
Once you've evaluated your Processity Data History Configuration and you have identified the artifacts to promote between your environments, you can use one of the following methods:

Using your existing CI/CD Setup

If your organisation has an existing CI/CD setup, this is recommended to promote Processity Data History Configuration artifacts, as these changes will then be subject to all your existing promotion checks and validations.

Using Salesforce Change Sets

Salesforce Change Sets can be used to promote Processity Data History Configuration artifacts between environments with established deployment connections (see **Setup > Environments > Deploy > Deployment Settings**)¹⁰. Processity Data History Configuration can be found under component type 'Data History Config [mantra]':



SETUP

Outbound Change Sets

Add to Change Set

An Example Change Set

Choose components to add to your change set.

Add To Change Set Cancel

Component Type: Data History Config [mantra]

☐	Name ↑
☐	Account
☐	Case
☐	Contact
☐	Contract

Select all the elements of Data History Configuration that have changed that you wish to promote to another environment¹¹.

Using Manual Promotion

There is a capability to make Production changes to Processity Data History Object Configurations (see [Object Configuration - Production Environments](#) for more) which can be used to manually promote evaluated Processity Data History Object Configurations from a test environment to a Production environment - this would require you to manually implement the configuration from the source environment into the target environment.

¹⁰ See 'Salesforce Help - Change Set Setup and Configuration' (<https://help.salesforce.com/s/articleView?id=000385608&type=1>) for more information.

¹¹ See 'Salesforce Help - Change Sets' (<https://help.salesforce.com/s/articleView?id=sf.changesets.htm&type=5>) for more.

Configuring Data Management

Processity Data History Data Management (PDM) is used to manage the data that Processity Data History generates (Data History Span Events - SEs). It is managed by choosing policies on how and when to delete SEs. 'Processity Data History Tracking Administrators' can access this functionality via the application tab **Data History Data Management**. The following terminology is used in the subsequent sections:

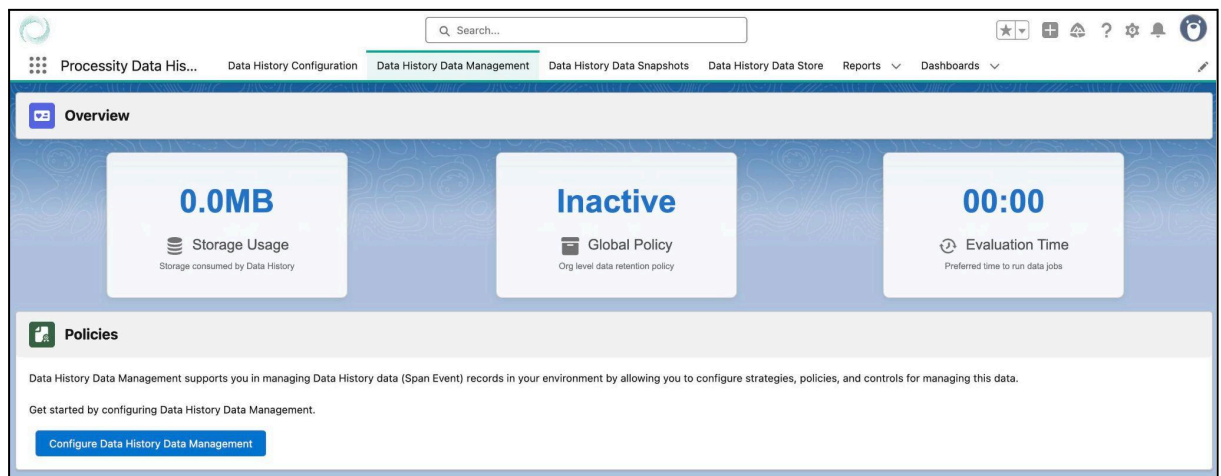
- Evaluation time - Data History Data Management will evaluate your configured policies daily, at a time you specify (executing as one or more chained Apex Queueable jobs)
- Level - the target of a data management policies' evaluation
- Strategy - what action to apply to eligible SEs
- Policy - determines the eligibility of SEs for a given strategy

Data History Data Management Tab

On navigating to the application tab Data History Data Management, you will see:

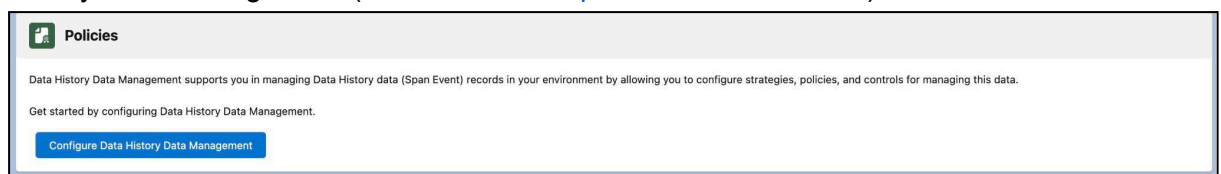
1. Overview section

This section displays data metrics focusing on Processity Data History's data usage and the data management configuration



2. Policies section

This section displays the policies you have configured. If no policies have previously been configured, a button **Configure Data History Data Management** will display - this will launch a guided setup wizard to help you in configuring Processity Data History Data Management (see [Guided Setup](#) for more information).



If Data History Data Management has previously been configured, you can additionally update individual policies (1.), re-enter the guided-setup process (2.), and

update Data Management's evaluation time (3.).

The screenshot shows the 'Policies' management interface. At the top right, there are two buttons: 'Enter Guided Setup' (labeled 2.) and 'Update Evaluation Time' (labeled 3.). Below these buttons, the text reads 'The configured data management policies for your environment:'. A search bar is located to the right of this text. The main part of the interface is a table with the following columns: Status, Level, Strategy, Policy, Policy Value, and Policy Unit. The table contains one row with the following values: Status is 'Active' (with a red box around the '1.' icon), Level is 'Global', Strategy is 'Retention', Policy is 'Time', Policy Value is '600', and Policy Unit is 'Days'. At the bottom left, there is a dropdown menu set to '50' and the text 'Total 1 rows'. At the bottom right, there are navigation arrows and a page number '1'.

Status	Level	Strategy	Policy	Policy Value	Policy Unit
1. Active	Global	Retention	Time	600	Days

3. Data Management Change Logs

Records the changes to the Data Management Configuration made from the Data History User Interface

Data Management Change Logs			
Search...			
Date	User	Configuration Name	Changes
08/09/2025, 11:30 am	Paul Bircham	Global	Policy set as Active and retaining data upto 600 Days
5 Total 1 rows			

Data History Data Management Guided Setup

The guided setup process will take you through the steps to configure **Data History Data Management**, explaining important terms and context as you move through the process.

Processity Data His...

Data History Configuration

Data History Data Management

Data History Data Snapshots

Data History Data Store

Reports

Dashboards

Processity Data History Data Management Configuration

Select a Level

Select a Strategy

Select a Policy

Acknowledge Policy

Configure Policy

Review the Configuration

The level you select controls the granularity with which Processity Data History Data Management evaluates your Span Event (SE) data.

Global-level

Considers all SE records, regardless of the specific object types their events are concerned with. Best for organisations wanting a simple policy that applies to all your environment's SEs

Object-level

Applies policies to SE records concerned with specific object types. Best for organisations where there are differing policy requirements between tracked objects. For example, SEs concerning Accounts must be retained for longer than SEs concerning Leads. Selecting 'Object-level' will deactivate any existing 'Global-level' configuration your environment has configured.

Hybrid-level

Applies a 'Global-level' policy that can be overridden at the Object-level. If an 'Object-level' policy is defined for an object, the Global-level policy does not apply to that object, for all objects **without** a 'Object-level' policy, the 'Global-level' policy applies. Best for organizations with a standard policy with a few object-level exceptions. For example, the 'Global-policy' is to retain SEs for 6 months, but SEs concerning Accounts must be retained for 2 years.

Configuration Level

☒ Global-level
 ☐ Object-level
 ☐ Hybrid-level

Previous

Next

A brief overview of the process and terminology presented:

1. Evaluation time

Data History Data Management will evaluate your configured policies daily, at a time you specify. This enables you to choose a time during the day that is best for your Salesforce environment (E.g., when there is minimal system processing performed by your org., or other applications).

- The time selected is local to your user's timezone.
- Data History Data Management jobs execute as one or more chained Apex Queueable jobs, **not** as batch Apex.

2. Level

Allows you to configure PDM to operate on one of three levels:

- Global - applies to all SEs
- Object - applies a policy to SEs related to a traced record of a specific object type, ignores all other SEs
- Hybrid - applies a global policy to all SEs without a specific object policy, and applies object policies

3. Strategy

For any SEs that are eligible for a policy, the action to take.

- a. Retention - SEs becoming eligible for a policy are hard deleted from your environment

4. Policy

Determines the eligibility of SEs for a given strategy

- a. Time - the amount of time in days before SEs should be considered eligible for a strategy, oldest SEs first. E.g., 365 - SEs should only be eligible 365 days after their creation, eligible in order of their creation date, oldest first
- b. Storage - the amount of storage for SEs, with SEs above this limit becoming eligible for the configured strategy, eligible in order of their creation date, oldest first

5. Acknowledgement

As data retention policies will result in SEs (Data History tracking records) becoming eligible for hard-deletion from your environment, any requirement to persist Processity Data History tracking data will require the implementation of an independent and ongoing backup or extraction process for this data before the data becomes eligible for your chosen data management policy. An explicit acknowledgement is required to proceed.

