POWERED BY DUG

DYNAMICSCON LIVE

MAY 2024

live.dynamicscon.com

Error Proof Your Automations: a Deep Dive into Advanced Error Handling and Logging in Power Automate Desktop

Agnius Bartninkas, COO and Co-Founder @Definra

POWERED BY DUG
DYNAMICSCON
LIVE
MAY 2024

Let me introduce myself!

Microsoft Most Valuable Professional

- Microsoft Business Aplications MVP (Power Automate)
- Used to be the Excel guru of my office, but now am the grumpy guy telling everyone to stop using Excel.
- The most experienced PAD (formerly Softomotive) user in Lithuania. Most other users were trained by me.
- Top all time solution author in the PP community forums for PAD.
- A “certified” beer expert. Now an aspiring cocktail maker.

Agenda for today

Some “theory”:

- Types of error handling available in PAD
- Options available for logging errors

Demos:

- Different ways to handle errors
- The different options for logging error messages
- Logging runtime non-error messages
- Taking screenshots on error
- Monitoring logs in ELK

Q&A

Why error handling

Errors can and **will** occur in your flows.

There’s no official template available for handing it.

Open-sourced **PADFramework** available here: https://github.com/AgniusBartninkas/power-automate-desktop-framework

Types of Error Handling

The main options on how to handle errors in PAD are:

- **None** – the flow fails on any error (not recommended)
- **Action level** – special rules set on each action (not recommended, except for special cases)
- **Error blocks** – setting rules for blocks of actions or entire sub-flows (recommended)

