

“roll-it-up” Roll-Up Summary Helper Configuration and User Guide

Managed Package Guide Version R1.5

July 31, 2020

Roll-It Up Guide

ROLL-IT UP GUIDE	2
OVERVIEW	3
INSTALLATION	4
CONFIGURING THE APPLICATION	5
Batch Roll-Up Summaries	5
Synchronous (real-time) roll-up summaries	9
Asynchronous (near real-time) roll-up summaries	10
Roll-up summary types	10
Using filters in roll-up summaries	11
CONFIGURING YOUR FIRST ROLL-UP SUMMARY	12
Roll-Up Settings	13
Roll-Up Batch	16
More Examples: create a synchronous (real-time) roll-up summary	16
More Examples: Create an asynchronous (near real-time) roll-up summary for standard objects	17
Error Log	19
Email Notification	19
Errors	22
User Guides	23
ADVANCED ‘ROLL-IT-UP’ SUMMARY SETUP	23
Creating an Apex Trigger	23
Apex Trigger example	24
Testing Apex Triggers in a synchronous (real-time) or asynchronous (near real-time) roll-up summary	25

Overview

'roll-it-up' lets you roll-up any Salesforce data in real time, near real time or batch that is part of a lookup or master-detail relationship. Your roll-up summaries are no longer constrained by master-detail relationships, or only four possible calculations (count, sum, min, max).

Natively, Salesforce only allows for roll-up summaries on master-detail relationships and there are limits. *'roll-it-up'* works on any standard or custom objects where a lookup or master-detail relationship is in place. *'roll-it-up'* supports calculations for count, count distinct, sum, average, min and max.

Features include:

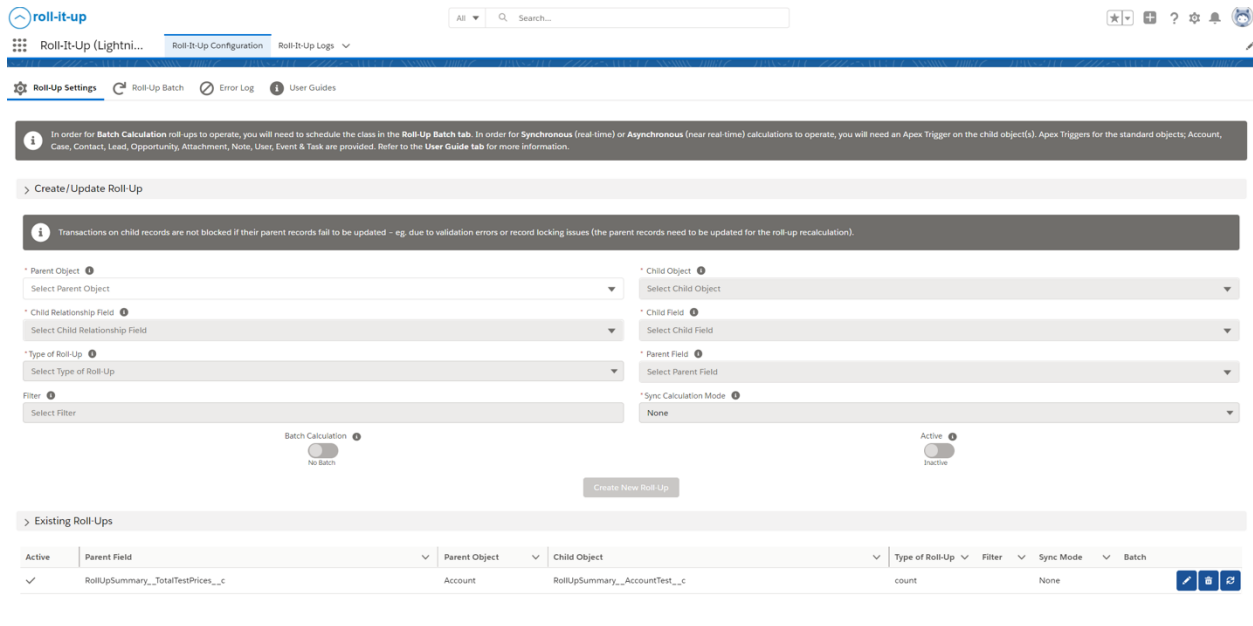
- Easy to install and configure – be up and running in minutes.
- Simple to configure through a rich and functional user interface.
- Real time, near real time and batch calculations are supported.
- Define rich filter criteria for calculations including the use of SOQL logic.
- Works with formula fields (where data is not subject to changes picked up in Apex Triggers).

'roll-it-up' lets you surface summary data for your lookup and master-detail relationships, giving you actionable, real-time summary data for use in reports, workflow and any of your Salesforce enabled processes. Such calculations are simply not possible using native roll-up summary features in Salesforce without high risk and expensive coded solutions. This powerful utility lets your Admins, Analysts, and Developers execute work faster and improve time to value by taking care of the heavy lifting for you.

New enhancements in this release:

This app has been updated in 2020 to configure these new features:

- Lightning (LEX) compatible interface
- List View of created roll-ups with quick link to edit or delete
- Easy error-handling via out-of-box Workflow rule/Email Alert that fires up when rollups have errors
- More explanations of synchronous v. asynchronous rollups within App
- User Guide and Instructions available from within the App
- Updated and improved User Guide with more screenshots



Installation

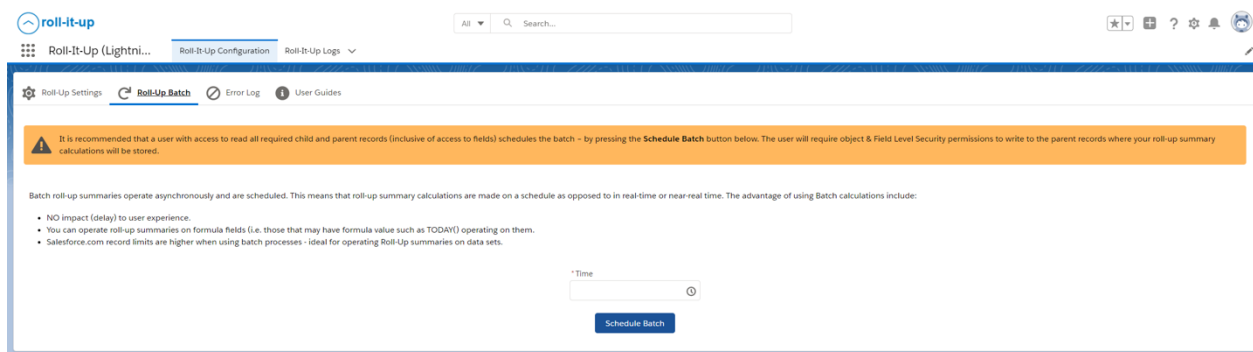
Search for the ‘roll-it-up’ app listing in the Salesforce App Exchange and use the ‘Get it now’ option to install.

Permissions

It is recommended that when installing the application, it be enabled for ‘Admins only’ as users should not require access to create or configure roll-up summaries.

Batch Scheduling

Your System Administrator will need to schedule the ‘roll-it-up’ Batch Process for your roll-up summaries to calculate. Details on how to schedule your ‘roll-it-up’ Batch Process can be found in the *Batch Roll-Up Summaries – Batch Scheduling* section of this guide.



Configuring the Application

'roll-it-up' supports three (3) different Roll-Up summary types.

1. Batch (scheduled),
2. Synchronous (real-time)
3. Asynchronous (near real-time).

See details below of the various roll-up summary types and understand their ideal application.

Batch Roll-Up Summaries

Batch roll-up summaries operate asynchronously and are scheduled. **This means that roll-up summary calculations are made on a schedule** as opposed to in real-time or near-real time.

The advantage of using Batch calculations include;

- NO impact (delay) to user experience.
- You can operate roll-up summaries on formula fields (i.e. those that may have formula value such as TODAY()) operating on them.
- Salesforce.com record limits are higher when using batch processes - ideal for operating Roll-Up summaries on data sets.

NOTE: because Batch Roll-Up calculations re-calculate the values for child records being rolled up to a parent, the Last Modified information of the Parent record will reflect the last time a user or the Batch process modified a record. As such, you may wish to consider scheduling the 'roll-it-up' batch process as a particular user (refer BATCH SCHEDULING below).

