Four Pees

Switch

App

IdleAlert

Table of Contents

1. Introduction

It's very quiet today. You've seen no errors or problem alerts – all seems well with your workflow. Your prepress team are having a slow day too, with no preflight errors or manual processes to deal with – this must be the result of the hard work you have been putting in. Come to think of it, there wasn't a lot happening when you walked through the print room this morning. Perhaps the work has dried up, or there's been some apocalyptic event that you all missed somehow. It's strange but somewhat familiar – the last time this happened it was because…

[phone rings ominously]

It turns out that the upstream process which delivers jobs to your Switch flow has failed again. Jobs have been piling up since last night waiting to be processed. If only there was some way that you could have been warned when there were no new jobs for over an hour!

IdleAlert will check if any job has passed through it within a chosen time period. If not, it will submit a text file into your flow containing information on the flow, flow element, and the last job that arrived.

There are additional options for handling business hours, days of the week and date exceptions (such as public holidays) so that you only receive alerts when you want them.

The resulting text file could be used to trigger an email or Slack notification so that you can be notified in your preferred way when there is no activity. It could even be used to trigger an API call or database update so that inactivity can be logged and used to generate useful reports.
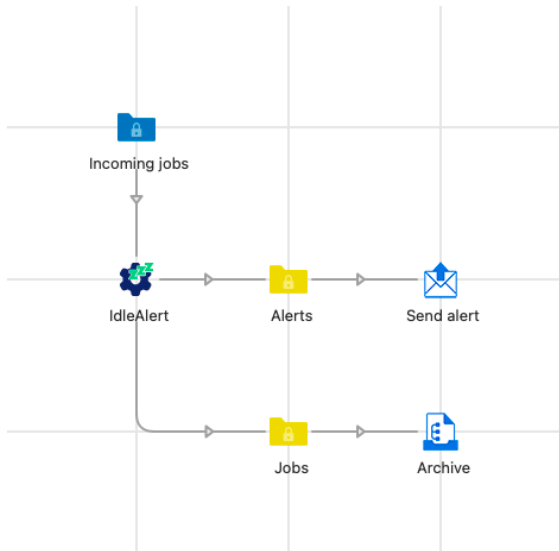
1.1. Versions

The following is a short version overview:

• version 1: initial version of the app.

2. How to use

IdleAlert can be placed anywhere in your flow. Outgoing connections can be configured to route incoming jobs or alert jobs, so you can filter the alerts and direct them to a *Mail send* element for instance.

2.1. Info available in private data

The content of the text file is also available as private data so that you can access individual elements within your flow

idleAlert.LastJobArrived	The timestamp of the last job to pass through the flow element
idleAlert.LastJobName	The name of the last job to pass through the flow element
idleAlert.Flow	The name of the flow in which the flow element resides
idleAlert.FlowElement	The name of the flow element
idleAlert.idleSeconds	The time elapsed since the last job arrived (in seconds)
idleAlert.idleMinutes	The time elapsed since the last job arrived (in minutes)
idleAlert.idleHours	The time elapsed since the last job arrived (in hours)
idleAlert.message	A short message, which could be used for an email subject, e.g: *"No jobs have arrived in flow 'My Awesome Flow' at element 'IdleAlert' for 30 mins"*
idleAlert.fullText	The full content of the text file, which could be used for an email body e.g: *"Last job arrived: 2024-09-20T14:18:20.409* *Last job name: Penguins.pdf* *Flow: My Awesome Flow* *Flow element: IdleAlert* *Idle time: 00:30:10"*

3. Properties

The following properties can be used to change the behaviour of the app.

3.1. Inactivity monitoring time periods

- **Time unit**: choose from *Hours*, *Minutes* or *Seconds*; this is the unit that applies to the time values defined in the app's properties.

- **Day mode**: choose from *Every day*, *Weekday/weekend,* or *Day of week*; this alters how you can define the time periods in which the app will check for inactivity.

- **Out-of-hours time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*) for reporting inactivity outside of business hours (if configured); select *None* to disable reporting outside of these hours

  If **Day mode** is set to *Every day*:

  - **Time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*); this will determine the maximum wait time between jobs, above which the app will report inactivity

  - **Business hours**: set to *Yes* to enable the *Start time*, *End time* and *Time period* properties

    - **Start time**: enter a time to start checking for inactivity

    - **End time**: enter a time to start checking for inactivity

    - **Out-of-hours time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*) for reporting inactivity outside of business hours; select *None* to disable reporting outside these hours

  If **Day mode** is set to *Weekday/weekend*:

  - **Time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*); this will determine the maximum wait time between jobs for weekdays, above which the app will report inactivity

  - **Business hours**: set to *Yes* to enable the *Start time*, *End time* and *Time period* properties

    - **Start time**: enter a time to start checking for inactivity

    - **End time**: enter a time to start checking for inactivity

  - **Out-of-hours time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*) for reporting inactivity outside of business hours (including weekends); select *None* to disable reporting outside these hours

  If **Day mode** is set to *Day of week*:

  - **Monday time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*); this will determine the maximum wait time between jobs for Mondays, above which the app will report inactivity

  - **Monday business hours**: set to *Yes* to enable the *Start time*, *End time* and *Time period* properties

    - **Start time**: enter a time to start checking for inactivity on Mondays

    - **End time**: enter a time to start checking for inactivity on Mondays

  For the remaining 6 days of the week, the properties are all identical to Monday's properties, i.e.…

  - **Tuesday time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*); this will determine the maximum wait time between jobs for Tuesdays, above which the app will report inactivity

  - **Tuesday business hours**: set to *Yes* to enable the *Start time*, *End time* and *Time period* properties

    - **Start time**: enter a time to start checking for inactivity on Tuesdays

    - **End time**: enter a time to start checking for inactivity on Tuesdays

  …etc.

3.2. Defining exception dates

- **Exception dates**: optionally enter a list of dates (one per line) in the format yyyy-MM-dd to define days (for example public holidays) on which a separate time window should be used; select *None* to disable this feature

- **Exception time period**: enter a number (in hours, minutes or seconds, depending on the selected *Time unit*) to determine the maximum wait time between jobs for exception days, above which the app will report inactivity; if set to *None* no reports will be generated on these days

3.3. Routing jobs (outgoing connections)

- **Route incoming jobs**: whether to send incoming jobs to this connection (only one connection is allowed to have this property enabled); you do not need to enable this for any connections if you don't want to keep the original job

- **Route alerts**: whether to send inactivity alerts to this connection; one, and only one, connection must have this property enabled

4. More information

4.1. Four Pees ━ feel the good flow

This app was created by Four Pees. You can find more information about our company here: http://www.fourpees.com. We created this app based on the experience we have with projects where Switch is used, but of course that is not a guarantee that the app will be suitable for every project out there.

If you run into a problem, or this app doesn't completely cover what you had hoped it would, don't hesitate to send us feedback. There are multiple ways you can do this:

• Go to our website and use the contact page: https://www.fourpees.com/en/contact.

• Send us an email at support@fourpees.com. You'll get a confirmation message and we'll be with you before you can say "Automation".

4.2. Free apps

If you're using one of our free apps, please keep in mind that our support on those is limited. We believe this is fair as free apps can't be handled the same way as payable project work.

That having been said, we of course will try to help you as best as we can! Just get in contact and we'll have a conversation on how we can help you.