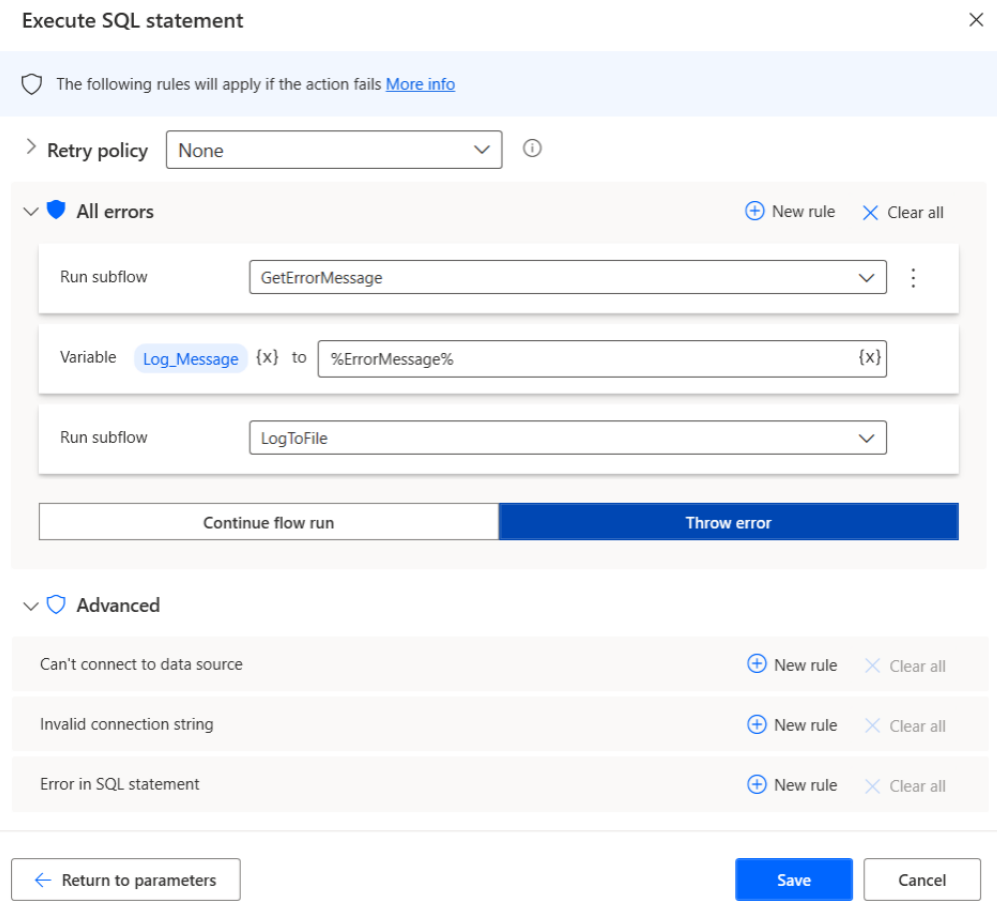
Action Level Error Handling

+ Very powerful, lots of rules

- Time-consuming to build and maintain

- Does not apply to all actions

- Cannot handle unexpected logic errors



Execute SQL statement

The following rules will apply if the action fails More info

Retry policy: None

All errors

- Run subflow: GetErrorMessage
- Variable Log_Message {x} to %ErrorMessage%
- Run subflow: LogToFile

Continue flow run | Throw error

Advanced

- Can't connect to data source
- Invalid connection string
- Error in SQL statement