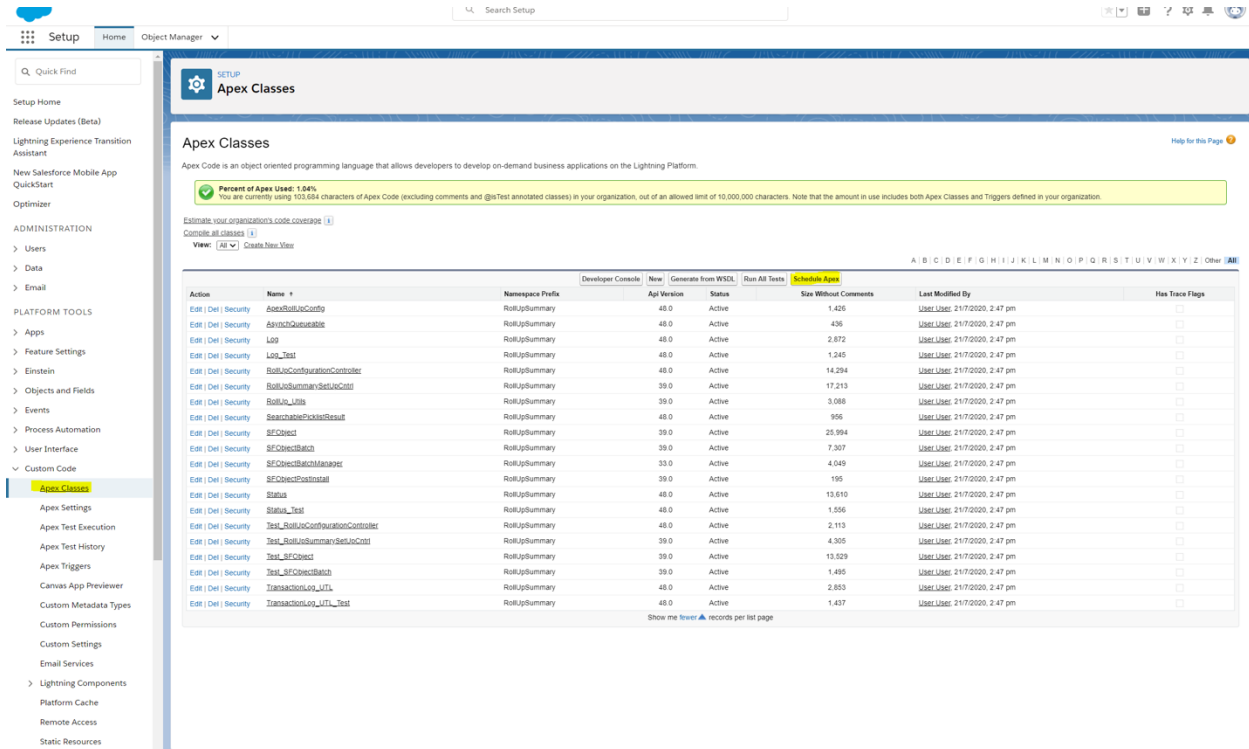
Batch Scheduling

Your System Administrator will need to schedule the 'roll-it-up' Batch Process for your roll-up summaries to calculate. **It is recommended that you schedule your Batch Process with a running user that will have access to read all required child and parent records inclusive of access to fields.** The running user you select will require Object & Field Level Security Permissions to write to the parent records where your roll-up summary calculations will be stored.

Basic Setup; run your batch process once daily

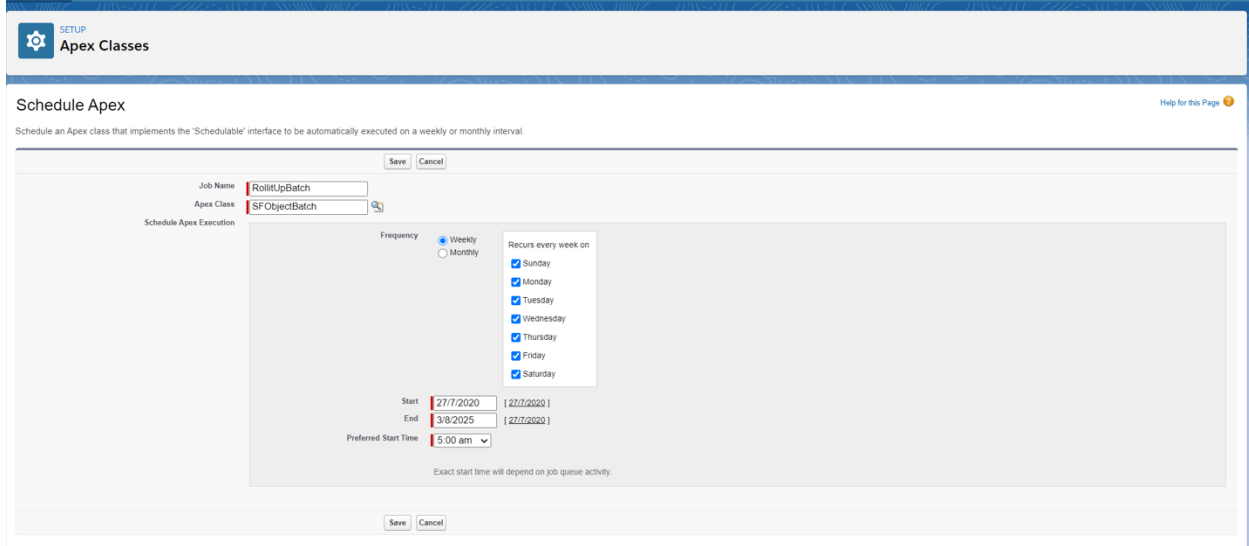
To schedule your 'roll-it-up' Batch Process to run once daily these steps;

- Go to Setup | Custom Code | Apex Classes
- Hit the Schedule Apex Button
- Enter the Job Name as "RollitUpBatch"
- Set the Apex Class to "SFOBJECTBatch"
- Set the Frequency to Weekly and All Days and select an appropriate time.



The screenshot shows the Salesforce Setup page for Apex Classes. The left sidebar contains navigation links for Setup Home, Release Updates (Beta), Lightning Experience Transition Assistant, New Salesforce Mobile App Quickstart, and Optimizer. Under ADMINISTRATION, there are links for Users, Data, and Email. Under PLATFORM TOOLS, there are links for Apps, Feature Settings, Einstein, Objects and Fields, Events, Process Automation, User Interface, and Custom Code. The Custom Code section is expanded, showing Apex Classes, Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers, Canvas App Previewer, Custom Metadata Types, Custom Permissions, Custom Settings, Email Services, Lightning Components, Platform Cache, Remote Access, and Static Resources.

The main content area is titled "Apex Classes" and includes a "Schedule Apex" button. Below this, there is a table listing Apex Classes. The table has columns for Action, Name, Namespace Prefix, API Version, Status, Size Without Comments, Last Modified By, and Has Trace Flags. The table lists 14 Apex Classes, including RollitUpSummary, RollitUpBatch, RollitUpBatchTest, RollitUpBatchTestController, RollitUpBatchTestControllerTest, RollitUpBatchTestControllerTestController, RollitUpBatchTestControllerTestControllerTest, RollitUpBatchTestControllerTestControllerTestController, RollitUpBatchTestControllerTestControllerTestControllerTest, RollitUpBatchTestControllerTestControllerTestControllerTestController, RollitUpBatchTestControllerTestControllerTestControllerTestControllerTest, RollitUpBatchTestControllerTestControllerTestControllerTestControllerTestController, RollitUpBatchTestControllerTestControllerTestControllerTestControllerTestControllerTest, and RollitUpBatchTestControllerTestControllerTestControllerTestControllerTestControllerTestController.



The screenshot shows the Salesforce Setup page for Schedule Apex. The page is titled "Schedule Apex" and includes a "Help for this Page" link. Below the title, there is a description: "Schedule an Apex class that implements the 'Schedulable' interface to be automatically executed on a weekly or monthly interval."

The form contains the following fields:

- Job Name: RollitUpBatch
- Apex Class: SFOBJECTBatch
- Frequency: Weekly (selected), Monthly
- Recurs every week on: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday (all selected)
- Start: 27/7/2020
- End: 3/8/2025
- Preferred Start Time: 5:00 am

At the bottom of the form, there is a note: "Exact start time will depend on job queue activity." There are "Save" and "Cancel" buttons at the bottom of the form.

