

Guide to Polling Triggers in Boost.space Integrator

<https://docs.boost.space/knowledge-base/system/features/polling-triggers/>

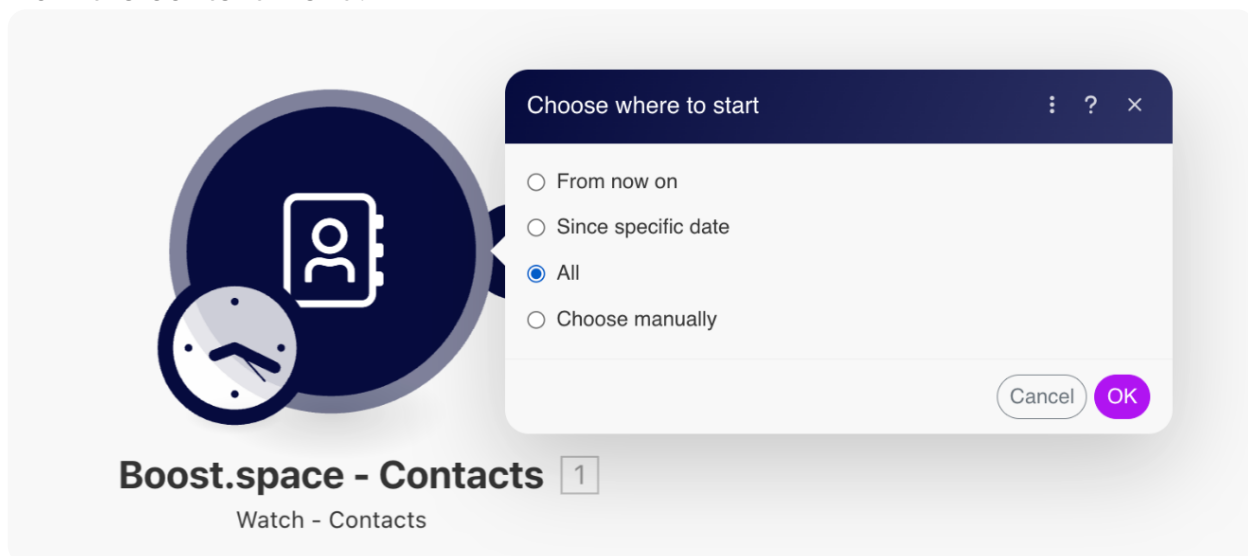
- [Docs](#)
- [Database](#)
- [Features](#)
- Polling triggers

Table of Contents

- [Understanding Polling Triggers in Boost.space Integrator](#)
 - [Use of these modules - Best practice](#)

Polling triggers are designed to regularly poll a given service whether there has been a change since their previous run. So you will typically schedule a scenario containing a polling trigger [module](#) to run periodically by selecting for example **"At regular intervals"** option from the **"Schedule setting panel"**. If there is a change the trigger will return bundles containing information about the change. If there is no change the trigger will output no bundles.

Polling triggers allow you to select the first bundle they should output via the epoch panel. The panel is displayed automatically after you save a trigger or when you make a substantial change in the trigger settings. You can also display the panel by right-clicking the module and choosing **"Choose where to start"** from the context menu.



Each [Boost.space](#) application in Boost.space Integrator has two polling trigger [modules](#). The first one reacts only to new [records](#) created in Boost.space. An example of this module is **"Watch - Contact Created"**



Watch - Contact Created ACID

Triggers when a Contact is created.

The second module responds to both new records created and changes in records. An example of this module is "**Watch - Contact**"



Watch - Contacts ACID

Triggers when a contact is created or updated.

Use of these modules – Best practice

These modules should be used for data synchronization. Inside the module you can set a Limit, which indicates how many records can be set in one run of the scenario. This way you can regulate the number of data processing and avoid scenario time outs. So for example when synchronizing 1000 records, I can set the limit to 100, set the [record](#) to start from (or if it should process all), set the scenario scheduling, and each run of the scenario will process 100 records and the next run will remember the last processed record and start working from there until it processes all 1000 records. After that it will just watch to see if a new record has been created or if a new record change has occurred and it will process only these records.

If you encounter any difficulties while going through this process do not hesitate to reach out to us at support@boost.space.

[Remote IDs Watch function in Boost.space – Eye symbol in modules/addons](#)