POWERED BY DUG DYNAMICSCON LIVE MAY 2024

Error Blocks

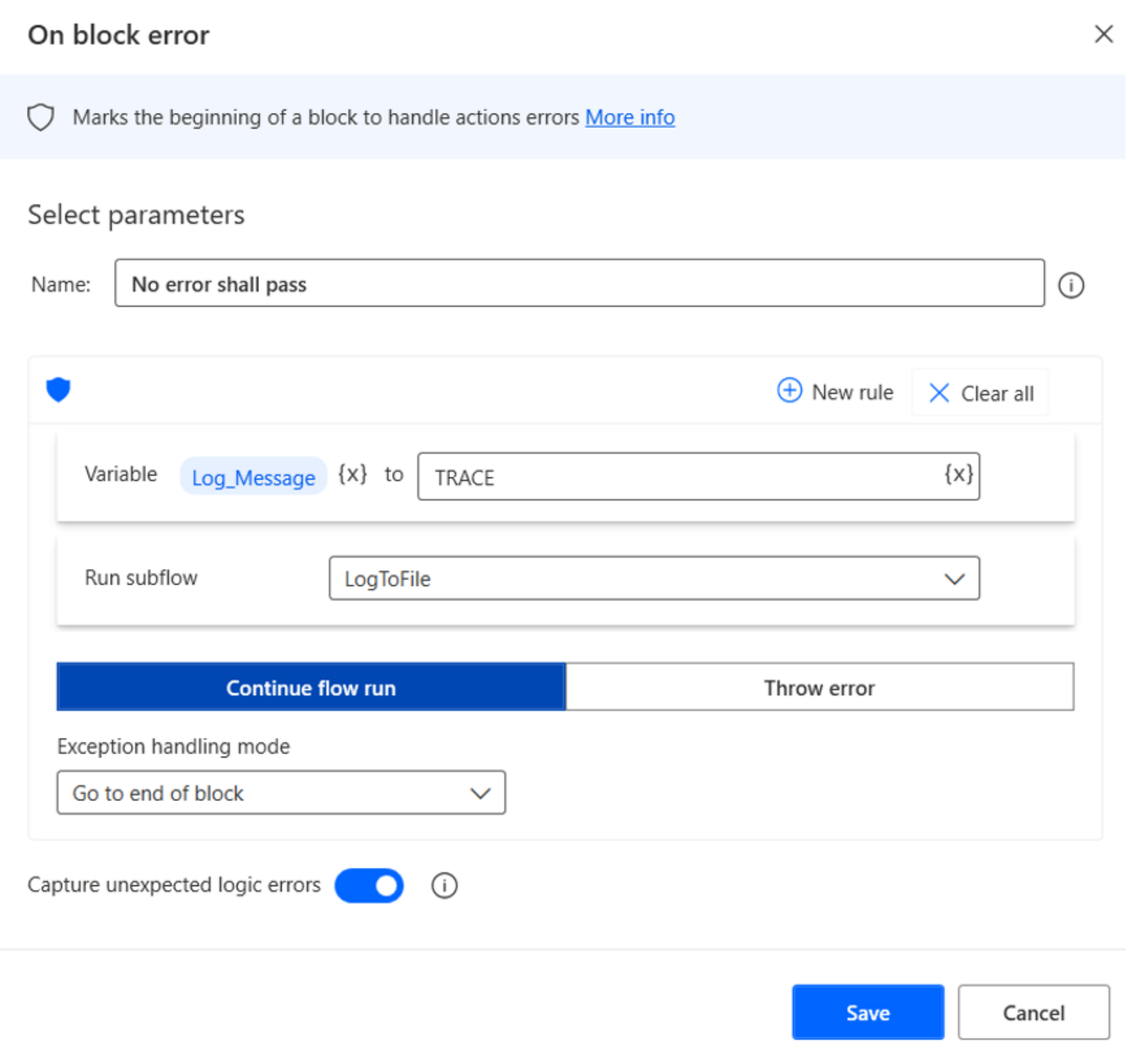
+ Very poweful

+ Easy to setup and maintain

+ Can handle all exceptions

-Less options

-Same rules apply to all actions



On block error

Marks the beginning of a block to handle actions errors More info

Select parameters

Name: No error shall pass

Variable Log_Message {x} to TRACE {x}

Run subflow: LogToFile

Exception handling mode: Go to end of block

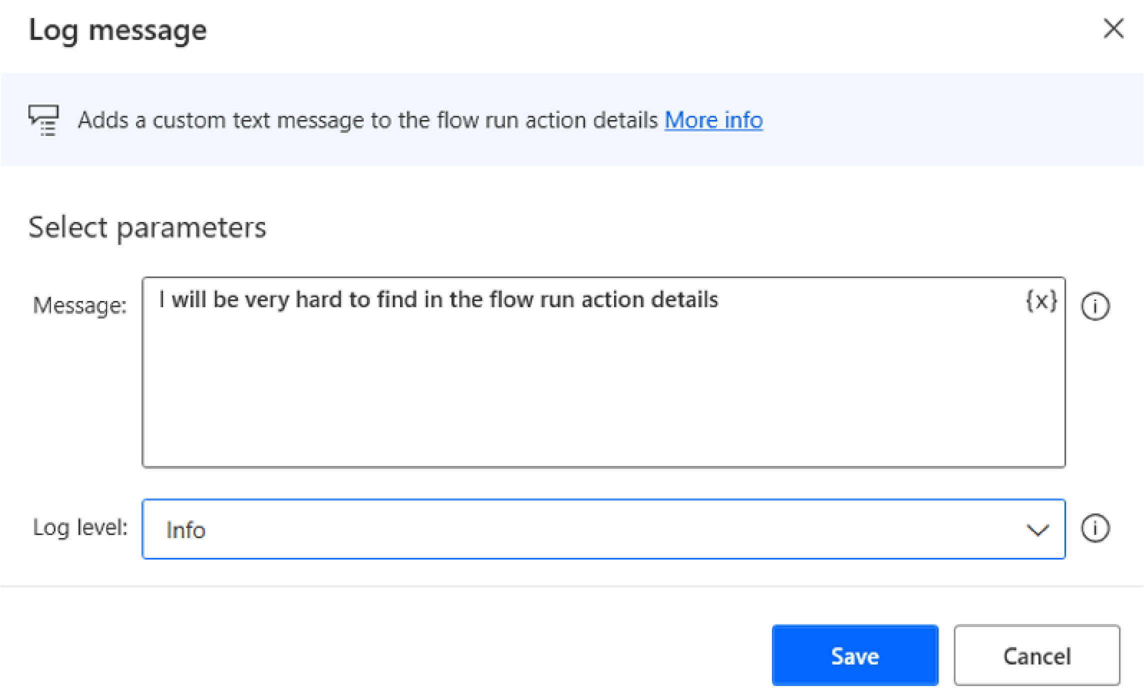
Capture unexpected logic errors: ON

Logging Errors

Handling errors is cool, but without logs, it will usually be quite hard to find the issue.

Currently, PAD supports the **Log message** action.

While **Log message** is better than no logging, any custom logging alternative is better than using Log message.