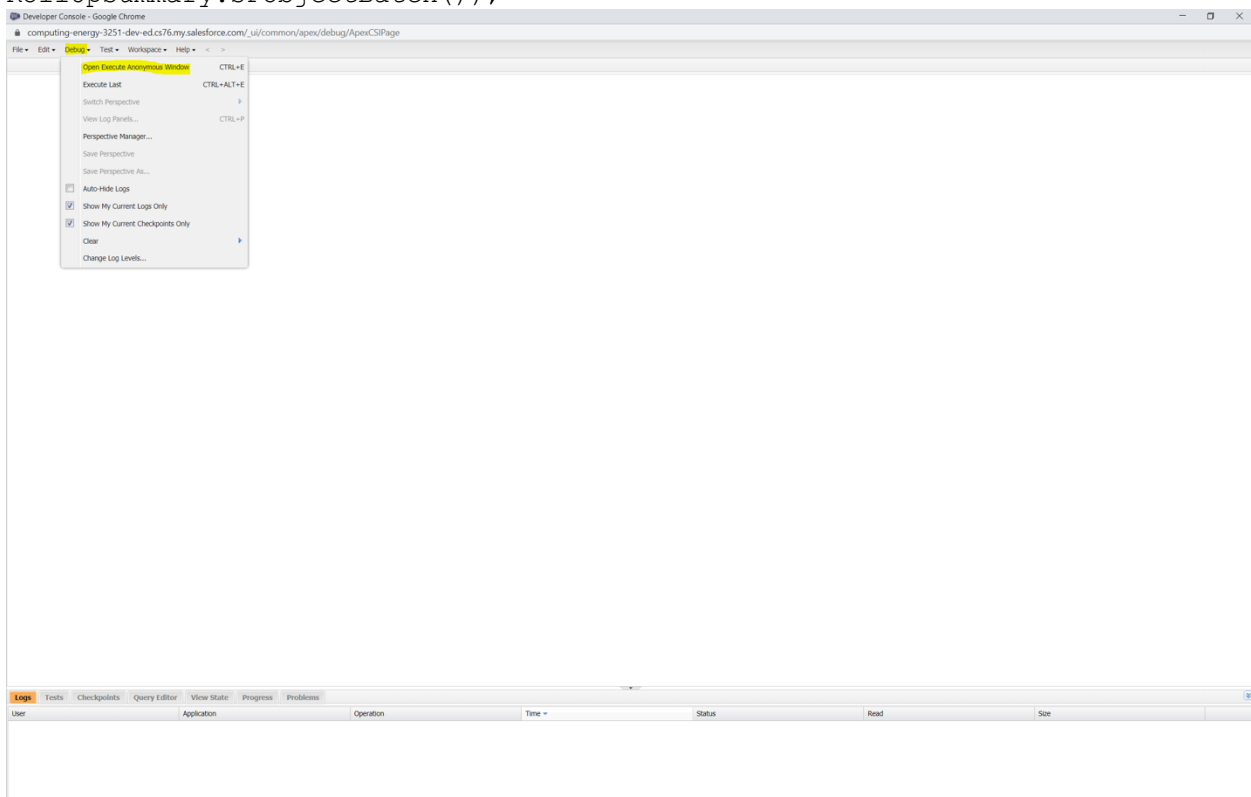
Advanced setup; run your batch processes once hourly

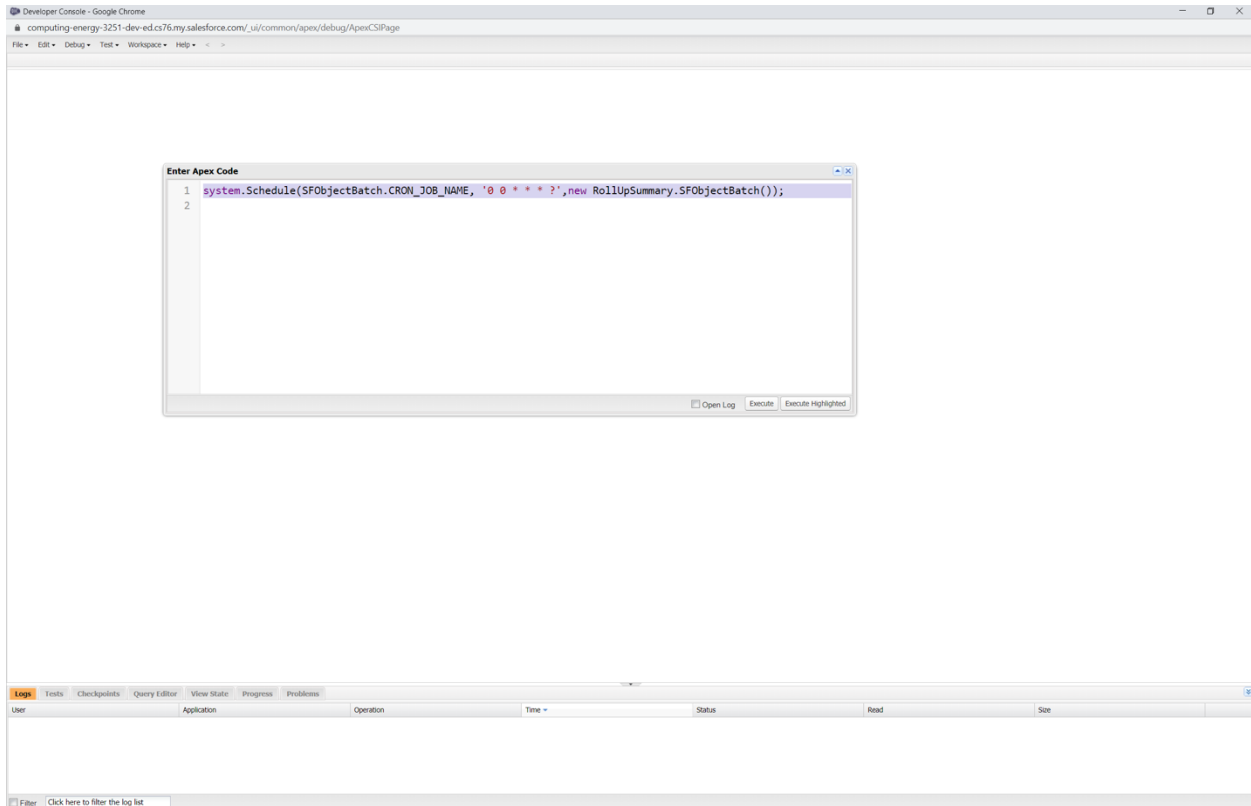
To schedule your Batch Process to run once hourly, you will need to setup the Batch Schedule through the Developer Console (if you are not sure where to find this, click on the Gear that normally takes you to Setup and instead choose “Developer Console”).



You will need to execute the following in the Execute Anonymous section of the Developer Console to run the ‘roll-it-up’ Batch Process hourly:

```
system.Schedule(SFObjectBatch.CRON_JOB_NAME, '0 0 * * * ?', new RollUpSummary.SFObjectBatch());
```





The System Administrator can change the Batch Schedule in the Apex Scheduler. For details see the [Salesforce Apex Schedule Documentation](#).

NOTE: it is highly recommended that System Administrators consider the number of Roll-Up summaries being carried out in Batch and the size of the data sets involved. **Careful consideration should be given to the frequency of the Batch Schedule so as to ensure that jobs are allowed to finish before the next scheduled job commences.** Typically, setting processing to hourly should allow most organisations to complete their Batch processing before the next process commences.

Synchronous (real-time) roll-up summaries

The Synchronous roll-up summaries operate in real-time and deliver roll-up summary results to the user immediately when a child record is:

- inserted or updated, and
- when it is part of your roll-up summary configuration.

It is important to note that real-time roll-up summaries impact the user experience.

Calculations are performed on the spot and can impact the time it takes to view a record that has been inserted or updated.

‘roll-it-up’ has the ability to perform real-time calculations on the following standard objects if used as a child in a lookup or master-detail relationship:

- Account
- Contact
- Case
- Task
- Event
- Lead
- Opportunity
- Attachment
- Note
- User

NOTE: For other standard and custom objects that require a real-time (synchronous) calculation, an Apex Trigger will need to be inserted into your Org. For details on how to insert an Apex Trigger, please refer to the ADVANCED ROLL-UP SUMMARY SETUP section of this document. The Google Maps Settings page is where you will optionally maintain your Google Maps API access preferences.

Asynchronous (near real-time) roll-up summaries

Asynchronous roll-up summaries operate in near real-time and deliver roll-up summary to the user shortly after a child record is inserted or updated that is part of your roll-up summary configuration. **It is important to note that near real-time Roll-Up summaries DO NOT impact the user experience** in that calculations are performed asynchronously (as resources become available) and in the background. **This can mean that the results of your roll-up summary may not be immediately visible to the user** on the parent record to which you are rolling up to but should be visible shortly after.

Out of the box *'roll-it-up'* ships with the ability to perform real-time calculations on the following standard objects (if used as a child in a lookup or master-detail relationship);

- Account
- Contact
- Case
- Task
- Event
- Lead
- Opportunity
- Attachment
- Note
- User

NOTE: For other standard and custom objects that require a real-time (synchronous) calculation, an Apex Trigger will need to be inserted into your Org. For details on how to insert an Apex Trigger, please refer to the ADVANCED ROLL-UP SUMMARY SETUP section of this document.

Roll-up summary types

A roll-up summary field calculates values from related records, such as those in a related list. You can create a roll-up summary field to display a value in a master record based on the values of fields in a detail record. The detail record must be related to the master through a master-detail or lookup relationship.

