

Flow Structural Conventions - Scheduled

As detailed in the General Notes section, these conventions are heavily opinionated towards maintenance and scaling in large organizations. The conventions contain:

- a "[common core](#)" set of structural conventions that apply everywhere
- conventions for [Record Triggered Flows](#) specifically
- conventions for Scheduled Flows specifically (this page!)

Scheduled Flow Design

As detailed in the Common Core conventions, despite it not being evident in the Salesforce Builder, there is a VERY big difference between the criteria in the Schedule Flow execution start, and an initial GET element in a Scheduled Flow that has no Object defined.

- Putting criteria in the Start Element has less conditions available, but effectively limits the scope of the Flow to only these records, which is great in **big environments**. It also fires **One Flow Interview per Record**, and then bulkifies operations at the end.

Choose Object and Filter Conditions

To have the scheduled flow run for a batch of records, specify the object and the conditions that each record must meet. A flow interview runs for each record in the batch. You can access and update the record's field values in the \$Record global variable.

Object

Account

Condition Requirements

All Conditions Are Met (AND)

Field

Search fields...



Operator

Select...



Value

Enter value or search resources...



+ Add Condition

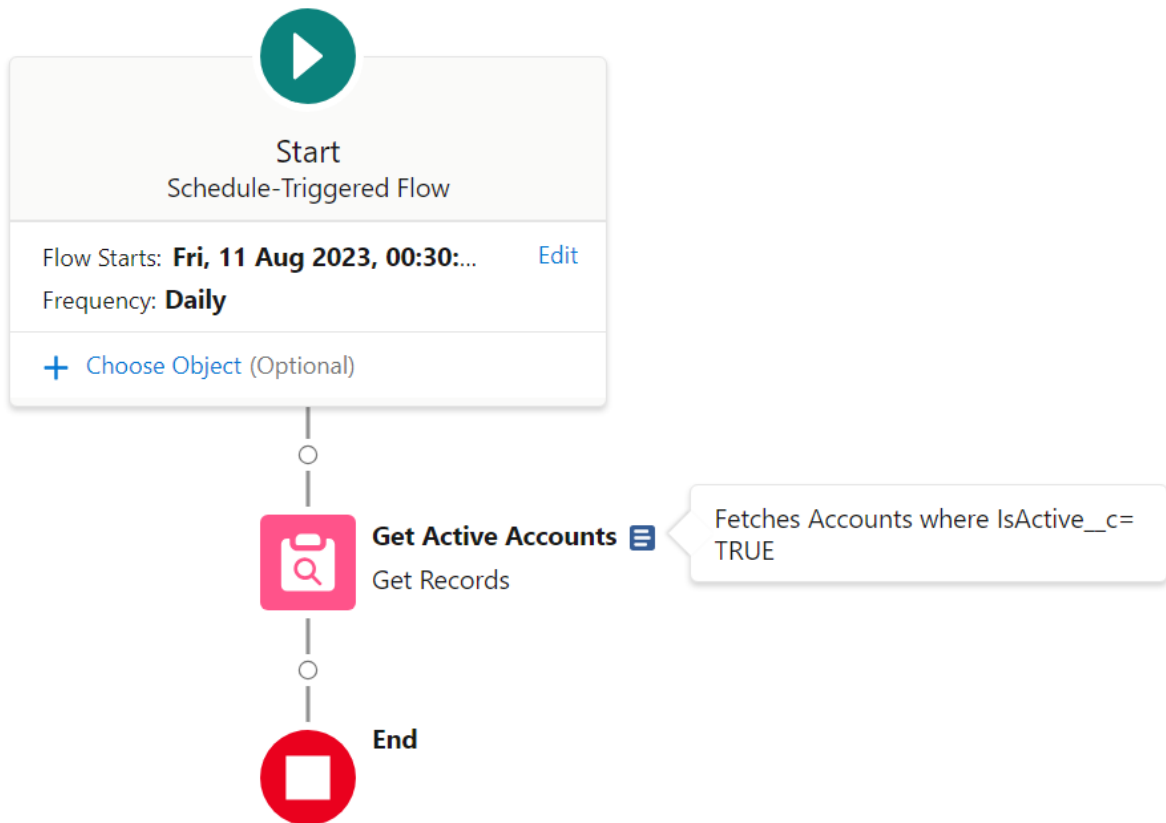
Cancel

Done

An often-done mistake is to do the above selection, say "Accounts where Active = TRUE" for example, and then doing a Get Records afterwards, querying the accounts again, because of habits tied to Record-Triggered Flows.

If you do this, you are effectively querying the entire list of Accounts X times, where X is the number of Accounts in your original criteria. Which is bad.

- On the opposite, putting no criteria and relying on an initial Get does a single Flow Interview, and so will run less effectively on huge amounts of records, *but* does allow you to handle more complex selection criteria.



In the first case, you should consider that there is only one record selected by the Flow, which is populated in `$Record` - much like in Record-Triggered Flows.

In the second screenshot, you can see that the Choose Object is empty, but the GET is done afterwards - `$Record` is as such empty, but the Get Active Accounts will generate a collection variable containing multiple accounts, which you will need to iterate over (via a `Loop` element) to handle the different cases