

RPA Design and Development

v4.0





Lesson 8 | Logging

Logging – Exam Topics

1. Describe and interpret robot execution logs
2. Apply logging best practices during development

Logging is the process of keeping logs of various events that occur during project execution. There are four types of logs:

Studio Logs

- Diagnostic log messages generated by Studio regarding its behavior

Orchestrator Logs

- Diagnostic log messages generated by Orchestrator regarding its behavior

Robot Logs

- Diagnostic log messages providing information related to the Robot and its context

Setup Logs

- Track events related to installing UiPath Studio (logged in the Event Viewer)

There are two types of Robot logs::

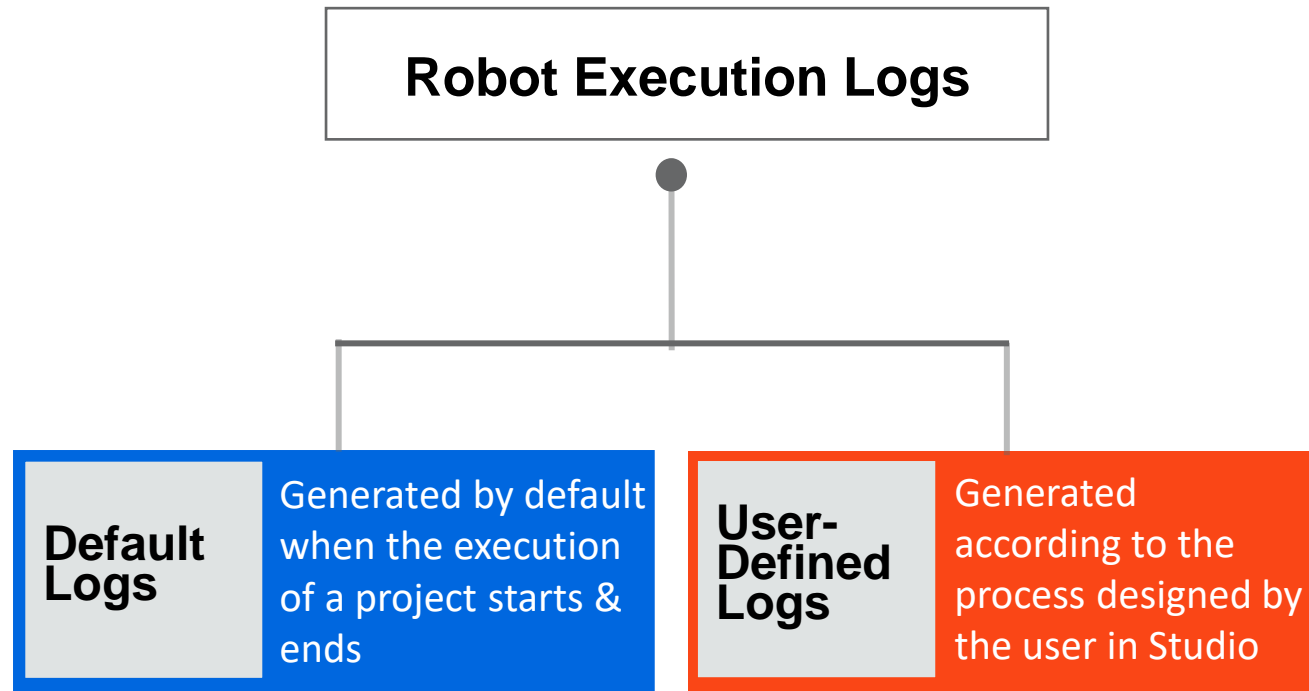
Robot Execution Logs

These are messages generated by the execution of a process and contain information related to its behavior and user-defined messages.