6. Policy-value

The value used to assess SE policy eligibility

- a. Time (Days) - the number of days before a SE is considered eligible (E.g., a policy of 100 days, means SEs 101 days and older become eligible for the policy, eligible in order of their creation date, oldest first)
- b. Storage (Gigabytes) - the amount of storage SEs can consume before becoming eligible for the policy (E.g., SEs can consume 5Gb of data storage before becoming eligible for the policy, with SEs in excess of 5Gb becoming eligible in order of their creation date, oldest first)

7. Review

Review the configuration you have selected before saving it.

8. Finish

Save the configuration and activate Processity Data History Data Management - persists the configuration ready for the next occurrence of Data Management at your chosen evaluation time.

Configuring Data Store

Data History Data Store provides the capability to store Span Event (SE) data outside the standard Salesforce custom object storage.

Data Store is particularly important for Salesforce organisations that generate a large volume of SE records that may use up their record storage allocation available to them on Salesforce, or that currently have a very high utilisation of their Salesforce record storage.

SEs are exported to the Data Store on a scheduled basis. They are then retrieved on-demand when users view Audit Trails or Trace Explorer.

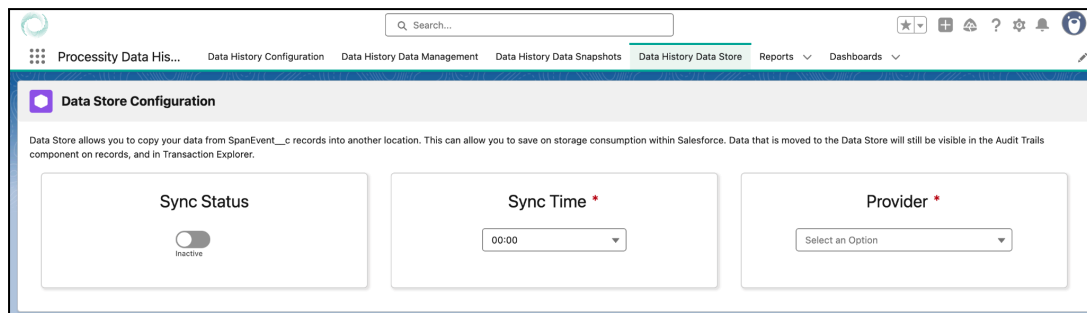
Data Store supports two providers:

1. Big Objects
Stores SE records using Salesforce Big Object definitions within your Salesforce environment.
2. Data History Data Vault
Stores SE records securely on Microsoft Azure cloud, localised to a region of your choice. (This is available as a purchased add-on to Processity Data History.)

To enable a Data Store provider for your Processity Data History installation, please contact support@processity.ai.

Once a provider has been enabled, Data History Data Store is configured the same regardless of the provider chosen.

Data History Data Store Tab



On the Data Store tab, an administrator can configure 3 aspects of Data Store:

- Sync Status - when enabled, SEs will be sent to Data Store and Data Store will be used when SEs are queried instead of just querying for SEs stored in Salesforce
- Sync Time - this is the time of day when a Data Store sync will run
- Provider - this is the Data Store provider you have selected (see section [Enabling Data Store](#) for further guidance)

Accessing Span Events (SEs) inside Data Store

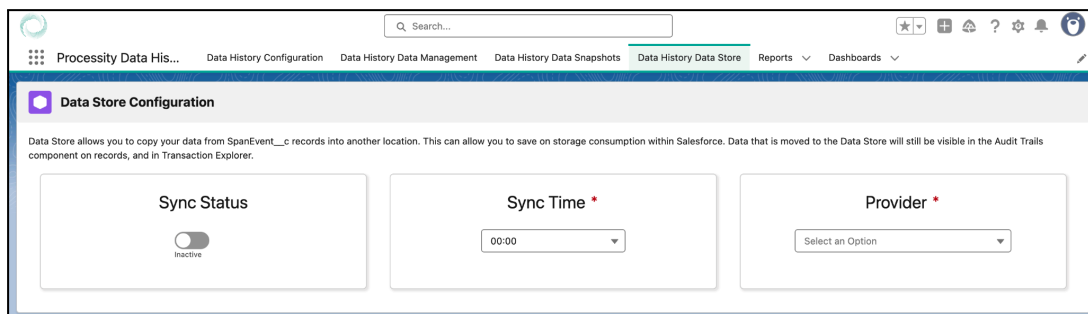
When Processity Data History Data Store is configured and activated, Data History components are capable of displaying SE data (such as in the [Audit Trails](#) component) whether the underlying SE data is stored on-platform in Salesforce, or off-platform in Data History Data Store.

Note: Processity Data History features [Data History Data Snapshots](#) and [Data History Data Management](#) are scoped to exclusively process SE records stored on the Salesforce platform, specifically excluding SE records stored in Data History Data Store.

Enabling Data Store

To enable Data History Data Store follow these steps:

1. Select your Data Store option:
 - a. If using Big Objects, install and configure the Big Object Data Store package using the Processity Data History Data Store Big Object [Quick Start](#).
 - b. If using Data Vault, email support@processity.ai to request the enablement of the Data Vault provider for your organisation. Our support team will:
 - i. Ensure all the necessary prerequisites for Data Vault are configured on the public cloud provider.
 - ii. Provide guidance on setting up your provider in your Salesforce environment.
2. Once your provider is enabled, navigate to the Processity Data History application's **Data History Data Store** tab.



3. Set the **Sync Status** to *Active*.
4. Select a Sync Time that is appropriate and convenient for your Salesforce environment to move data between Salesforce and Data History Data Store¹².
5. Select the Provider that has been created and enabled (in step 1)

¹² This time is always local to the configuring user's timezone, and can be selected according to your organisation's data processing loads, to avoid other scheduled jobs, or a time when your environment is under light usage loads.

Using the Application

While users are doing their work in Salesforce, the operation of Processity Data History writing data is invisible to them. And the reading of data operates just like the standard field tracking that they are already familiar with.

Permissions Layer Personas

The following permission personas are implemented for using the application.

Processity Data History Tracking Administrator

Permission Set Group: *Processity Data History Tracking Administrator*

These users administer the Data History application - such as controlling Global and Object configurations, can view SEs in the context of a traced record, and write SEs.

Processity Data History Tracking Administrator also includes the permissions of Processity Data History Trace Viewer, below.

Processity Data History Trace Viewer

Permission Set Group: *Processity Data History Trace Viewer*

These users can view SEs in two contexts.

First, they can view SEs in the context of a related record - for example, when a SE is generated against a specific Account record, Processity Data History Trace Viewer users can see the SEs generated in the Audit Trails component if this component has been added to the specific Lightning Record Page that a specific account record Lightning Record Page assignment resolves to (see [Viewing Processity Data History Span Events](#) for more). This requires the Processity Data History Trace Viewer user to have existing sharing access to the given account record.

Second, they can view SEs in the Processity Data History Trace Explorer, showing the SEs in the context of a Trace instead of a record.

Processity Data History Trace Writer

Permission Set Group: *Processity Data History Trace Writer*

These users create SEs by making edits to in-scope fields on records of objects that are active for Processity Data History tracking.

Permissions Layer Summary

Persona PSG	High-level Application Capability		
	Administer Processity Data History	View SEs (via package components)	Write SEs (via triggers and classes operating in system mode)
Processity Data History Tracking Administrator	X	X	X
Processity Data History Trace Viewer		X	
Processity Data History Trace Writer			X

Limitations

These are the documented limitations and known issues for this version of the application.

Objects and Types

The following Salesforce objects/types **cannot** be tracked:

- Custom Settings
- Custom Metadata Types

Salesforce objects with these characteristics **cannot** be tracked¹³:

- Any object that:
 - Does not support Apex Triggers
 - Is not updatable
 - Is not queryable

Standard Salesforce objects which do not have the ability to create a custom App Builder record page cannot display SEs because there is no configurable user interface in which to place the Audit Trails component¹⁴.

Fields

The following types of object fields **cannot** be tracked:

¹³ See 'Salesforce Tooling API Reference - EntityDefinition' for more (https://developer.salesforce.com/docs/atlas.en-us.api_tooling.meta/api_tooling/tooling_api_objects_entitydefinition.htm)

¹⁴ See '[Lightning Record Page/Flexipage Component](#)' for a potential workaround for child objects with this limitation, E.g., Campaign Members.