Log message

Adds a custom text message to the flow run action details More info

Select parameters

Message: I will be very hard to find in the flow run action details

Log level: Info

Better Ways to Log Errors

Some options that I’ve seen:

- **Logging to databases** – great for techies, might be harder to use for non-tech people.
- **Logging to SharePoint lists** – better for non-tech people
- **Logging to CSV files** – easy to implement, but a bit limited
- **Logging to plain text files** – very easy to implement and easy to use

Logging to Plain Text Files

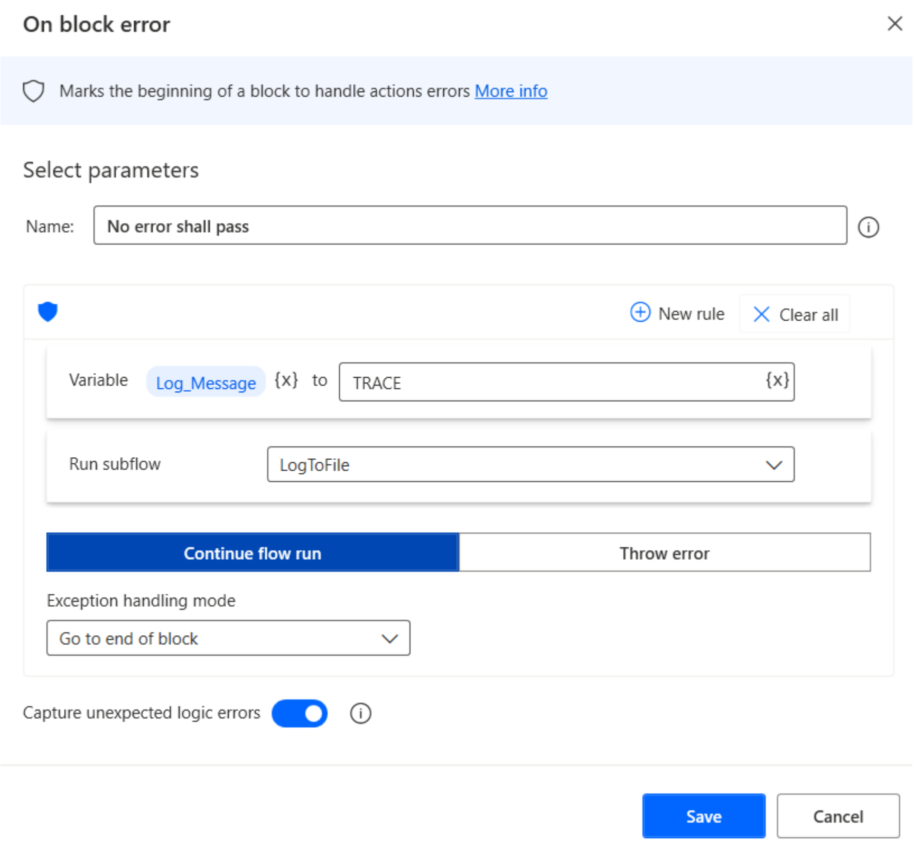
The reason I like plain text files:

- They're very easy to set up in PAD
- Write text to file is much less likely to fail due to application errors.

The one limitation of logging to local files over databases and SharePoint lists are the fact that they're local to the machine (unless we push them to some network location).

DEMO TIME!