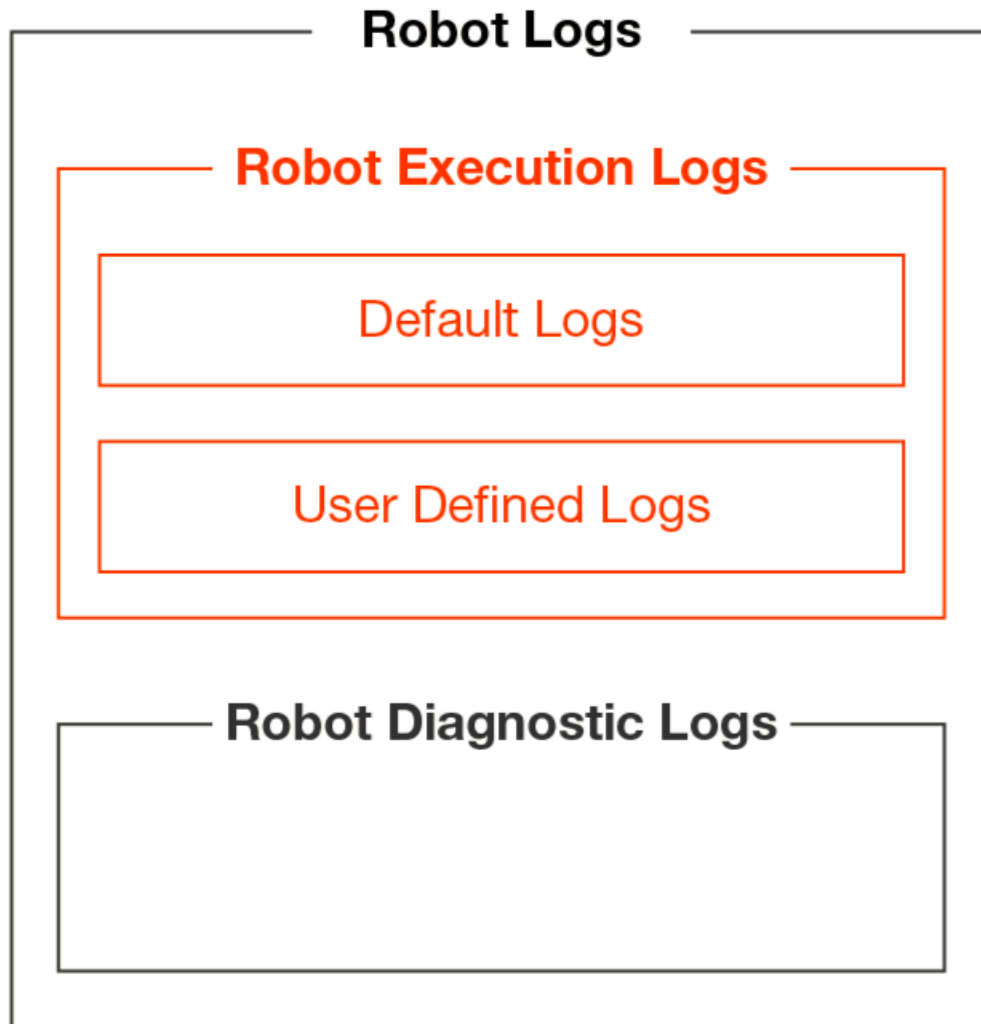
Robot Diagnostics Logs

These provide information related to the Robot itself and its context.

The types of logs generated while running processes in Studio are a subset of Robot logs, called Robot Execution logs. Robot Execution Logs can be of two types: **Default Logs & User-Defined Logs**