- Fields with these characteristics¹⁵:
 - Is not updateable
 - Is calculated¹⁶
- Some standard formula fields, including but not limited to:
 - Contract.EndDate

Transaction Characteristics

All Salesforce transactions (create and update operations) are constrained by Salesforce's governor limits¹⁷ that manage an individual Salesforce tenant's resource utilisation within Salesforce's multi-tenant architecture. This prevents any single tenant from over-consuming resources and impacting the performance of other tenant's orgs. Given the finite transaction resources imposed by these governor limits, there is, by consequence, a finite amount of transaction resources that can be utilised by Processity Data History whilst operating your configured tracking configuration within your org's transactions. As such, the transaction characteristics and the maximum number of fields that can be supported for tracking, per transaction, is influenced by a number of factors:

- Processing Batch Sizes
Large batch processing sizes within batch Apex will reduce the number of field changes that can be successfully tracked by Processity Data History. You can tune batch sizes to get the right balance between processing throughput and ensuring Processity Data History tracking has enough transaction resources to execute successfully.
- Configured Automations
As with the speed of your transactions, large volumes of automations, and/or inefficient automations, all consume transaction resources. The number of automations invoked by a transaction will impact the amount of resources available to Processity Data History to execute successfully
- Cascading Operations
Transactions that invoke further processing on multiple Salesforce objects consume more transaction resources, reducing the amount of free resources for Processity Data History to execute successfully.
- Processity Data History Tracking Mode
The configured Processity Data History Tracking Mode (see [Tracking Mode](#)) will impact transaction and governor limits characteristics in your organisation.
 - Synchronous
A Processity Data History configuration in synchronous mode will see governor limit utilisation commensurate with the level and complexity of automations on the Salesforce objects being tracked.
 - Queueable
A Processity Data History configuration in queueable mode will need to bear

¹⁵ See 'Salesforce Developers - Apex Reference Guide - DescribeFieldResult Class' for more information:

(https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_methods_system_field_s_describe.htm)

¹⁶ Roll-up summary fields are an exception and **are** trackable

¹⁷ See Salesforce Developers - Apex Governor Limits

(https://developer.salesforce.com/docs/atlas.en-us.salesforce_app_limits_cheatsheet.meta/salesforce_app_limits_cheatsheet/salesforce_app_limits_platform_apexgov.htm) for more information.

in mind the maximum number of asynchronous Apex method executions (batch Apex, future methods, Queueable Apex, and scheduled Apex) per a 24-hour period in their organisation, to which Processity Data History will contribute to in this tracking mode.

- Platform Event

A Processity Data History configuration in platform event mode will need to bear in mind the maximum event deliveries¹⁸ per 24-hour period which applies to their organisation, to which Processity Data History will contribute to in this tracking mode.

- Installed Managed Packages

Transaction processing introduced by other managed packages can impact transaction characteristics and the availability of transaction resources.

- Configured Salesforce Functionality

The enablement of Salesforce native functionality can impact transaction characteristics and the availability of transaction resources.

Data Management

Processity Data History features [Data History Data Snapshots](#) and [Data History Data Management](#) are scoped to exclusively process SE records stored on the Salesforce platform, specifically excluding SE records stored in Data History Data Store.

¹⁸ See 'Default Allocations' in Salesforce Developers - Default Platform Event Allocations (https://developer.salesforce.com/docs/atlas.en-us.platform_events.meta/platform_events/platform_events_limits_default.htm) for more information.

Known Issues

Managed Packages

If a managed package trigger runs after Processity Data History, then Processity Data History will still track the changes correctly for auditing purposes. However, the relationship between changes in Transaction Explorer may be incorrect. This can be addressed by using After Save Flow Triggers in addition to Apex triggers.

Rollbacks

Rollbacks within transactions are problematic. Processity Data History will have logs prepared, but will not be able to determine whether a rollback has happened so may log changes that did not, in fact, get committed.

Bulk Transactions

Although Processity Data History can provide lineage of how a bulk transaction leads to sub-transactions, Processity Data History is not able to do that at a per-record level. E.g., If inserting an Account leads to an insert of a Contact, and two Accounts are inserted together, then the child sub-transaction will show both Contacts being inserted. This is awkward, but since any code can run in a trigger to insert further records, Processity Data History is not able to determine that lineage per-record. It is important to note, however, that for interactive use, the initial transaction will mostly be on a single record.

End User Guide

This information is useful for those users that are working within environments where Processity Data History has been enabled.

Generating Span Events (SEs)

The generation of SEs occurs seamlessly in the background as Processity Data History Trace Writer or Processity Data History Admin users make in-scope updates to your Salesforce environment's records.

When will SEs be generated by the application?

SEs will be generated when the following criteria are met:

1. The Processity Data History Global Configuration is set to 'Active'.
2. The Processity Data History Object Configuration for a given Salesforce object is set to 'Active'.
3. The required Apex artifacts for a given Salesforce object have been implemented (see '[Object Triggers](#)').
4. The running user for an update for a record of a given Salesforce object is a Processity Data History Trace Writer or Processity Data History Tracking Administrator user (see '[Processity Data History Trace Writer](#)' & '[Processity Data History Tracking Administrator](#)').
5. The updates to a record of a given Salesforce object are on fields that are in scope for Processity Data History tracking (see '[Object Configuration](#)').

Viewing Span Events (SEs)

Users with sharing access¹⁹ to a given Salesforce record that are also Processity Data History Trace Viewer users can view SEs in the context of the traced Salesforce record they are related to.

When will SEs be viewable by an end user?

When attempting to view SEs on a given record of a Salesforce object, the following criteria are considered:

1. SE(s) for a given record must have been previously generated by a Processity Data History Trace Writer user.
2. The Audit Trails Lightning Web Component must be added to the assigned Lightning Record Page/Flexipage (see '[Lightning Record Page/Flexipage Component](#)') for a given record of a Salesforce object.

¹⁹ Salesforce Sharing Settings - https://help.salesforce.com/s/articleView?id=sf.managing_the_sharing_model.htm&type=5

3. The viewing user must be a Processity Data History Trace Viewer (see [Processity Data History Trace Viewer](#)) user or a Processity Data History Tracking Administrator (see [Processity Data History Tracking Administrator](#)) user.
4. The viewing user must have Field Level Security access to any given field that is captured as an event by a SE. For example, if a SE has captured an update to the Account field 'Description' for a given Account record, the viewing user **must** have Field Level Security read access to 'Account.Description' to view that SE (as well as sharing access to the given Account record). If the viewing user has FLS access to a subset of field updates captured in SEs, then only that subset of SEs will be displayed to the viewing user.
5. If the Audit Trails component has been configured with **Fields to Exclude** (see [Lightning Record Page/Flexipage Component](#) for more information) - does the user have the View Processity Data History Excluded Fields permission - if yes, all SEs viewable from steps 1-4 will be presented, if no, fields configured in **Fields to Exclude** will **not** be presented (resulting in a subset of SEs or no SEs being displayed, depending on the fields SEs are concerned with and the configured Fields to Exclude.)

Ways to View Span Events (SEs)

There are several common data points when viewing Processity Data History Span Events:

- Trace - a unique Id value to a group of SEs, indicating co-occurrence/relation of multiple SEs
- Date-Time - the date-time when the SE or group of SEs occurred
- User - the Salesforce user identified as making or initiating the in-scope changes captured by the SEs

By default, field changes captured by SEs will display the field labels for the fields in question - this can be changed to display field API names by setting the **Display Field API Names** toggle to *On*.



There are several different ways to view Processity Data History Span Events:

SE Trace View

Processity Data History User Type(s): *Processity Data History Trace Viewer* and *Processity Data History Tracking Administrator*

The Trace View is the default view provided by the Audit Trails component - as indicated by the **Trace View** toggle being set to *On* by default. This view aims to provide an easy-to-read summary of changes, allowing the end user to have an at-a-glance view of what changes to a record occurred.

 A screenshot of the Audit Trail component showing a table of traces. The table has columns: Trace, Date, User, and Summary. The 'Trace View' toggle is set to 'On'. One row is highlighted with a red box around the summary text.

Trace	Date	User	Summary
221ae41f-3f15-4210-b41f-db633a4daabd	10/06/2024, 04:10 pm	User User	Record updated with Account Number from "" to 62378239 and Website from "" to www.edge-comms.com and Industry from "" to Banking ... See More
645c98f9-cb63-4008-9014-f405bdf8314d	10/06/2024, 04:08 pm	User User	Record updated with Account Name from Edge Communications to Edge Communications PLC .

[View All](#)

This view groups one or more **co-occurring** SEs into this summary view so that the viewer can see the SEs that occurred in context with other SEs. If more than three SEs are part of a grouping, the summary view can be expanded by clicking **See More**:

 A screenshot of the 'Trace Summary' dialog box. It displays the trace ID, date, and user. Below, under 'Field Changes', a list of updates is shown, including Account Number, Website, Industry, Annual Revenue, Employees, Ticker Symbol, Account Rating, and Account Site. A 'Close' button is at the bottom right.

Trace Summary

Trace: TID:319030300000bc8bc4 Date: 10/06/2024, 04:10 pm User: User User

Field Changes

Record updated with **Account Number** from "" to **62378239**
 and **Website** from "" to www.edge-comms.com
 and **Industry** from "" to **Banking**
 and **Annual Revenue** from "" to **100000000**
 and **Employees** from "" to **10000**
 and **Ticker Symbol** from "" to **ECO**
 and **Account Rating** from "" to **Warm**
 and **Account Site** from "" to **New York, NY**

[Close](#)

SE Field-by-field View

Processity Data History User Type(s): *Processity Data History Trace Viewer* and *Processity Data History Tracking Administrator*

The Field-by-field view displays each individual SE as an individual row in the Audit Trails component. It is accessed by setting the **Trace View** toggle to the *Off* position.