On Block Error at Sub-Flow Level + Get Last Error



On block error

Marks the beginning of a block to handle actions errors More info

Select parameters

Name: No error shall pass

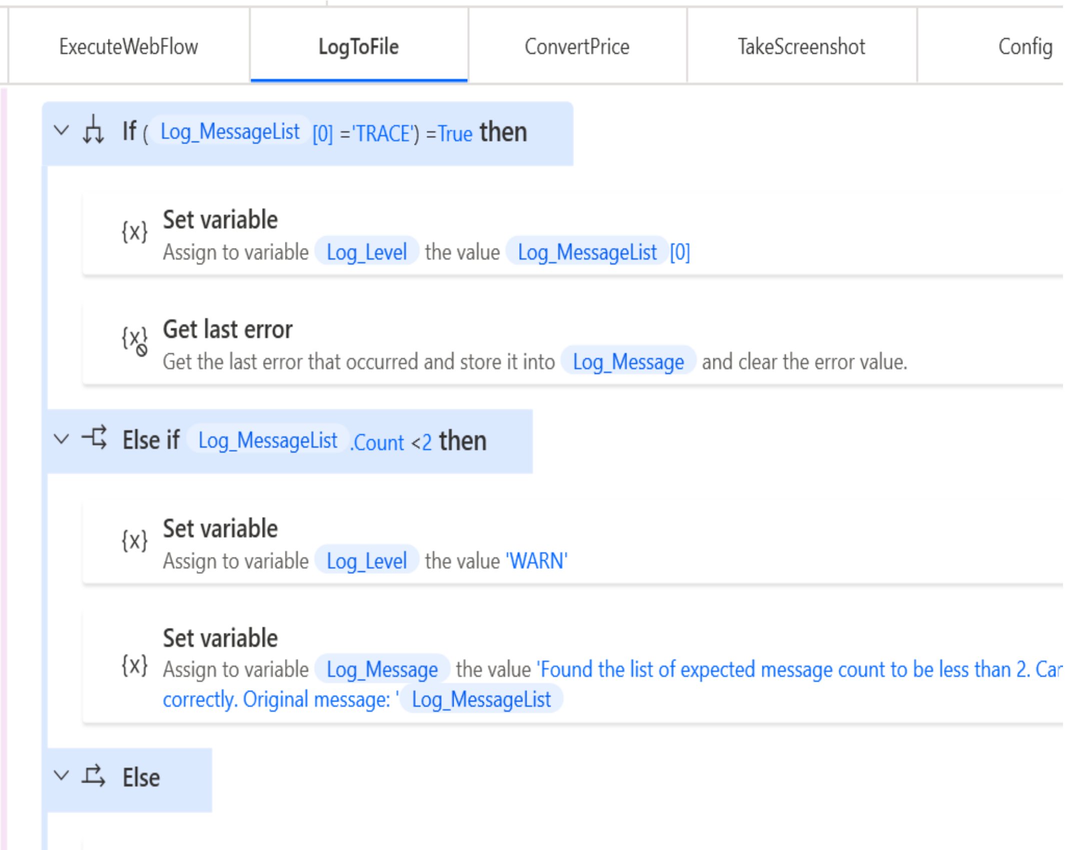
Variable Log_Message {x} to TRACE

Run subflow LogToFile

● Continue flow run ○ Throw error

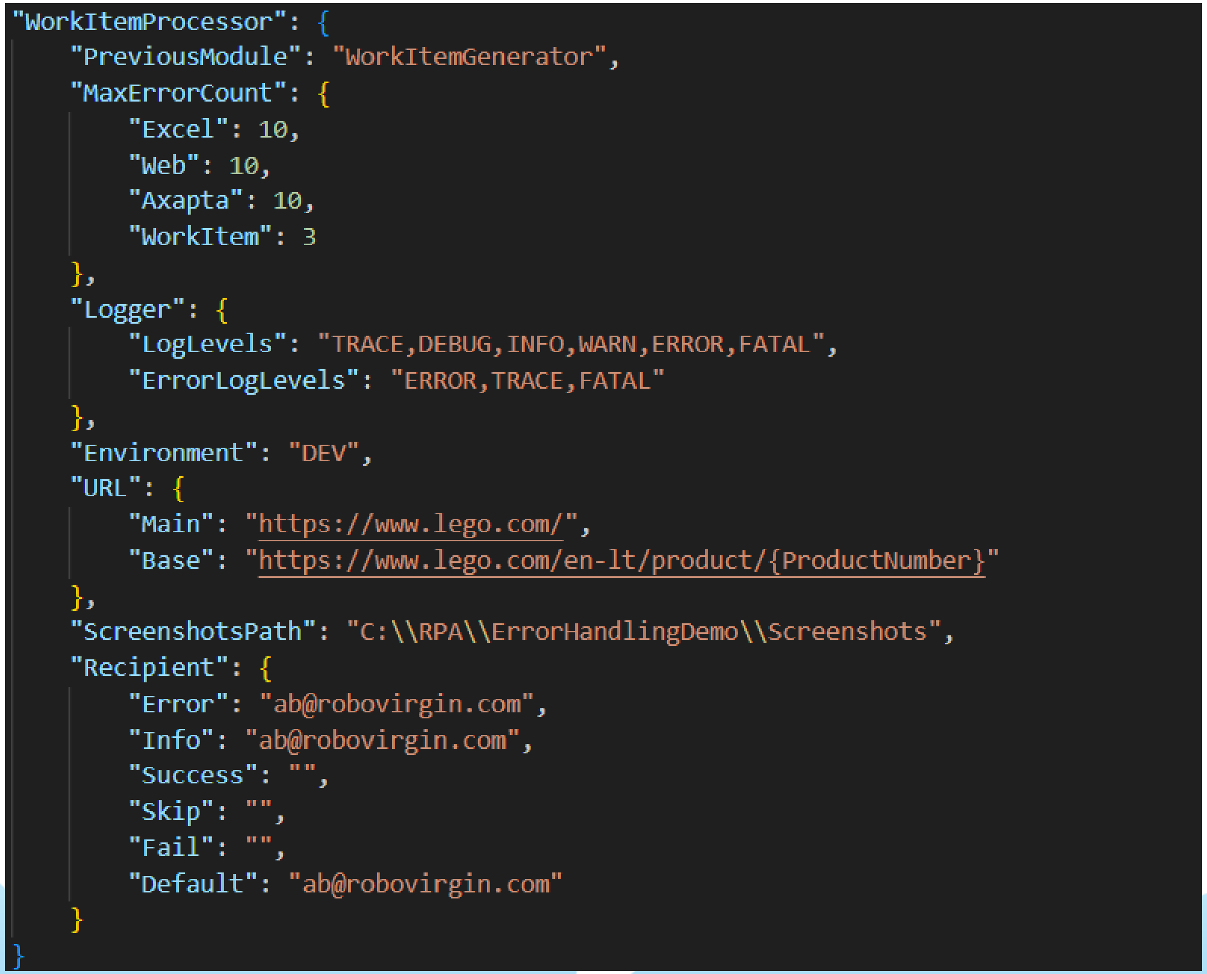
Exception handling mode: Go to end of block