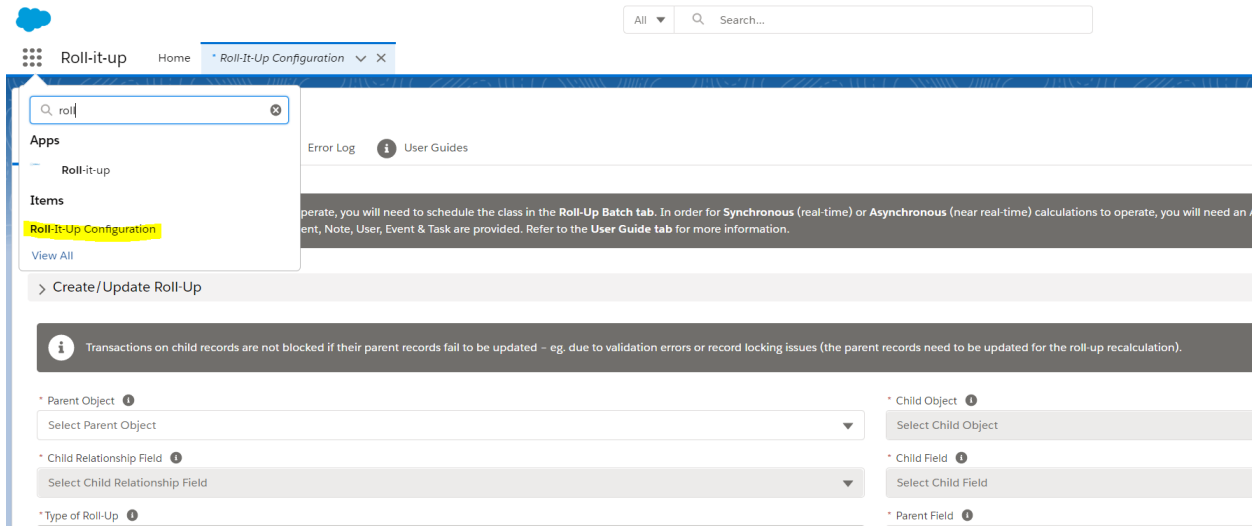
You can perform different types of calculations with a *'roll-it-up'*. You can count (or count distinct) the number of detail records related to a master record. Or, you can calculate the sum, average, minimum value, or maximum value of a field in the detail records.

Using filters in roll-up summaries

'roll-it-up' lets you apply simple or complex filters to your roll-up calculations. Using SOQL (Salesforce Object Query Language) you can create filter conditions on your roll-up summary definitions. Refer to [this Salesforce SOQL & SOSL document](#) for further information.

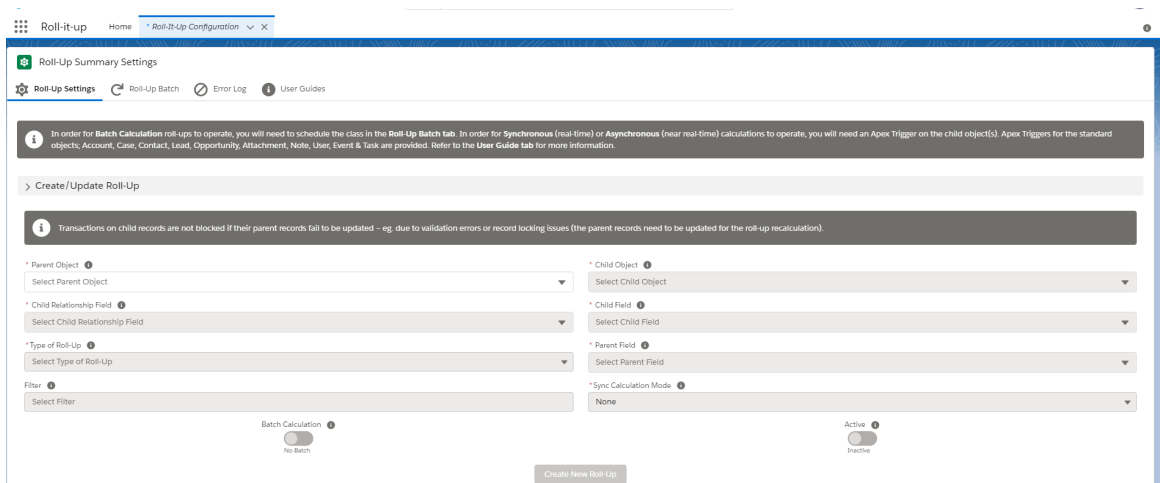
Configuring your first roll-up summary

To configure your first roll-up summary with ‘roll-it-up’ you will need to navigate to the ‘roll-it-up’ configuration, which should be visible in your App menu:



Go to the ‘roll-it-up’ configuration tab to access the ‘roll-up’ settings page. The ‘roll-up’ settings page is broken into two sections:

- Create/Update Roll-Up – this is where you configure your new roll-ups and/or make edits to currently existing ones, once selected.



- Existing Rollups – this where you will find your currently existing rollups, and select them to make edits as needed.

Existing Roll-Ups						
Active	Parent Field	Parent Object	Child Object	Type of Roll-Up	Filter	Sync Mode
☒	RollUpSummary__TotalTestPrices__c	Account	RollUpSummary__AccountTest__c	count	None	

Roll-Up Settings

The roll-up settings page will guide you through the setup of valid roll-up summaries and present you with valid selections for the parent/child roll-up you are looking to create.

When configuring ‘roll-it-up’ the following settings are available (and most are required):

Setting name	Required	Description
PARENT OBJECT	Yes	The parent object that you wish to write your roll-up summary data to.
CHILD OBJECT	Yes	The child object used in a roll-up summary calculation.
CHILD RELATIONSHIP FIELD	Yes	The name of the field on the child record that represents the lookup or master-detail relationship for which you are defining your roll-up summary.
CHILD FIELD	Yes	The child field to be used in the roll-up summary. You can introduce additional fields in the filter section as required.
TYPE OF ROLL-UP	Yes	The type of roll-up summary calculation to be performed. Options include COUNT, COUNT DISTINCT, SUM, AVERAGE, MIN and MAX.
PARENT FIELD	Yes	The field to which the results of the roll-up summary calculation will be written. You will need to create this field in the object setup before configuring the rollup so that field can be referenced as the “Parent Field”.
FILTER	No	Used to apply filters (using SOQL WHERE type syntax) to roll-up summary calculations. Refer to the Using filters in roll-up summaries section of this document for details on filter conditions.

SYNC CALCULATION MODE	Yes	Determines the mode in which your roll-up summary will run. Options include; Synchronous (real-time), Asynchronous (near real-time), None (not applicable) – typically selected if you intend to operate the roll-up summary calculation in batch only.
BATCH CALCULATION (toggle)	No	Indicates if you intend to operate the roll-up summary calculation in Batch Mode (i.e. will be picked up as part of the Schedule Batch Job process). If you intend to choose “Batched”, ensure that you first schedule the batch in the “Roll-Up Batch” tab. It is recommended that if you intend to perform a roll-up summary on a formula field in a child record that is subject to change without user intervention (i.e. formula fields that contain date/time values) that you select Batch as true. It is also feasible to have roll-up summary calculations that operate synchronously (real-time) or asynchronously (near real-time) in addition to Batch (i.e. synchronous or asynchronous can co-exist with batch). This is ideal for scenarios where you would like the Batch process to act as a "sweeper" recalculating all roll-up summaries on a schedule.
ACTIVE (toggle)	No	Indicates if your ‘roll-it-up’ summary configuration should be active and included in Batch, Synchronous or Asynchronous calculations. If Active is not checked the roll-up summary will not be run.

As an example of how to navigate these fields, here is an example of the settings completed for a roll-up field that populates a field on Account named “Average Oppty Value” that calculates the average value of all opportunities on an account asynchronously (near real-time):

Create/Update Roll-Up

Transactions on child records are not blocked if their parent records fail to be updated – eg. due to validation errors or record locking issues (the parent records need to be updated for the roll-up recalculation).