In the Field-by-field view, each field change captured as a SE is presented, as follows:

- The date and time the captured SE occurred
- The field the SE concerns
- The user making or initiating the event
- The original and new field value captured by the event
- The Trace, representing SEs that are co-occurring/related

Audit Trail (9)						
				Display Field API Names ☐ Off	Trace View ☐ Off	
Date	Field	User	Original Value	New Value	Trace	
10/06/2024, 04:10 pm	Account Number	User User		62378239	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Website	User User		www.edge-comms.com	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Industry	User User		Banking	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Annual Revenue	User User		100000000	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Employees	User User		10000	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Ticker Symbol	User User		ECO	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Account Rating	User User		Warm	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:10 pm	Account Site	User User		New York, NY	221ae41f-3f15-4210-b41f-db633a4daabd	
10/06/2024, 04:08 pm	Account Name	User User	Edge Communications	Edge Communications PLC.	645c98f9-cb63-4008-9014-f405bdf8314d	

SE View All

Processity Data History User Type(s): *Processity Data History Trace Viewer* and *Processity Data History Tracking Administrator*

From the Audit Trails component you can view all SEs related to a given traced record in an expanded view by clicking on **Audit Trail (n)** in the component header.

Lead

Mr. Jack Samson

Title

CEO and Founder

Company

SAMSON & CO.

Phone (2)

18008237653

Email

jack@samsonco.com

Working - Contacted

Closed - Not Converted

Audit Trail (5+)

Display Field API Names

off

Group By Transaction

on

Trace	Date	User	Summary
cd4c3289-a95f-44f5-8481-e2046361a6ff	17/03/2025, 02:33 pm	User User	Record updated with Salutation from "" to Mr. and Website from "" to samsonco.com and Annual Revenue from 100000000 to 150000000
14b812bf-ee23-4f87-8a72-b6586b311c42	17/03/2025, 02:33 pm	User User	Record updated with Phone from "" to 18008237653
1e9f63af-e313-4dd9-8397-67fa8111e726	17/03/2025, 02:33 pm	User User	Record updated with Email from "" to jack@samsonco.com and Status from Open - Not Contacted to Working - Contacted
4d9a7a0a-0f46-4bd0-a5f3-c72a618064d0	17/03/2025, 02:32 pm	User User	Record updated with Rating from "" to Hot and Employees from "" to 100000
5dc87e83-e0d2-4d07-835b-0e10008ae930	17/03/2025, 02:32 pm	User User	Record updated with Industry from "" to Electronics and Annual Revenue from "" to 100000000

View All

Check Archive

Load More

The View All mode is then presented:

Audit Trail (7)			
		Display Field API Names	Group By Transaction
		Off	On
Trace	Date	User	Summary
cd4c3289-a95f-44f5-8481-c2046361a8ff	17/03/2025, 02:33 pm	User User	Record updated with Salutation from "" to Mr. and Website from "" to samsone.com and Annual Revenue from 100000000 to 150000000
14b812bf-ec23-4f87-8a72-b6586b311c42	17/03/2025, 02:33 pm	User User	Record updated with Phone from "" to 18006237653
1e9f53af-e313-4d09-8397-6768116728	17/03/2025, 02:33 pm	User User	Record updated with Email from "" to jack@samsone.com and Status from Open - Not Contacted to Working - Contacted
4d9a76da-0f45-4bd0-a5f3-c72a618094d0	17/03/2025, 02:32 pm	User User	Record updated with Rating from "" to Hot and Employees from "" to 100000
5dc87e83-e0d2-4d07-835b-0e10008ae930	17/03/2025, 02:32 pm	User User	Record updated with Industry from "" to Electronics and Annual Revenue from "" to 100000000
f0f0d9a-7b88-4114-92b1-df2765778c1a	17/03/2025, 02:32 pm	User User	Record updated with Lead Source from "" to Web
23003d6f-b341-4b48-bf0-8113a8f8e55	17/03/2025, 02:29 pm	User User	Record created with Unread By Owner as true and Last Name as Samson and First Name as Jack ... See More

In this View All mode - the SEs can be viewed in both the [SE Trace View](#) and the [SE Field-by-field View](#) described above.

Trace Explorer

Processity Data History User Type(s): *Processity Data History Trace Viewer* and *Processity Data History Tracking Administrator*

The Trace Explorer view provides an additional mechanism to inspect the larger context that any given SE or group of SEs has, revealing potential interrelations of events between records of one object, with events captured on other object types.

The Trace Explorer view can be accessed by clicking on a **Trace** link from the Audit Trails component on a Lightning Record Page, from either the SE Trace View or the SE Field-by-field View:

Related			
		Display Field API Names	Group By Transaction
		Off	On
Trace	Date	User	Summary
a7c3690e-a8e4-4104-9052-dbad2a...	28/02/2025, 04:16 pm	User User	Record created with Account ID as Vizcool Corp and Private as false and Name as Vizcool Corp 1000 Widgets ... See More
View All			

On opening the 'Trace Explorer':

Audacity Trace Details

Trace Summary

Trace Initiated By: User User
 Trace Duration: 429 ms
 Trace Operation: Database Write
 Trace Quiddity: Synchronous

Trace Id: a7c3690e-a8e4-4104-9052-dbad2aa33e79
 Trace Start: 28/02/2025, 04:16:24 pm
 Trace End: 28/02/2025, 04:16:24 pm

Trace Explorer Trace Chart

Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Opportunity Vizool Corp 1000 Widgets - Apex Before Insert (11)
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Account Vizool Corp - Apex Before Update
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Opportunity Vizool Corp 1000 Widgets - Apex After Insert
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Opportunity Vizool Corp 1000 Widgets - Flow After Insert
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Account Vizool Corp - Apex After Update
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Account Vizool Corp - Flow After Update

CPU Time: 56 ms | Apex Heap Size: 12,831 bytes | DML Statements: 0 | SOQL Queries: 0 | queueableJobs: 0 | dmlRows: 0 | queryRows: 0

Related Audit Trails (11) Display Field API Names: ☐ off

Entity	Date	Field	User	Original Value	New Value
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Account ID	User User		Vizool Corp
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Private	User User		false
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Name	User User		Vizool Corp 1000 Widgets
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Stage	User User		Prospecting
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Amount	User User		20000.00
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Probability (%)	User User		10.0
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Close Date	User User		2025-04-30 00:00:00
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Opportunity Type	User User		New Customer
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Lead Source	User User		Other
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Forecast Category	User User		Pipeline
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Owner ID	User User		User User

Once the Trace Explorer has been opened, summary information on the group of SEs is provided at the top of the screen:

- **Trace Id** representing the unique Id for this grouping of SEs
- **Trace Start** the date-time for the start of the events in this grouping of SEs
- **Trace End** the date-time for the end of the events in this grouping of SEs
- **Trace Events Tracked** represents the number of SE sub-groups that are correlated to this overall grouping
- **Trace Operation Types** representing the distinct operation types that were recorded when capturing SEs for this overall grouping
- **Trace Quiddities** representing the different request types²⁰ recorded when capturing SEs for this overall grouping.

On the lower-left side of the Trace Explorer are the SE sub-groups that were captured as part of this overall grouping. Clicking onto each sub-group reveals the underlying SEs that are part of this sub-group, on the lower-right side of the view - this view of SEs matches the presentation of the [SE Field-by-field View](#):

Trace Explorer Trace Chart

Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Opportunity Vizool Corp 1000 Widgets - Apex Before Insert (11)
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Account Vizool Corp - Apex Before Update
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Opportunity Vizool Corp 1000 Widgets - Apex After Insert
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Opportunity Vizool Corp 1000 Widgets - Flow After Insert
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Account Vizool Corp - Apex After Update
 Database Write - Synchronous (Transaction TID:2891130900009595f5d)
 Account Vizool Corp - Flow After Update

CPU Time: 56 ms | Apex Heap Size: 12,831 bytes | DML Statements: 0 | SOQL Queries: 0 | queueableJobs: 0 | dmlRows: 0 | queryRows: 0

Related Audit Trails (11) Display Field API Names: ☐ off

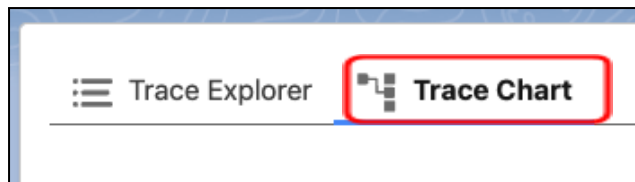
Entity	Date	Field	User	Original Value	New Value
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Account ID	User User		Vizool Corp
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Private	User User		false
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Name	User User		Vizool Corp 1000 Widgets
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Stage	User User		Prospecting
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Amount	User User		20000.00
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Probability (%)	User User		10.0
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Close Date	User User		2025-04-30 00:00:00
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Opportunity Type	User User		New Customer
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Lead Source	User User		Other
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Forecast Category	User User		Pipeline
Vizool Corp 1000 Widgets	28/02/2025, 04:16:24 pm	Owner ID	User User		User User