Capture unexpected logic errors: On



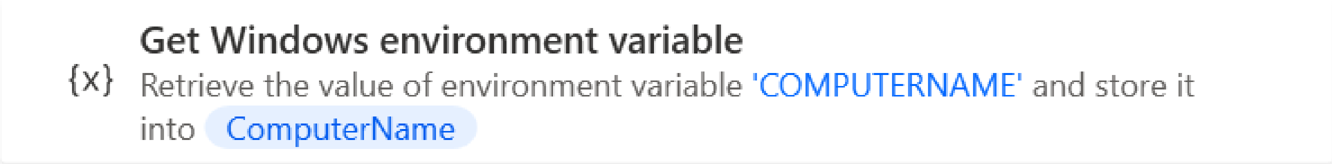
```
If ( Log_MessageList [0] ='TRACE') =True then
    Set variable
    Assign to variable Log_Level the value Log_MessageList [0]
    Get last error
    Get the last error that occurred and store it into Log_Message and clear the error value.
Else if Log_MessageList .Count <2 then
    Set variable
    Assign to variable Log_Level the value 'WARN'
    Set variable
    Assign to variable Log_Message the value 'Found the list of expected message count to be less than 2. Car
    correctly. Original message: ' Log_MessageList
Else
```

Logger Object + Screenshots Path in External Config

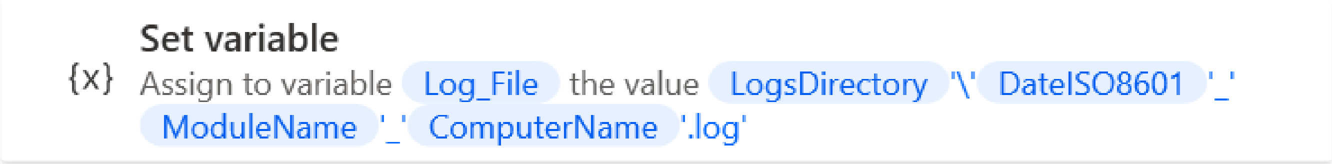


```
"WorkItemProcessor": {
    "PreviousModule": "WorkItemGenerator",
    "MaxErrorCount": {
        "Excel": 10,
        "Web": 10,
        "Axapta": 10,
        "WorkItem": 3
    },
    "Logger": {
        "LogLevels": "TRACE,DEBUG,INFO,WARN,ERROR,FATAL",
        "ErrorLogLevels": "ERROR,TRACE,FATAL"
    },
    "Environment": "DEV",
    "URL": {
        "Main": "https://www.lego.com/",
        "Base": "https://www.lego.com/en-lt/product/{ProductNumber}"
    },
    "ScreenshotsPath": "C:\\RPA\\ErrorHandlingDemo\\Screenshots",
    "Recipient": {
        "Error": "ab@robovirgin.com",
        "Info": "ab@robovirgin.com",
        "Success": "",
        "Skip": "",
        "Fail": "",
        "Default": "ab@robovirgin.com"
    }
}
```