Parent Object
Account

Child Relationship Field
Account ID

Type of Roll-Up
avg

Filter
Select Filter

Child Object
Opportunity

Child Field
Amount

Parent Field
Avg. Oppty Value

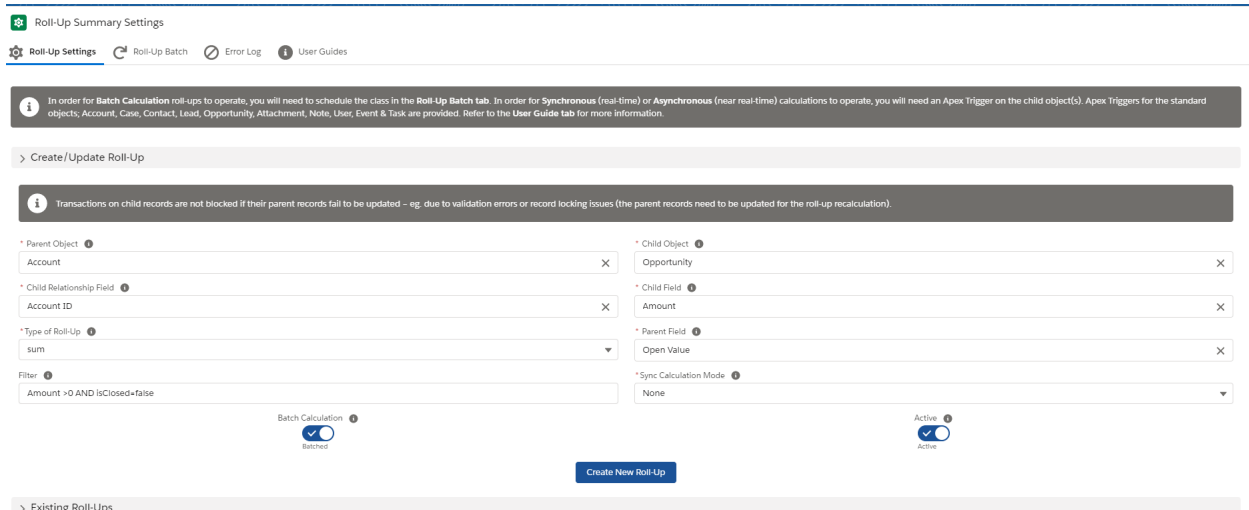
Sync Calculation Mode
Asynchronous

Batch Calculation
No Batch

Active
Active

Create New Roll-Up

Here is another example of a roll-up that populates a field on Account named “Open Value” that calculates the sum of open opportunities for that account via batch:



Roll-Up Summary Settings

[Roll-Up Settings](#) [Roll-Up Batch](#) [Error Log](#) [User Guides](#)

Information: In order for Batch Calculation roll-ups to operate, you will need to schedule the class in the Roll-Up Batch tab. In order for Synchronous (real-time) or Asynchronous (near real-time) calculations to operate, you will need an Apex Trigger on the child object(s). Apex Triggers for the standard objects: Account, Case, Contact, Lead, Opportunity, Attachment, Note, User, Event & Task are provided. Refer to the User Guide tab for more information.

[Create/Update Roll-Up](#)

Information: Transactions on child records are not blocked if their parent records fail to be updated – eg. due to validation errors or record locking issues (the parent records need to be updated for the roll-up recalculation).

Parent Object: Account

Child Object: Opportunity

Child Relationship Field: Account ID

Child Field: Amount

Type of Roll-Up: sum

Parent Field: Open Value

Filter: Amount > 0 AND IsClosed=false

Batch Calculation: Batch

Sync Calculation Mode: None

Active: Active

[Create New Roll-Up](#)

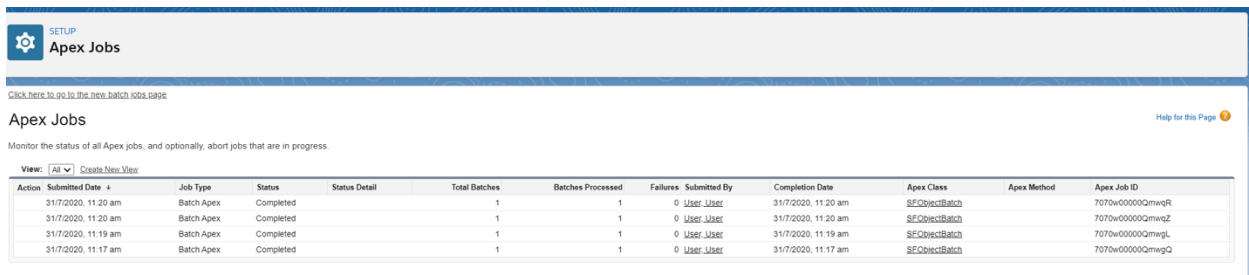
[Existing Roll-Ups](#)

Testing your first batch roll-up summary

The above example will run in batch (if you have not modified the Batch Job Schedule since installation), this will be hourly. Check an Account record that has an open Opportunity and a value greater than 0 once the batch process has run, the Account record should have the Open Value field populated.

Verify the batch Job has run

From the Apex Jobs section there should be an entry of SFOBJECTBATCH:



Apex Jobs

Click here to go to the new batch jobs page

Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.

View: All Create New View

Action	Submitted Date +	Job Type	Status	Status Detail	Total Batches	Batches Processed	Failures	Submitted By	Completion Date	Apex Class	Apex Method	Apex Job ID
	31/7/2020, 11:20 am	Batch Apex	Completed		1	1	0	User User	31/7/2020, 11:20 am	SFOBJECTBATCH		7070w00000QmwrR
	31/7/2020, 11:20 am	Batch Apex	Completed		1	1	0	User User	31/7/2020, 11:20 am	SFOBJECTBATCH		7070w00000QmwrZ
	31/7/2020, 11:19 am	Batch Apex	Completed		1	1	0	User User	31/7/2020, 11:19 am	SFOBJECTBATCH		7070w00000QmwrL
	31/7/2020, 11:17 am	Batch Apex	Completed		1	1	0	User User	31/7/2020, 11:17 am	SFOBJECTBATCH		7070w00000QmwrQ

Existing Rollups

Your ‘roll-it-up’ existing rollups will be visible in the section immediately below “Create/Update Roll-Up”. The visible fields here mirror most of the settings from when the roll up was created in the above “Create/Update Roll-Up” section.

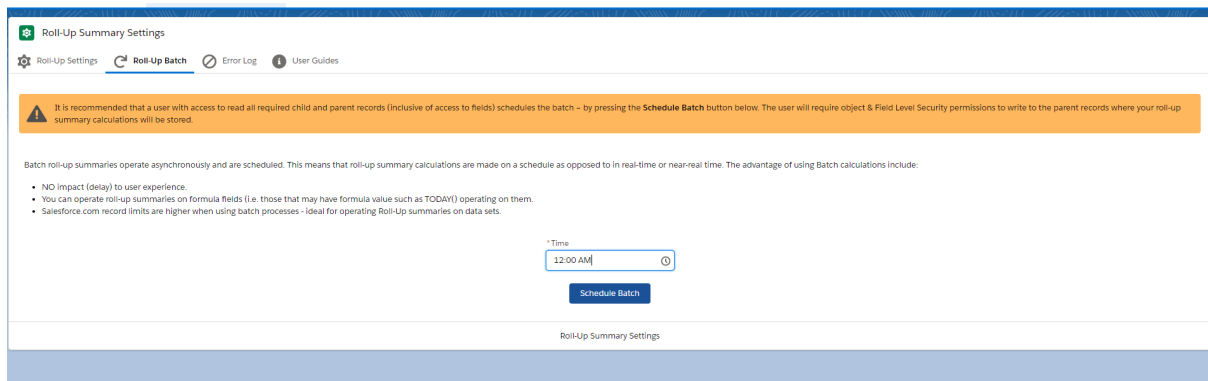
Roll-Up Batch

Should you create a roll-up that is batched, you will need to first schedule the batch in this tab of the App (“Roll Up Batch”).

Batch roll-up summaries can be performed on any standard or custom object where a lookup or master-detail relationship is in place.

There is no need for Apex Triggers to be in place in for Batch roll-up calculations to operate.

To schedule the batch, simply enter in the time and choose “Schedule Batch” as in this example for the batch to run at midnight:



Roll-Up Summary Settings

Roll-Up Settings **Roll-Up Batch** Error Log User Guides

Warning: It is recommended that a user with access to read all required child and parent records (inclusive of access to fields) schedules the batch - by pressing the **Schedule Batch** button below. The user will require object & Field Level Security permissions to write to the parent records where your roll-up summary calculations will be stored.

Batch roll-up summaries operate asynchronously and are scheduled. This means that roll-up summary calculations are made on a schedule as opposed to in real-time or near-real time. The advantage of using Batch calculations include:

- NO Impact (delay) to user experience.
- You can operate roll-up summaries on formula fields (i.e. those that may have formula value such as TODAY()) operating on them.
- Salesforce.com record limits are higher when using batch processes - Ideal for operating Roll-Up summaries on data sets.