Processity Data History Trace Chart

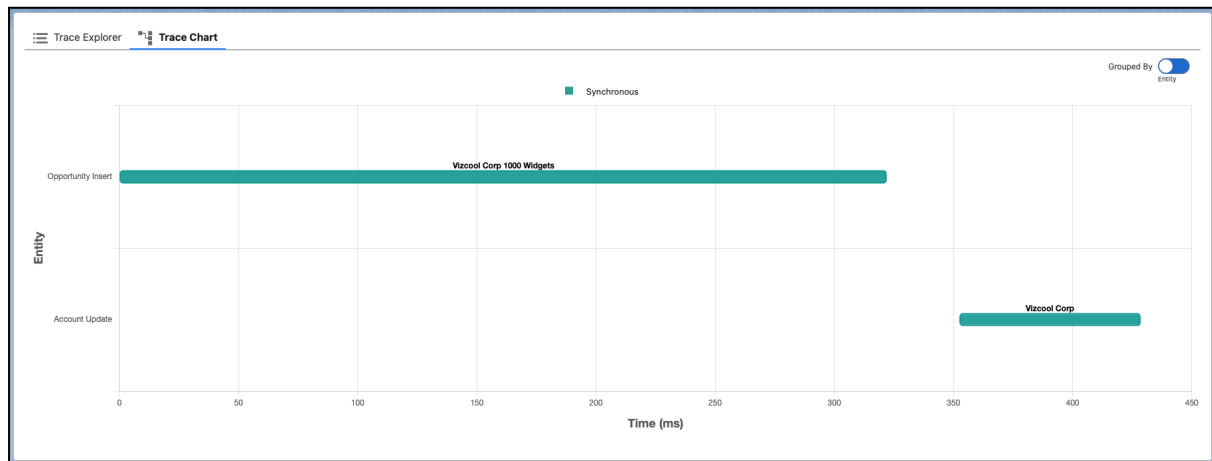
Next to the **Trace Explorer** tab you can find the **Trace Chart**. This is a visual representation of different phases of a single transaction trace.

²⁰ Salesforce Apex Reference Guide - Quiddity Enum

https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_enum_System_Quiddity.htm

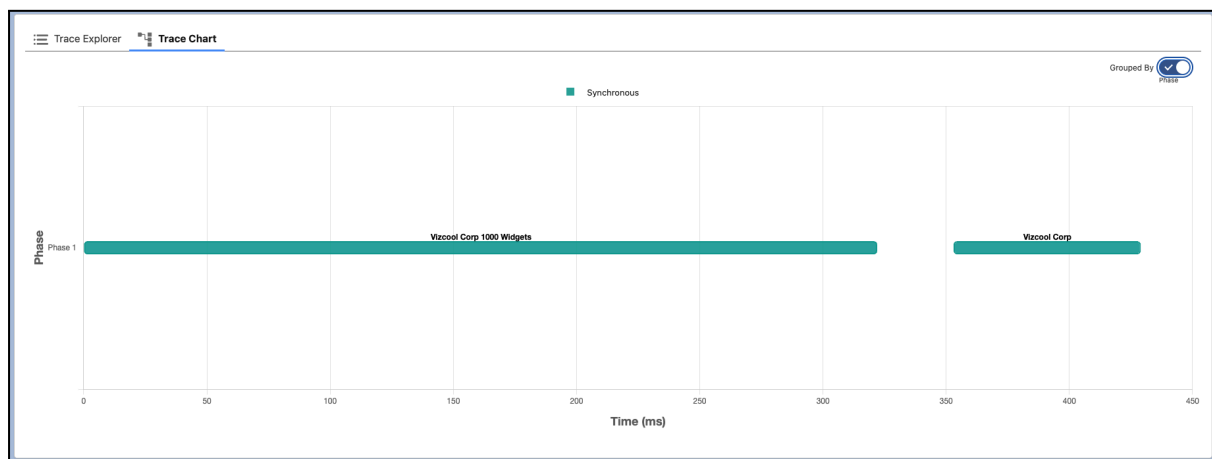


When you click **Trace Chart** you'll see the same captured changes presented in a waterfall chart.

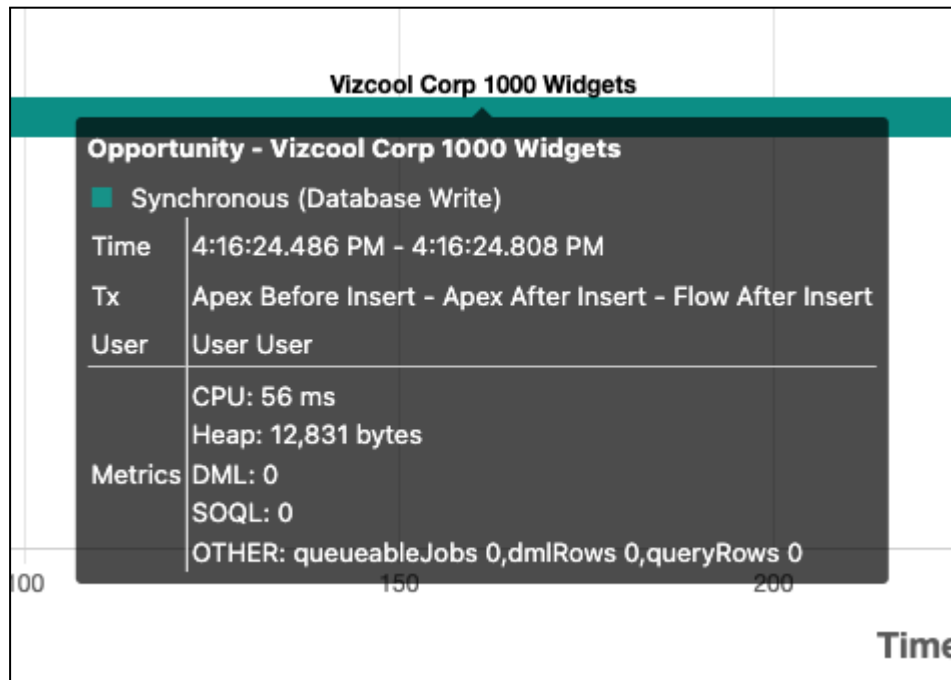


By default the chart is presented grouped by entity, meaning each individual Salesforce object will appear in its own bar in the graph.

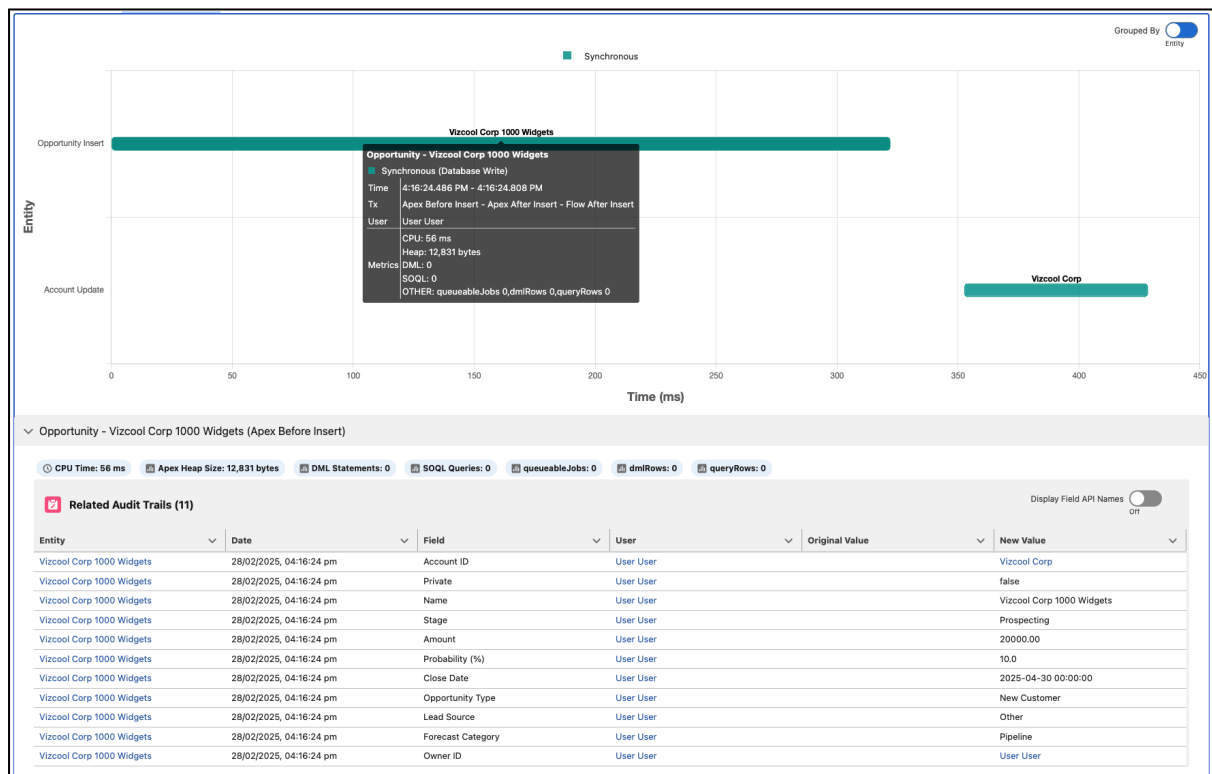
You can also visualize by transaction phase, meaning you can see whether a change took place during a trigger, asynchronous Apex, etc. To view by phase, toggle the **Grouped By** button.



Placing your mouse over any of the bars allows you to inspect the performance metrics of any segment of a transaction.



