

How to conditionally suspend or terminate a synchronous job

000045995 | General Info | Last updated 2026-09-14 | TotalAgility

ISSUE

The customer creates synchronous TotalAgility jobs via `/services/sdk/v1/jobs/sync`. When a downstream step in the process (e.g., email, CCM/Customer Communications Management) fails, they need the `/jobs/sync` call itself to return a non-200 HTTP response, so the calling application can detect the failure and decide whether to roll back related actions.

SOLUTION

We would recommend implementing custom logic if they choose to use a synchronous process. A synchronous process is handled in a single thread, and only written to the database when the job is complete.

In our Best Practices Guide we advise the following with troubleshooting a Synchronous Process:

- If requiring a synchronous process as part of a solution, first create the process as asynchronous to ensure the behavior is correct. Once the synchronous process is created, do the following to debug the process:
- Check the event log for additional error information.
- Turn the Record history on to view the path taken through the job viewer and to see the final values of the variables.
- If you need to break this down further, add decisions or branching rules (XOR) with end nodes at the various breakpoints to view the variable values at that point.
- Once the problem is isolated, copy the node into a test process for further isolated testing

Considering the requirement, TS expertise suggested they run this process to be asynchronous, this will give much better control if a custom dependency or endpoint fails.

Here is a further suggestion. A C# Activity Exception can be used to force desirable error handling.

As Synchronous jobs run entirely in memory on the calling thread. If an activity throws an unhandled exception, the engine aborts the job execution and throws the exception straight out of the SDK call, resulting in an HTTP 500 status.

Setup:

- 1/ Catch the downstream failure (Email/CCM) via an Error decision/transition.

2/ Route to a C# / .NET Script Activity.

3/ Inside the script, throw an unhandled exception.

Result: /jobs/sync aborts execution and returns an HTTP 500 status code with the exception details in the response body.

(NOTE: The above is a high level suggestion, and Technical support cannot advise further on its design or implementation. For custom solutions, please reach out to Professional Services.)

REFERENCES

Customizations: Adding, Updating, Troubleshooting and how to contact Professional Services

<https://aio-eus-uat-cae-aif-app14-local.redglacier-35d7ee4f.eastus.azurecontainerapps.io/article/45995>