*Time
12:00 AM

Schedule Batch

Roll-Up Summary Settings

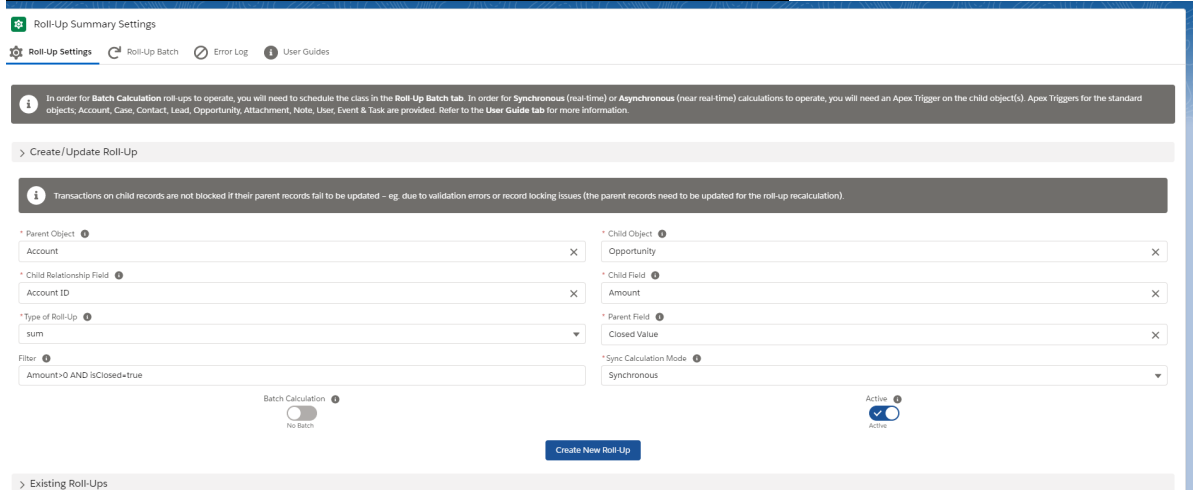
More Examples: create a synchronous (real-time) roll-up summary

See below a simple outline for settings that could be used on a Synchronous (real-time) ‘roll-it-up’ summary for a Standard object. In this example we will SUM up the amount of closed Opportunities (the children) for a specific Account (the parent) and write that value to the Account record.

PRE-REQUISITE: Recall that you will need to create a new field on Account (type currency) to store the roll-up calculated value (for this example we will refer to the newly created field on Account as Closed Value).

Setting name	Description
PARENT OBJECT	Account
CHILD OBJECT	Opportunity
CHILD RELATIONSHIP FIELD	Account ID
CHILD FIELD	Amount

TYPE OF ROLL-UP	SUM
PARENT FIELD	Closed Value
FILTER	Amount>0 AND isClosed=true
SYNC CALCULATION MODE	Synchronous
BATCH CALCULATION	NO BATCH
ACTIVE	ACTIVE



Roll-Up Summary Settings

Roll-Up Settings | Roll-Up Batch | Error Log | User Guides

Information: In order for Batch Calculation roll-ups to operate, you will need to schedule the class in the Roll-Up Batch tab. In order for Synchronous (real time) or Asynchronous (near real-time) calculations to operate, you will need an Apex Trigger on the child object(s). Apex Triggers for the standard objects, Account, Case, Contact, Lead, Opportunity, Attachment, Note, User, Event & Task are provided. Refer to the User Guide tab for more information.

Create/Update Roll-Up

Warning: Transactions on child records are not blocked if their parent records fail to be updated - eg. due to validation errors or record locking issues (the parent records need to be updated for the roll-up recalculation).

* Parent Object: Account

* Child Relationship Field: Account ID

* Type of Roll-Up: sum

Filter: Amount>0 AND isClosed=true

* Child Object: Opportunity

* Child Field: Amount

* Parent Field: Closed Value

* Sync Calculation Mode: Synchronous

Batch Calculation: ☐ No Batch

Active: ☒ Active

Create New Roll-Up

Existing Roll-Ups

Testing your first synchronous roll-up summary

The above example will run synchronously (real-time). Edit an Opportunity record that has a value greater than 0 and set the Stage to Closed - Won then save your changes. Check an Account record that has an open Opportunity with an Amount greater than 0 and Stage value equal to Closed - Won or Stage value equal to Closed Lost. The Closed Value on the Account (parent) record should be populated.

More Examples: Create an asynchronous (near real-time) roll-up summary for standard objects

Asynchronous (real-time) roll-up summaries can be performed on any Standard or Custom object where a lookup or master-detail relationship is in place.

Standard objects include:

- Account
- Contact
- Case
- Task
- Event

- Lead
- Opportunity
- Attachment
- Note
- User

See below a simple outline for settings that could be used on a Synchronous (real-time) ‘roll-it-up’ summary for a Standard object. In this example we will SUM up the amount of closed Opportunities (the children) for a specific Account (the parent) and write that value to the Account record.

PRE-REQUISITE: Recall that you will need to create a new field on Account (type number) to store the roll-up calculated value (for this example we will refer to the newly created field on Account as Open Opportunity Count).

Setting name	Description
PARENT OBJECT	Account
CHILD OBJECT	Opportunity
CHILD LOOKUP FIELD	Account
CHILD FIELD	Amount
TYPE OF ROLL-UP	COUNT
PARENT FIELD	Open Opportunity Count
FILTER	Amount>0 AND isClosed=true
SYNC CALCULATION MODE	Asynchronous
BATCH CALCULATION	NO BATCH
ACTIVE	ACTIVE

Roll-Up Summary Settings

Roll-Up Settings | Roll-Up Batch | Error Log | User Guides

Info In order for **Batch Calculation** roll-ups to operate, you will need to schedule the class in the **Roll-Up Batch** tab. In order for **Synchronous (real time)** or **Asynchronous (near real-time)** calculations to operate, you will need an Apex Trigger on the child object(s). Apex Triggers for the standard objects: Account, Case, Contact, Lead, Opportunity, Attachment, Note, User, Event & Task are provided. Refer to the **User Guide** tab for more information.

> Create/Update Roll-Up

Info Transactions on child records are not blocked if their parent records fail to be updated – eg. due to validation errors or record locking issues (the parent records need to be updated for the roll-up recalculation).

* Parent Object **Account**

* Child Relationship Field **Account ID**

* Type of Roll-Up **count**

Filter **Amount > 0 AND IsClosed = true**

* Child Object **Opportunity**

* Child Field **Amount**

* Parent Field **Open Opportunity Count**

* Sync Calculation Mode **Asynchronous**

Batch Calculation **Off**

Active **On**

Create New Roll-Up

Testing your first asynchronous roll-up summary

The above example will run asynchronously (near real-time). Edit an Opportunity record that has a value greater than 0 and set the Stage not equal Closed - Won or Closed - Lost then save your changes. Because the roll-up summary will run asynchronously it may take a short while for the roll-up summary to calculate. After some time, check the Account record that had the open Opportunity you edited and an amount greater than 0 and not in Stage Closed - Won or Stage Closed - Lost. The Open Opportunity Count field should be populated with a count of Opportunities for that Account which meet that criteria.

Error Log

Email Notification

The ‘roll-it-up’ application allows you to configure workflow rule-based email alerts when an error occurs during calculation of roll-ups

This is an optional feature that can be of great use to the Salesforce Admin.

To configure the email alerts, follow the steps displayed in the “Error Log” tab and:

- Activate the **Send Roll-It-Up Error Email** Workflow Rule (WFR), and
- Update the **Recipients** of the Email Alert to include relevant users (ex: the Salesforce System Administrator)
- If desired, update the **From Email Address** to list out a relevant email (ex: generic shared email for the organisation)

roll-it-up

Roll-It-Up (Lightni...

Roll-It-Up Configuration

Roll-It-Up Logs

Roll-Up Settings

Roll-Up Batch

Error Log

User Guides

Email Notification

If you would like email alerts to be sent when an error occurs during calculation of roll-ups:

- Press the **Activate/Deactivate Workflow Rule** button below
- Activate the **Send Roll-Up Error Email** workflow rule

Then, define the recipient(s) of the email alert by following these steps:

- Press the **Configure Email Alert** button below
- Open **Roll-It-Up Error Email Alert**
- Update its **Recipients**
- Update its **From Email Address** (if needed)

Activate/Deactivate Workflow Rule

Configure Email Alert

Errors

If you're receiving LimitException errors (eg. too many query rows, Apex CPU time limit exceeded etc.), consider changing your roll-up calculation modes to **Asynchronous** or **Batch**. These errors are difficult to avoid if you have large amounts of data, particularly if it's coupled with a complicated data model and lots of cascading operations. For instance, a move to **Asynchronous** mode gives you a 60 second CPU limit per batch instead of the 10 second synchronous single transaction limit.