Clicking on any bar will identify which changes took place within that part of the transaction.



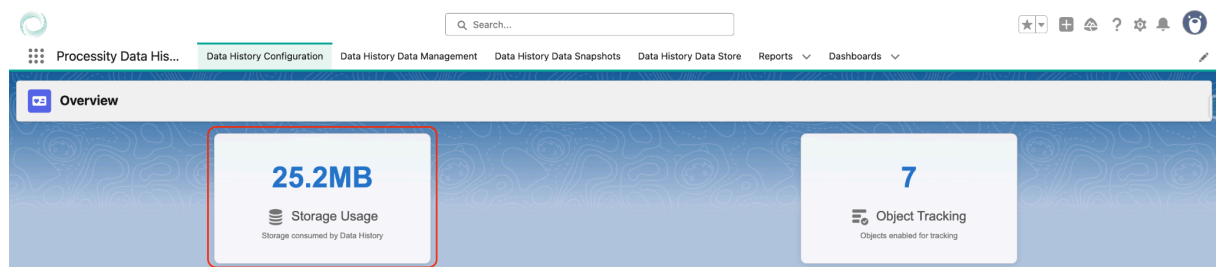
Viewing Processity Data History Data Consumption

The generation of SEs consumes Salesforce Data Storage²¹ which should be monitored by Processity Data History Tracking Administrators and/or your org's Salesforce Administrators. There are two ways to view the current data consumption of SEs:

1. Data History Configuration

Processity Data History User Type(s): *Processity Data History Tracking Administrator* users only

The Data History Configuration page provides the approximate absolute data consumption in Gb/Mb for SEs in your Salesforce environment:



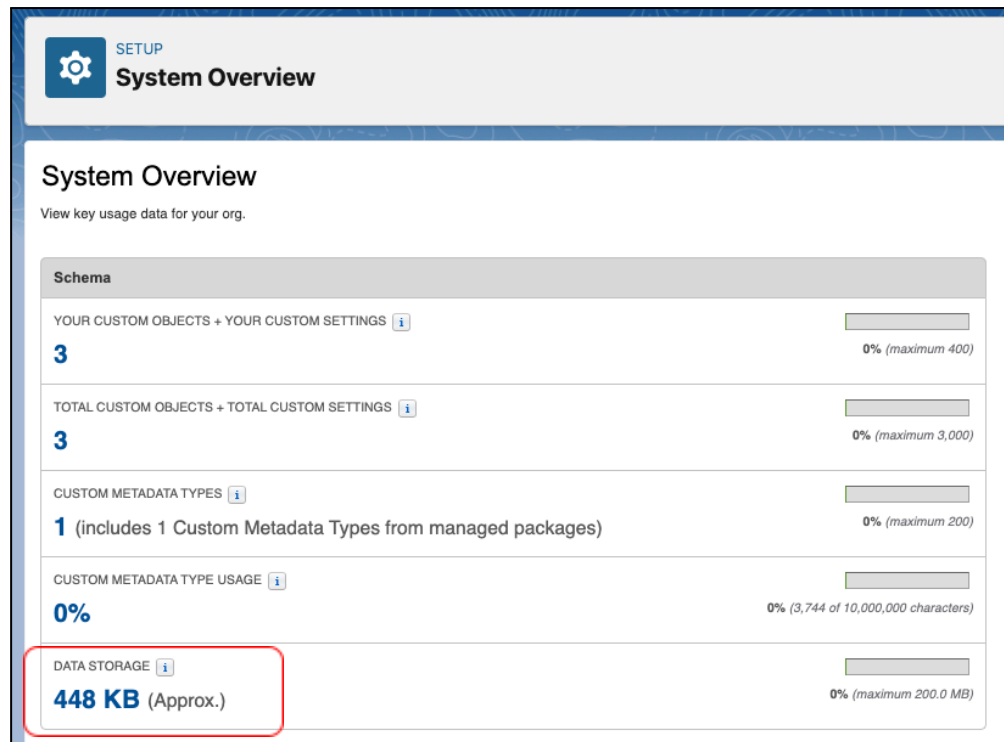
2. Setup/System Overview

User Type(s): Any Salesforce Administrator

Salesforce Setup provides information on absolute and relative consumption of data, as well as your environments total allowances. You can access this information as follows:

- a. Click the cog  in the top right of the browser window, click **Setup**
- b. Using the navigation menu on the left, navigate to **Environments > System Overview**:
- c. Once you have opened System Overview, click the linked data storage usage number, under **Data Storage**:

²¹ See Salesforce Help for more information on this concept:
https://help.salesforce.com/s/articleView?id=sf.overview_storage.htm&type=5



- d. Under the section **Current Data Storage Usage** find the row for Span Events - this provides you with the total SE record count, storage in KB consumed, and the relative consumption:
- e. The total consumption of your Salesforce environment can also be monitored from this page, under the **Storage Type** section:

Data History Data Snapshots

Processity Data History Span Events (SEs) records are created in a data format that is optimized for data storage efficiency (i.e., to minimise the impact SE data storage has within your Salesforce environment). A single SE record can contain multiple record histories (i.e., multiple field changes - record histories - on a given tracked object can be recorded in a single SE record).

This storage optimization makes it difficult for a Processity Data History subscriber to meaningfully report on and/or introspect SE data, natively in Salesforce - which is unable to make use of structured data inside records.

One possible solution would be to export SE data into Business Intelligence or other tools. Such tools can read the data inside records during the reporting process, or pre-process it to expand into more records with simpler internal structure.

Data History Data Snapshots provides a capability to generate expanded SE data, natively in Salesforce. This allows subscribers to report on and/or introspect SE data, without having to export it to external tools.

Data Snapshots

Data Snapshots allow you to export Processivity Data History data into your own custom object records. This enables reporting with standard Salesforce reports. You can, for example, export the audit trails related to all Account records into custom object records with a lookup field to Account. Then, a standard report on Account can include your custom object.

It is your responsibility to ensure the security of any data that you export in this way. The data will be restricted to what you are able to view with your user. But if other users have access to the custom object target of a Data Snapshot, then those users will be able to see the data even if they lack access to the fields as they were originally tracked.

Take the following steps to create a Data Snapshot

1. If you do not already have a custom object defined to receive the data, then create one.
2. Define a Filter to choose which audit trail records to include. You can combine filters using AND/OR operators.
3. (Optional) If you want to restrict the fields included in the snapshot, then fill in a comma-separated list of fields to include in the format of Object.Field e.g. to include Contact First Name and Contact Last Name, write Contact.FirstName, Contact.LastName.
4. Map Processivity Data History data fields to fields on your target object.
5. (Optional) Select "Dry run" to make sure that the snapshot will work and to find out how many records it will create.
6. Start the snapshot. You will be taken to an Async Job record that will be updated as the snapshot runs, showing how many records have been created (or would be created if this is a dry run).

Filter Records

Add Filter

Select

*Field

Select a Field

*Comparison Operator

Select an Operator

Value

Data History Data Snapshot Process

The Processivity Data History Data Snapshot process consists of several sequential steps that can be undertaken by Salesforce Administrators and users with Processivity Data History Tracking Administrator permissions in collaboration:

Define or identify a target custom object (Salesforce Administrators)

The subscriber is responsible for defining²² or identifying a Custom Object in their Salesforce environment that can receive the normalised Span Event (SE) record histories.

For example, they might define an object called "Field History" that contains fields similar to the standard history objects (Account History, Contact History, etc.).

They may also wish to include some fields for the additional information that Processivity Data History captures, e.g. Trace UUID.

The scope of which fields to include will depend on the intended use.

Secure the target custom object (Salesforce Administrators)

The subscriber is responsible for ensuring the security configuration of the identified target Custom Object (in terms of its Sharing Settings²³ and Field-level Security²⁴) at all times - it

²² For guidance on creating Custom Objects in Salesforce, see 'Salesforce Help & Training - Create a Custom Object'

(https://help.salesforce.com/s/articleView?id=sf.dev_objectcreate_task_parent.htm&type=5).

²³ See 'Salesforce Help & Training - Sharing Settings'

(https://help.salesforce.com/s/articleView?id=sf.managing_the_sharing_model.htm&type=5) for further information.

²⁴ See 'Salesforce Help & Training - Configure Field-Level Security for Users'

(https://help.salesforce.com/s/articleView?id=sf.emergency_response_service_fls.htm&type=5) for further information.

should be adequate to protect the data produced by this process and be in accordance with the subscriber organisation's security and data management policies.

Choose which SE records will be processed (Processity Data History Tracking Administrator users)

Navigate to the Processity Data History application tab **Data History Data Snapshots** (Processity Data History Tracking Administrator users)

Subscribers that are familiar with the Salesforce Object Query Language (SOQL) will find creating this selection expression familiar. Using the available source data fields of the SE records, logical selection criteria for SEs can be specified.

1. Select a Field - select a SE field (E.g., Entity Type)
2. Select a comparison operator (E.g., =, !=, >)
3. Select a value
 - where valid values are a bound list, input is presented as a drop-down with options
 - where a value should be expressed as a date-time value, input is presented as a date and time selector
 - in all other cases, free text input is presented

In the example below, SE records that are concerned with the Account object are being selected.