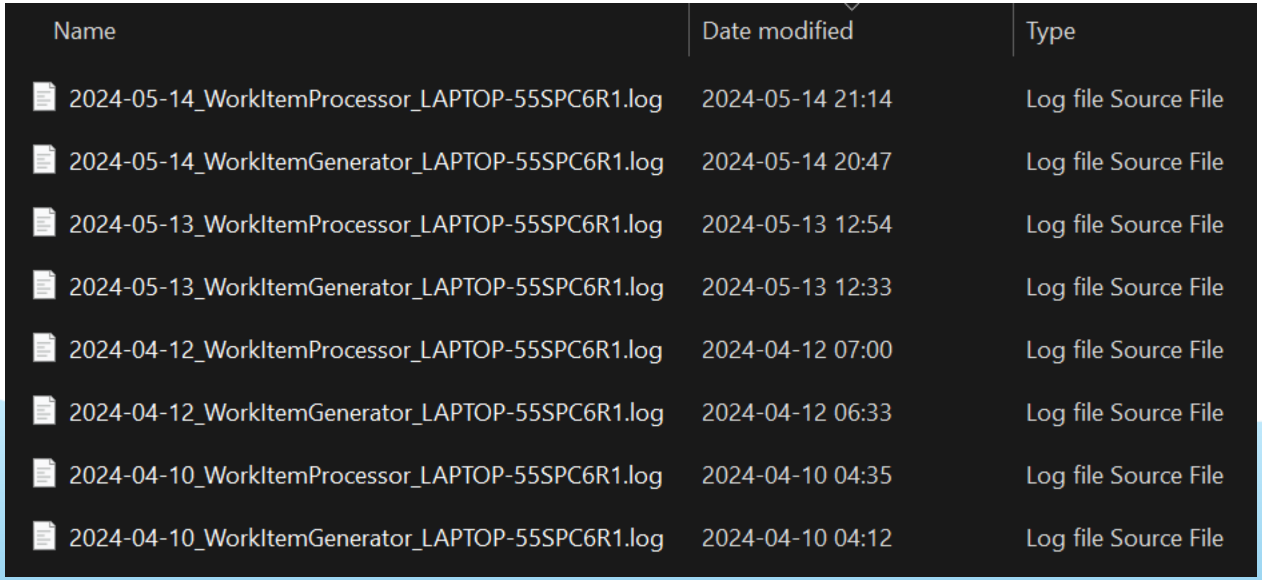
Log File per Day per Flow (+ per Machine Optionally)



{x} **Get Windows environment variable**
Retrieve the value of environment variable 'COMPUTERNAME' and store it into ComputerName



{x} **Set variable**
Assign to variable Log_File the value LogsDirectory '\' DateISO8601 '_' ModuleName '_' ComputerName '.log'



Name	Date modified	Type
2024-05-14_WorkItemProcessor_LAPTOP-55SPC6R1.log	2024-05-14 21:14	Log file Source File
2024-05-14_WorkItemGenerator_LAPTOP-55SPC6R1.log	2024-05-14 20:47	Log file Source File
2024-05-13_WorkItemProcessor_LAPTOP-55SPC6R1.log	2024-05-13 12:54	Log file Source File
2024-05-13_WorkItemGenerator_LAPTOP-55SPC6R1.log	2024-05-13 12:33	Log file Source File
2024-04-12_WorkItemProcessor_LAPTOP-55SPC6R1.log	2024-04-12 07:00	Log file Source File
2024-04-12_WorkItemGenerator_LAPTOP-55SPC6R1.log	2024-04-12 06:33	Log file Source File
2024-04-10_WorkItemProcessor_LAPTOP-55SPC6R1.log	2024-04-10 04:35	Log file Source File
2024-04-10_WorkItemGenerator_LAPTOP-55SPC6R1.log	2024-04-10 04:12	Log file Source File

Thank you!

Let’s connect on Linkedin!

Please rate my session

www.linkedin.com/in/agnius-bartninkas