Log name	Severity	Summary	Message	Created...
L-000000008	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE (AND((email=null) OR ((Phone=null OR Mobile=null)))) AND ID = " LIMIT 1)	unexpected token: 'AND'	31 Jul 2020, 01:46:04 pm
L-000000007	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE (AND((email=null) OR ((Phone=null OR Mobile=null)))) AND ID = " LIMIT 1)	unexpected token: 'AND'	31 Jul 2020, 01:45:45 pm
L-000000006	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE ((Phone=null) AND ID = " LIMIT 1)	expecting a right parentheses, found 'LIMIT'	31 Jul 2020, 01:45:00 pm
L-000000005	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE ((Phone=null OR Mobile=null)) AND ID = " LIMIT 1)	No such column 'Mobile' on entity 'Contact'. If you are attempting to use a custom field, be sure to append the '._c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.	31 Jul 2020, 01:44:55 pm
L-000000004	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE ((Phone=null OR Mobile=null)) AND ID = " LIMIT 1)	No such column 'Mobile' on entity 'Contact'. If you are attempting to use a custom field, be sure to append the '._c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.	31 Jul 2020, 01:44:06 pm
L-000000003	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE ((Phone=null OR Mobile=null) AND (email=null)) AND ID = " LIMIT 1)	No such column 'Mobile' on entity 'Contact'. If you are attempting to use a custom field, be sure to append the '._c' after the custom field name. Please reference your WSDL or the describe call for the appropriate names.	31 Jul 2020, 01:43:35 pm
L-000000002	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId ,	No such column 'Mobile' on entity 'Contact'. If you are attempting to use a custom field, be sure to append the '._c' after the custom field name. Please reference your WSDL or the describe call for the appropriate	31 Jul 2020,

Setup

Home

Object Manager

Quick Find

Setup Home

Release Updates (Beta)

Lightning Experience Transition Assistant

New Salesforce Mobile App QuickStart

Optimizer

ADMINISTRATION

Users

Data

Email

PLATFORM TOOLS

Apps

Workflow Rules

All Workflow Rules

Configure your organization's workflow by creating workflow rules. Each workflow rule consists of:

- Criteria that cause the workflow rule to run.
- Immediate actions that execute when a record matches the criteria. For example, Salesforce can automatically send an email that notifies the account team when a new high-value opportunity is created.
- Time-dependent actions that queue when a record matches the criteria, and execute according to time triggers. For example, Salesforce can automatically send an email reminder to the account team if a high-value opportunity is still open ten days before the close date.

View: All Workflow Rules Create New Rule

Action	Rule Name	Description	Object	Active
Edit Delet Activate	Send Roll-Up Error Email	Sends an email alert when a Roll-Up log record is created (which is created due to an error).	Roll-Up Log	☒



SETUP

Email Alerts

Email Alert

Roll-It-Up Error Email Alert

[Help for this Page](#)

Rules Using This Email Alert (1)

Approval Processes Using This Email Alert (0)

Email Alert Detail

Description

Roll-It-Up Error Email Alert

Edit

Delete

Clone

Unique Name

Roll-It-Up_Error_Email_Alert

Protected Component

☐

Namespace Prefix

RollUpSummary

Email Template

Roll-It-Up_Error_Email_Template

From Email Address

Current User's email address

Object

Roll-It-Up Log

Recipients

Record Creator

Additional Emails

Created By

User User: 31/7/2020, 10:41 am

Modified By

User User: 31/7/2020, 10:41 am

Rules Using This Email Alert

Rules Using This Email Alert Help

Action	Rule Name	Description	Object	Active
Edit Del Activate	Send Roll-It-Up Error Email	Sends an email alert when a Roll-It-Up log record is created (which is created due to an error).	Roll-It-Up Log	☐

Approval Processes Using This Email Alert

Approval Processes Using This Email Alert Help

This alert is currently not used by any approval processes

[Back To Top](#)

Always show me [more records per related list](#)

Edit Email Alert

Roll-It-Up Error Email Alert

[Help for this Page](#)

Create an email alert to associate with one or more workflow rules, approval processes, or entitlement processes. When changing an email alert, any modifications will apply to all rules, approvals, or entitlement processes associated with it.

Email Alert Edit

Save

Save & New

Cancel

Edit Email Alert

Description

Roll-It-Up Error Email Alert

Unique Name

Roll-It-Up_Error_Email_A

Namespace Prefix

RollUpSummary

Object

Roll-It-Up Log

Email Template

Roll-It-Up Error Email Tem

Protected Component

☐

Recipients

Recipient Type

User

Search

Find

Available Recipients

User: Integration User

User: Security User

User: User User

Add

Remove

Selected Recipients

Record Creator

You can enter up to five (5) email addresses to be notified.

Additional Emails

From Email Address

Current User's email address

☐ Make this address the default From email address for this object's email alerts.

Save

Save & New

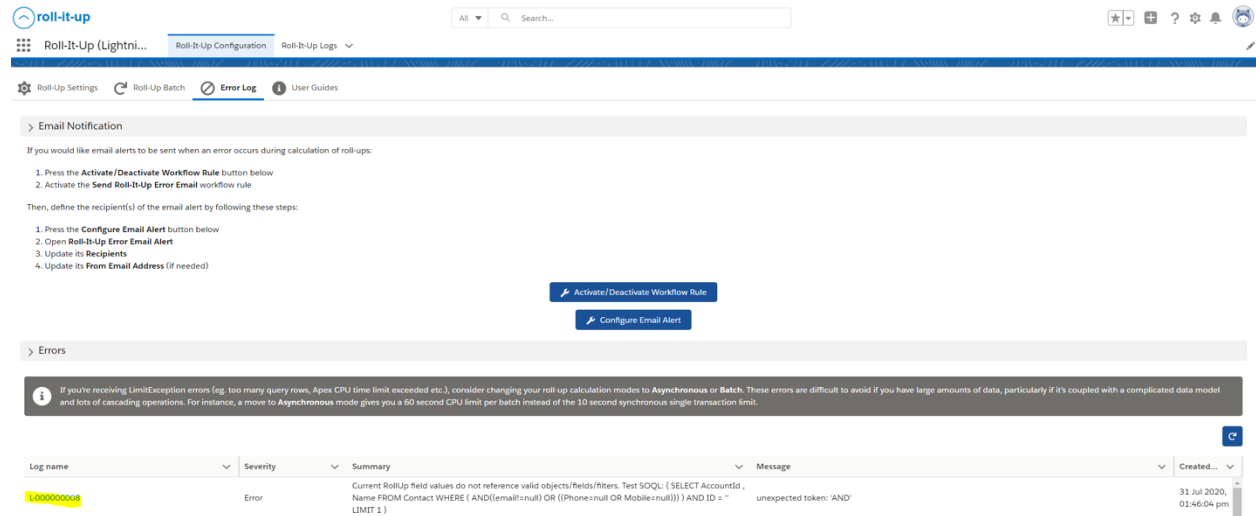
Cancel

"roll-it-up" Roll-Up Summary Helper Configuration and User Guide R1.5

Page 21 of 25

Errors

If there are an error message sent out, they will also appear in this **Errors** section of the App. Here is one example:



Email Notification

If you would like email alerts to be sent when an error occurs during calculation of roll-ups:

1. Press the **Activate/Deactivate Workflow Rule** button below
2. Activate the **Send Roll-It-Up Error Email** workflow rule

Then, define the recipient(s) of the email alert by following these steps:

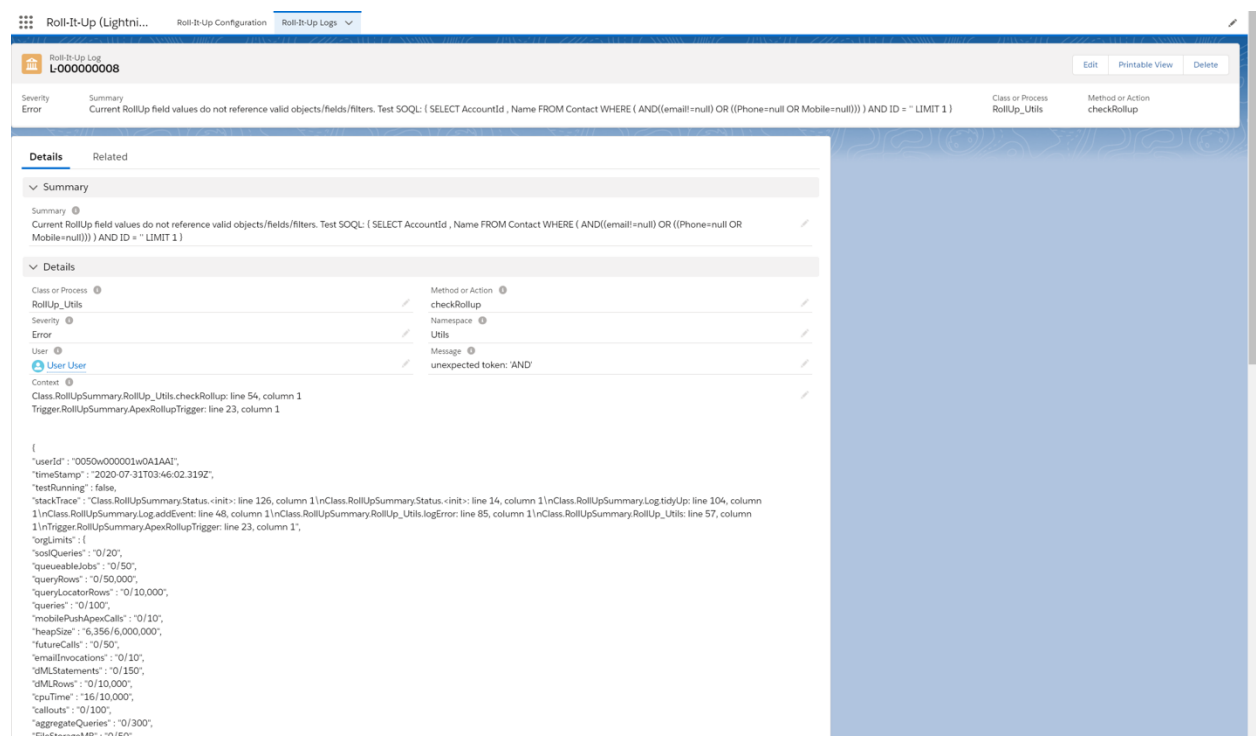
1. Press the **Configure Email Alert** button below
2. Open **Roll-It-Up Error Email Alert**
3. Update its **Recipients**
4. Update its **From Email Address** (if needed)