The screenshot shows a dialog box titled "Select Span Events". It contains four input fields: "Logical Operator" with a dropdown menu showing "Select an O...", "Field" with a dropdown menu showing "Entity Type", "Comparison Operator" with a dropdown menu showing "=", and "Value" with a dropdown menu showing "Account".

4. Logical conditions expressed as AND or OR can be selected from the **Logical Operator** picklist.

The screenshot shows the same "Select Span Events" dialog box, but the "Logical Operator" dropdown menu is open, displaying two options: "AND" and "OR". The other fields remain the same as in the previous screenshot.

Configure mapping to target object (Processity Data History Tracking Administrator users)

Select the Custom Object identified or defined in step 1, from a drop-down with options of all Custom Objects available in your environment. Note: only Custom Objects are supported as target objects for Data History Data Snapshots. Standard Objects are not supported.

Configure the field mapping (Processity Data History Tracking Administrator users)

In the left column, select one of the available source fields from the SE object.

In the right column, select a compatible field on the target Custom Object.

You can map as many or as few fields as you require. Note: if the selected Custom Object has required fields, you must configure the Data Snapshot mapping to populate these fields in all cases to produce a full set of results.

[OPTIONAL] Dry Run (Processity Data History Tracking Administrator users)

Select the Dry Run option to test whether this data snapshot configuration executes successfully. Note: **no** Custom Object records will be created with this option selected.

You will be able to see whether the Data Snapshot would have been successful, and how many records it would create. This can be useful to make sure that you do not exceed your storage allowance in Salesforce by expanding a very large number of records.

Click Start Snapshot (Processity Data History Tracking Administrator users)

This will initiate the configured Data Snapshot, or a dry run, and will navigate to the produced Async Job record that can be used to track the progress of the job's execution.

Completion (Processity Data History Tracking Administrator users)

Once **Total Executions** and **Successful Executions** match on the Async Job, the job has successfully executed (either as an actual execution or as a dry run).

Custom Object records - the normalised SE data (any users with the requisite security and access permissions as configured in step 2)

For Data Snapshots that were initiated with Dry Run **deselected**, the job will have populated the selected Custom Object records according to the logical criteria and field mapping defined in steps 4-7.

The requested SE record data is now available for reporting²⁵ or introspection, by referencing the selected Custom Object.²⁶

If you have a requirement to schedule Data Snapshots to run repeatedly, or on a schedule, please reach out to support@processity.ai who can help you with this requirement.

²⁵ For guidance on creating Salesforce custom reports, see 'Salesforce Help & Training - Create a Custom Report Type'

(https://help.salesforce.com/s/articleView?id=sf.reports_defining_report_types.htm&type=5)

²⁶ If existing records for the Custom Object exist, additional filtering logic will need to be applied to ensure the correct set or subset of Custom Object records are included in the reporting or introspection process.

Data History Performance Monitoring

Processity Data History captures performance data about your org's transactions, and either in concert with Salesforce's Scale Center²⁷, or on its own, helps you monitor your org's performance versus Governor Limits, natively in Salesforce reports and Dashboards.

By default Processity Data History will capture performance data on Apex governor limit consumption for 'CPU Time' and 'Heap'. These limits are measured across an entire transaction and are *not* counted separately for certified managed packages. Additional limits, where limits *are* counted separately for certified managed packages, can be tracked by implementing a small class inside your org., and updating your Processity Data History configuration. This allows the following limits, that cannot be tracked inside Processity Data History's namespace context, to also be tracked in your org.:

Additional supported tracking metrics:

- DML
The number of DML statements (such as insert, update or the database.EmptyRecycleBin method) that have been called.
- SOQL
The number of SOQL queries that have been issued.

See 'Implementing additional performance tracking outside of the Processity Data History namespace' for more information on implementing additional performance tracking in your org.

Packaged reports can be found in the 'Processity Data History Reports' report folder:

Report	Purpose
Data History - Limit Consumption >=90%	Returns any limit utilization greater than or equal to 90% of limit consumption. This can be particularly useful for creating a performance dashboard.
Data History - [LIMIT], Limit Spread	Returns events for a given limit, bucketed by low/medium/high limit consumption, for a given transaction context.
Data History - [LIMIT], Distribution	Returns events for a given limit with their relative distribution across limit utilization.

²⁷ See 'Salesforce Help - Scale Center' (https://help.salesforce.com/s/articleView?id=xcloud.scale_center_overview.htm&type=5) for more information.

Performance questions that Processity Data History can help you answer

Limit Consumption Proximity - are my org's transactions close to Apex governor limits?

For each of the limits tracked, displays which org. transactions are consuming ($\geq 90\%$) of the respective limit. This can help you identify high level performance hotspots across your transactions, and helps you focus on transactions that have the least margin and could cause limit exceptions in the near-term.

Report: Data History - Limit Consumption $\geq 90\%$

Limit Consumption Spread - how are my org's transactions limit consumption spread across high-level low/medium/high buckets?

For each of the limits tracked, how is limit consumption spread between low/medium/high buckets of limit utilization. This can help you understand your available performance margins for the various objects you are tracking, and where optimization efforts might be prioritized.

Processity Data History Reports - Limit Consumption Spread

Limit	Report
CPU	Data History - CPU Time, Limit Spread
Heap	Data History - Heap, Limit Spread
DML	Data History - DML, Limit Spread
SOQL	Data History - SOQL, Limit Spread

Limit Consumption Distribution - how are my org's transactions for a given object distributed across detailed limit utilization percentages?

For each of the limits tracked, how are limits consumed as a percentage overall observations. This can help you understand your available performance margins for the various objects you are tracking, and the detailed proportion of your object transactions that are at a given level of limit utilization.

Processity Data History Reports - Limit Distribution

Limit	Report
CPU	Data History - CPU Time, Distribution

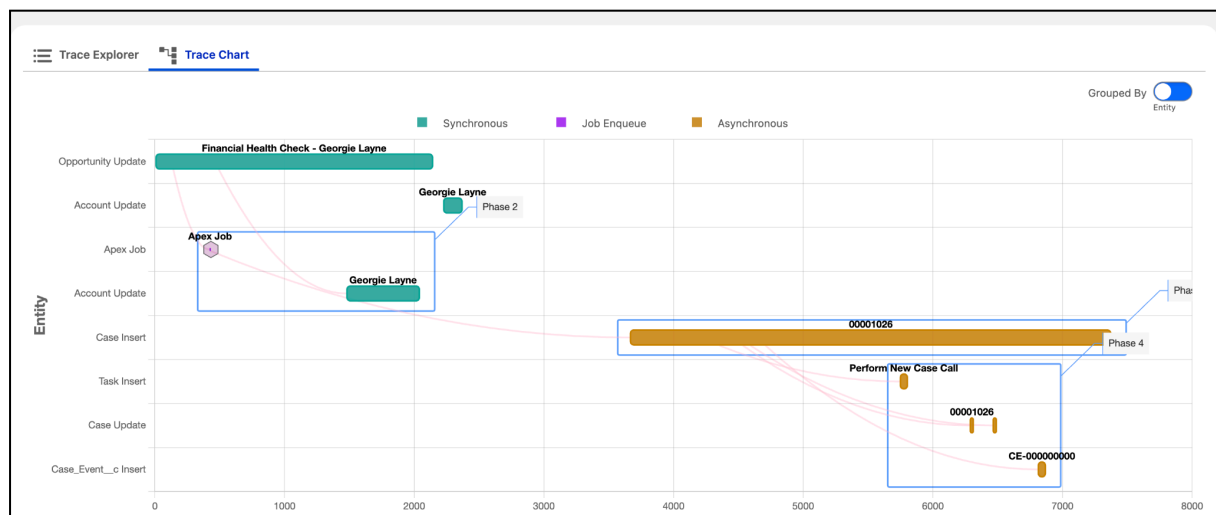
Heap	Data History - Heap, Distribution
DML	Data History - DML, Distribution
SOQL	Data History - SOQL, Distribution

Investigating specific record transaction performance - how can I investigate a Scale Center report for a specific record or a set of records?

You can investigate a Scale Center report by choosing one of the packaged reports and then filtering the chosen report further to a specific record Id or a subset of record Ids, using the Span Event field 'Entity Id' as the basis of the filtering. We recommend modifying the relevant supplied Processivity Data History report and then using 'Save As' to avoid changing the packaged report. This can help you respond to Scale Center warnings armed with additional information. You can optionally view the record itself via the 'Entity' field.