Errors

If you're receiving **LimitException** errors (eg. too many query rows, Apex CPU time limit exceeded etc.), consider changing your roll-up calculation modes to **Asynchronous** or **Batch**. These errors are difficult to avoid if you have large amounts of data, particularly if it's coupled with a complicated data model and lots of cascading operations. For instance, a move to **Asynchronous** mode gives you a 60 second CPU limit per batch instead of the 10 second synchronous single transaction limit.

Log name	Severity	Summary	Message	Created...
L000000008	Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE (AND((email=null) OR ((Phone=null OR Mobile=null)))) AND ID = " LIMIT 1)	unexpected token: 'AND'	31 Jul 2020, 01:46:04 pm

From clicking on the Log name hyperlink, more details appear from the log record:



Roll-It-Up Log
L000000008

Edit Printable View Delete

Severity	Summary	Class or Process	Method or Action
Error	Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE (AND((email=null) OR ((Phone=null OR Mobile=null)))) AND ID = " LIMIT 1)	RollUp_Units	checkRollUp

Details

Summary

Summary
Current RollUp field values do not reference valid objects/fields/filters. Test SOQL: (SELECT AccountId , Name FROM Contact WHERE (AND((email=null) OR ((Phone=null OR Mobile=null)))) AND ID = " LIMIT 1)

Details

Class or Process	Method or Action
RollUp_Units	checkRollUp
Severity	Message
Error	unexpected token: 'AND'
User	User User
Context	Class.RollUpSummary.RollUp_Units.checkRollUp: line 54, column 1 Trigger.RollUpSummaryApexRollUpTrigger: line 23, column 1

```
{
  "userId": "0050w000001w0A1AAI",
  "timestamp": "2020-07-31T03:46:02.319Z",
  "testRunning": false,
  "stackTrace": "Class.RollUpSummaryStatus.<init>: line 126, column 11;Class.RollUpSummaryStatus.<init>: line 14, column 11;Class.RollUpSummaryLogtidUp: line 104, column 11;Class.RollUpSummaryLog.addEvent: line 48, column 11;Class.RollUpSummary.RollUp_Units.logError: line 85, column 11;Class.RollUpSummary.RollUp_Units: line 57, column 11;Trigger.RollUpSummaryApexRollUpTrigger: line 23, column 1",
  "orgLimits": {
    "soslQueries": "0/20",
    "queueableJobs": "0/50",
    "queryRows": "0/50,000",
    "queryLocatorRows": "0/10,000",
    "queries": "0/100",
    "mobilePushApexCalls": "0/10",
    "heapSize": "6,356/6,000,000",
    "futureCalls": "0/50",
    "emailInvocations": "0/10",
    "dmlStatements": "0/150",
    "dmlRows": "0/10,000",
    "cpuTime": "16/10,000",
    "callouts": "0/100",
    "aggregateQueries": "0/300",
    "fileStorageMB": "0/50"
  }
}
```

If you're receiving **LimitException** errors (eg. too many query rows, Apex CPU time limit exceeded etc.), consider changing your roll-up calculation modes to **Asynchronous** or **Batch**. These errors are difficult to avoid if you have large amounts of data, particularly if it's coupled

with a complicated data model and lots of cascading operations. For instance, a move to **Asynchronous** mode gives you a 60 second CPU limit per batch instead of the 10 second synchronous single transaction limit.

User Guides

This guide may be referenced from within the Roll-Up Summary Settings itself by navigating to the **User Guides** App.

There are instructional sections specific to the following use cases:

- Creating a Roll-Up (also see “Roll-Up Settings” section of this document)
- Enabling Batch Calculations (also see “Roll-Up Batch” section of this document)
- Enabling Sync Calculations (also see “More Examples: create a synchronous (real-time) roll-up summary” section of this document)

Advanced ‘roll-it-up’ summary setup

Advanced roll-up summaries can be considered as those that require the inclusion of Apex Triggers (i.e. Synchronous (real-time) or Asynchronous (near real-time) that are not operating on Standard Objects or those where complex filters may be employed.

This guide is not designed or intended to provide exhaustive instructions on all types of roll-up summaries possible. The ‘roll-it-up’ application provides significant flexibility for Administrators and Developers to construct complex roll-up summary calculations using custom objects, formula fields (consider using Batch) and SOQL WHERE syntax.

A key component of creating synchronous (real-time) and asynchronous (near real-time) calculations is the creation of Apex Triggers. See the below section for details on how to employ Apex triggers for us with ‘roll-it-up’.

Creating an Apex Trigger

To use Synchronous (Real-Time) & Asynchronous (Near Real-Time) calculations on additional you will need to create and deploy Apex Triggers to your Org. This document is not intended to teach Administrators and Developers how to create Apex Triggers other than to state that Apex Triggers are required.

Different constraints exist pertaining to the creation and deployment of Apex Triggers depending on the type of Org you are operating in. For example, Apex Triggers cannot be created directly in Production Org's and must be built outside of Production and deployed to Production Orgs.

Refer to [this article](#) for further details on the creation and deployment of Apex Triggers.

See below details of a sample Apex Trigger that can be adapted to meet the needs of your organisation when looking to leverage synchronous (real-time) and asynchronous (near real-time) roll-up summary calculations.

Apex Trigger example

In the following example Apex Trigger pattern, we are defining the Trigger required for a Child Custom Object.

```
/**
 * Handles the order of execution for the child object. *
 * @author DataColada Pty Ltd
 * @author DataColada Pty Ltd
 * @since 30 July 2020
 * @date 30 July 2020 - created initial trigger structure
 * @see RollUpSummary.SFObject.rollUp(list<sObject> items);
 */

trigger customObjectChildTrigger on Custom_Object_c(after delete, after
insert, after undelete, after update, before delete, before insert, before
update) {

    /* Before Insert*/
    if (Trigger.isInsert && Trigger.isBefore) {
    }
    /* After Insert*/
    else if (Trigger.isInsert && Trigger.isAfter) {
        RollUpSummary.SFObject.rollUp((List<sObject>) trigger.new);
    }
    /* Before Update*/
    else if (Trigger.isUpdate && Trigger.isBefore) {
    }
    /* After Update*/
    else if (Trigger.isUpdate && Trigger.isAfter) {
        RollUpSummary.SFObject.rollUp((List<sObject>) trigger.new,
(List<sObject>) trigger.old);
    }
    /* Before Delete */
    else if (Trigger.isDelete && Trigger.isBefore) {
    }
    /* After Delete */
    else if (Trigger.isDelete && Trigger.isAfter) {
        RollUpSummary.SFObject.rollUp((List<sObject>) trigger.old);
    }
    /* After Undelete*/
    else if (Trigger.isUnDelete) {
        RollUpSummary.SFObject.rollUp((List<sObject>) trigger.new);
    }
}
```


Paste the above example and make sure the necessary changes to the name and the object in which the trigger is operating, then deploy to your organisation for testing.

Testing Apex Triggers in a synchronous (real-time) or asynchronous (near real-time) roll-up summary

Once your Apex Trigger is deployed to the Org, then configure a synchronous (real-time) or asynchronous (near real-time) roll- up summary that operates on the custom object (that is, non-standard object) for which you created the Apex Trigger.

Refer to the section Synchronous (real-time) roll-up for standard objects and Asynchronous (near real-time) roll-up for standard objects for examples of configuring the required roll-up summary types.