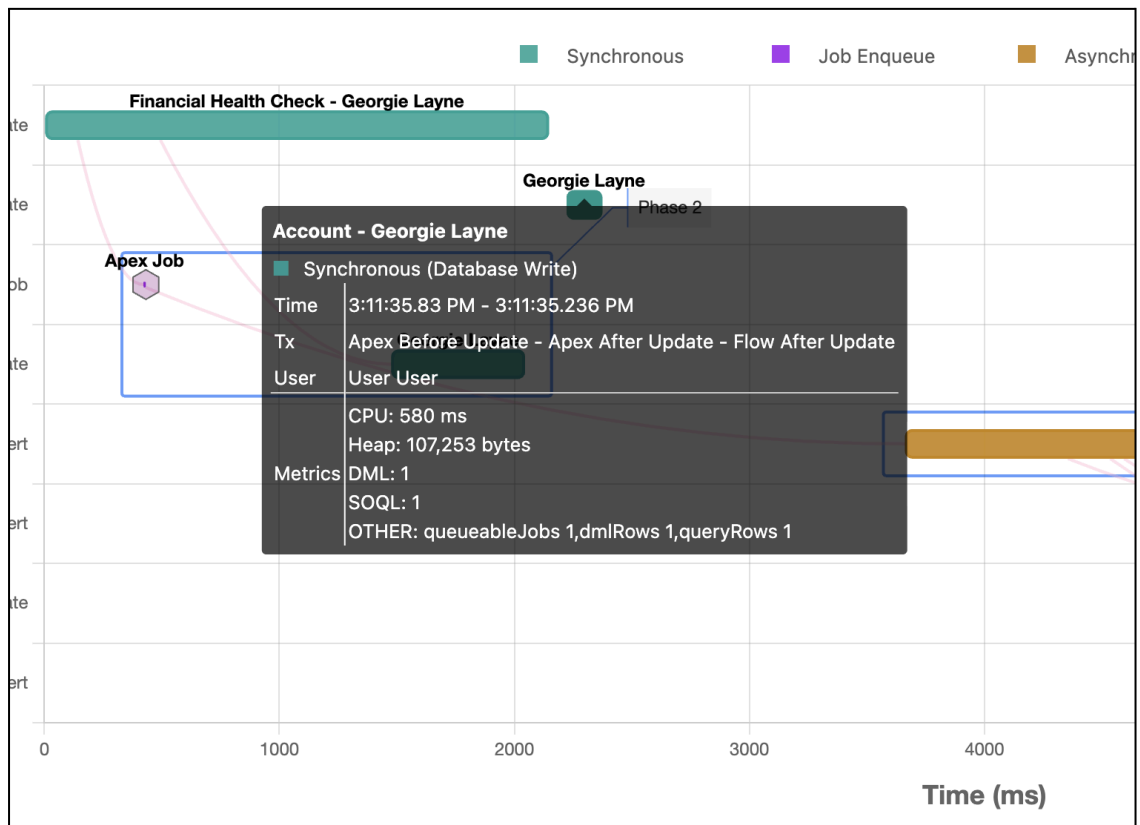
Investigating wider context of transaction performance - how can I investigate a Scale Center report to look at the wider context of the report?

You can also investigate the wider trace context of record reports from Scale Center. First, identify a Span Event record in the relevant performance report that matches the Scale Center record Id. Be careful to match not only the record Id, but also correlate the 'Time of Error' from Scale Center to the 'Timestamp Date/Time' field on the Span Event record. Once you have the correct row, click the 'Trace' field for this row. This will then open the respective Trace Details for the identified trace, and the wider transaction context can be further explored, including the Trace Chart:



You can then further inspect each element of the trace to review their performance

characteristics.



Implementing additional performance tracking outside of the Processity Data History namespace

You can implement additional performance tracking inside your org., by following two steps, outlined below. This enables the tracking of performance measures that are otherwise limited on a namespace-by-namespace basis, and provides limit performance tracking beyond 'CPU Time' and 'Heap', which are supported by default.

Enablement Steps

1. Implement a metrics instrument class.

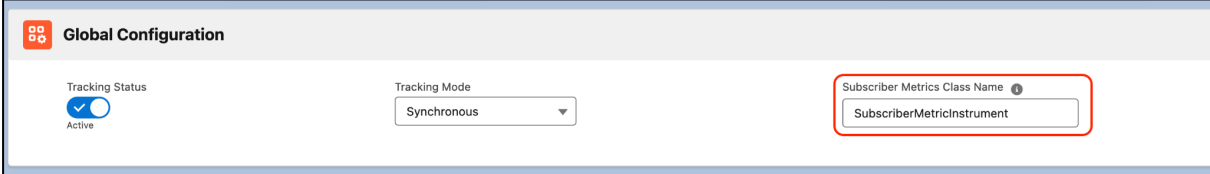
Using a tool like Developer Console or VS Code, in your org.:

None

```
public class SubscriberMetricInstrument implements Callable {
    public Map<String, Integer> call(String param1, Map<String,
Object> param2) {
        return new Map<String, Integer>{
            'queries' => Limits.getQueries(),
            'queryRows' => Limits.getQueryRows(),
            'dmlStatements' => Limits.getDmlStatements(),
            'dmlRows' => Limits.getDmlRows(),
            'queueableJobs' => Limits.getQueueableJobs()
        };
    }
}
```

2. Update your Processity Data History configuration with the name of the class.

Using the snippet provided above as an example you would add 'SubscriberMetricInstrument' to the 'Subscriber Metrics Class Name' field under Processity Data History's 'Global Configuration'. Finally, click 'Save'.



The screenshot shows the 'Global Configuration' window. It contains three main settings:

- Tracking Status:** A toggle switch labeled 'Active' that is currently turned on.
- Tracking Mode:** A dropdown menu currently set to 'Synchronous'.
- Subscriber Metrics Class Name:** A text input field containing the value 'SubscriberMetricInstrument', which is highlighted with a red rectangular box.

Processity Data History Performance Reporting Data Model

Processity Data History's performance reporting introduces a number of convenience fields to the Span Event object (mantra__SpanEvent__c) to make performance reporting easier and more convenient. FLS read is granted for all of these fields to the 'Processity Data History Tracking Administrator' (AudicityLoggingAdministrator) Permission Set Group. You can of course make grants to these fields for any of your custom Permission Sets as your user access policies and requirements demand.

Field Label	Field Name	Field Type	Purpose	Permission Set (API Name)
Entity	mantra__Entity__c	Formula (Text)	A convenience field providing a hyperlink to the entity that the Span Event concerns.	View Processity Data History Span Event Limits Fields (ViewProcessityDataHistorySpanEventLimitsFields)
Trace	mantra__Trace__c	Formula (Text)	A convenience field providing a hyperlink to the trace that contains this Span Event - allows you to explore the performance of a transaction and related events.	
CPU Consumption	mantra__CpuConsumption__c	Formula (Percent)	Calculates the CPU Time percentage consumption in a given transaction context.	
Heap Consumption	mantra__HeapConsumption__c	Formula (Percent)	Calculates the Heap percentage consumption in a given transaction context.	
DML Consumption	mantra__DmlConsumption__c	Formula (Percent)	Calculates the DML statement percentage consumption.	
SOQL Consumption	mantra__SoqlConsumption__c	Formula (Percent)	Calculates the SOQL statement percentage consumption in a given transaction context.	

Considerations for Performance Reporting with Processity Data History

Data Volumes

The performance reports packaged with the Processity Data History are scoped, by default, to manage large data volumes, by restricting the returned Span Events to those occurring in the last 7 days. Your reporting needs may require you to filter the returned results by specific date ranges, or your data volumes may afford you larger date ranges before experiencing report timeouts. Experimenting and iterating with date ranges in the reports is recommended to give you the best balance between reporting completeness and reporting performance.

Supported Storage

Processity Data History supports a number of storage mechanisms for Span Event data that the application produces whilst tracking is active. Performance reporting fields are only available on Span Events stored in the default Custom Object storage. Span Events stored in any Data Store solution (such as, Big Objects) are not currently supported for performance reporting natively in Salesforce reports.

Apex Governor Limit Values

As part of the performance reporting functionality, Processity Data History includes a number of convenience fields that calculate where a given execution is, in terms of a given limit, relative to the absolute limit value, for a given transaction context. These fields use limit values as specified by Salesforce. It's important to note that these convenience fields use the published limit values at the time of their implementation, and as such there may be some variance between the values we use and the latest values published by Salesforce. We will endeavour to keep the values Processity Data History uses, and the published values as defined by Salesforce, in alignment, but we cannot guarantee this at all times. The limit values are not stored in a 'point-in-time' or 'dated' manner, as they will reflect the latest values as specified by Salesforce. This may make the evaluation of limit consumption % for Span Events created before the limit value amendment(s), change over time.

For convenience, at the time of publication the current key Salesforce Apex Governor Limits are as follows²⁸

²⁸ See 'Salesforce Help – Apex Governor Limits' (https://developer.salesforce.com/docs/atlas.en-us.salesforce_app_limits_cheatsheet.meta/salesforce_app_limits_cheatsheet/salesforce_app_limits_platform_apexgov.htm) for the latest values and guidance.

Limit per Transaction	Transaction Context	
	Synchronous	Asynchronous
CPU Time	10,000 ms	60,000 ms
Heap Size	6 MB	12 MB
DML Statements	150	
DML Rows	10,000	
SOQL Queries	100	200
Query Rows	50,000	
Queueable Jobs	50	1

Get Started with Processity Data History

Not ready to try in your own org? Get to know Processity Data History in a safe environment, we recommend trying the Dreamhouse Processity Data History sample application [tutorial](#). In this tutorial you will create a scratch org, install a fork of the Dreamhouse sample application and install and configure Processity Data History. From that point you can begin to experience how Processity Data History does transaction tracing and historical data capture.

Processity Data History for Process Mining

We built Processity Data History to be the best historical data capture tool in the Salesforce ecosystem. We also built Processity, an on-platform business process mining tool for Salesforce. If your goal is to better understand, plan, and optimize business processes, be sure to also try Processity with Processity Data History.

— END —