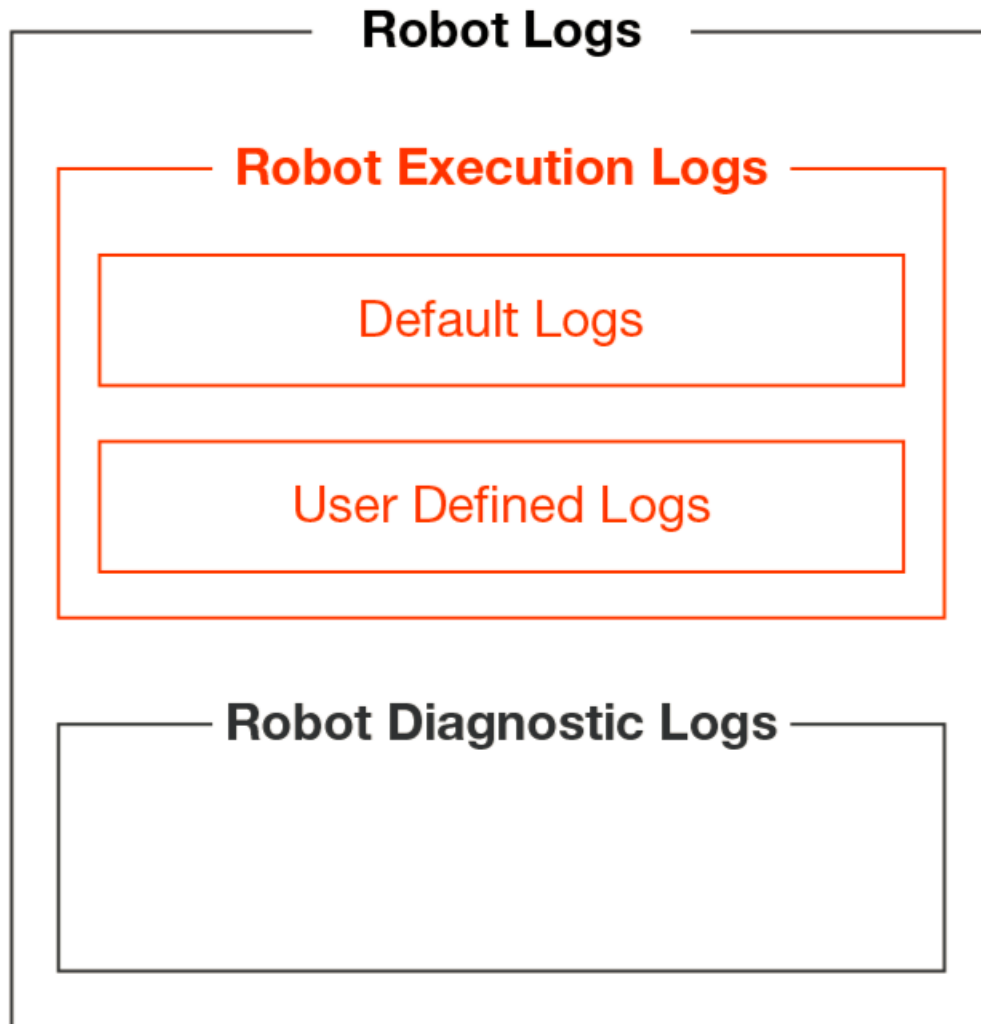
Logging – Default Logs



Default logs are generated automatically when certain events take place. These events can be the start/end of a process or a transaction, encountering an error, or when the Robot Logging Setting is set to Verbose.

- **Execution start** is generated every time a process is started (Level = Information).
- **Execution end** is generated every time a process is finalized (Level = Information).
- **Transaction start** is generated every time a transaction within a process is started (Level = Information).
- **Transaction end** is generated every time a transaction within a process is finalized (Level = Information).
- **Error log** is generated every time the execution encounters an error and stops (Level = Error).
- **Debugging log** is generated if the Robot Logging Setting is set to Verbose, and contains, Activity names, types, variable values, arguments, etc. (Level = Trace).

Logging – User Defined Logs



User-defined logs are generated according to the process designed by you in Studio when using

- **Log Message Activity** or
- **Write Line Activity**

Logging Levels

There are six levels of logging:



Critical Level

Indicates critical issues wherein a robot may not recover and stop working



Error Level

Indicates errors after which a robot retries for recovery and moves on



Warn Level

Indicates events that may have an adverse impact on a robot's performance



Info Level

To know the progress of a robot at each stage of execution



Trace Level

To collect information for developing or debugging

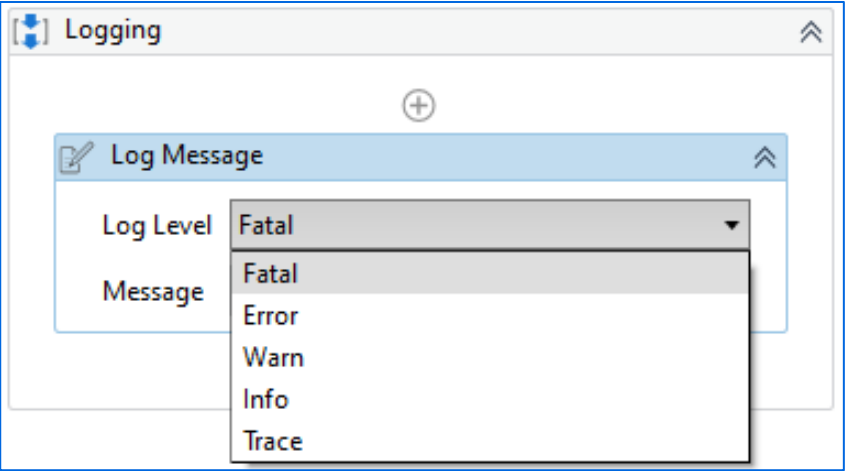


Verbose Level

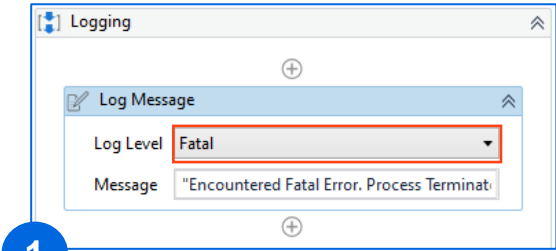
To log message at activity start and end, plus values of the variables and arguments

How to do Logging?

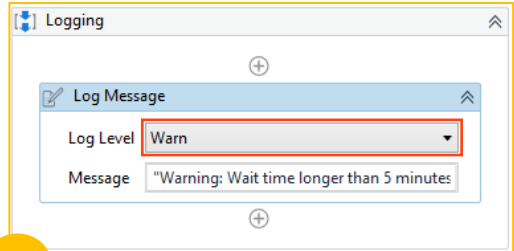
Logs are applied using the Log Message activity.



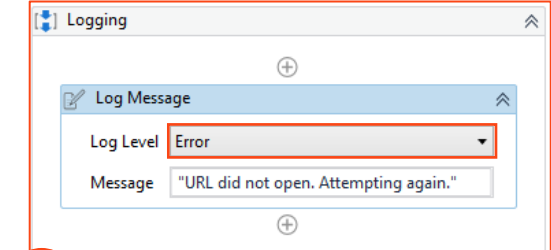
Log Message Activity



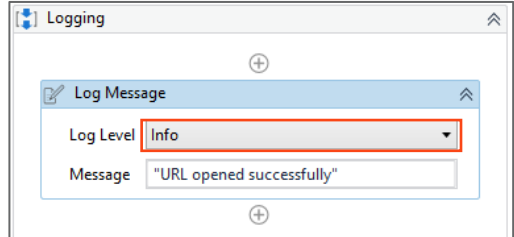
1 Fatal (Critical) Log Message



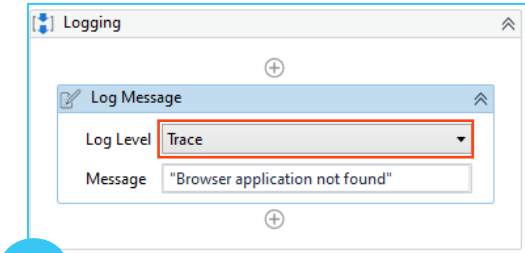
3 Warn Log Message



2 Error Log Message



4 Info Log Message



5 Trace Log Message



Topic	Link
Types of Logs	https://docs.uipath.com/studio/standalone/2022.10/user-guide/types-of-logs
Logging Levels Learn more about the logging levels in UiPath, the log message types, and log fields.	https://docs.uipath.com/studio/standalone/2022.10/user-guide/logging-levels
Robot Logs Learn more about the robot diagnostics and robot execution logs.	https://docs.uipath.com/robot/standalone/2022.10/user-guide/robot-logs#robot-execution-logs

Chapter 12:

Introduction to Logging in Studio

Accessing Local Logs



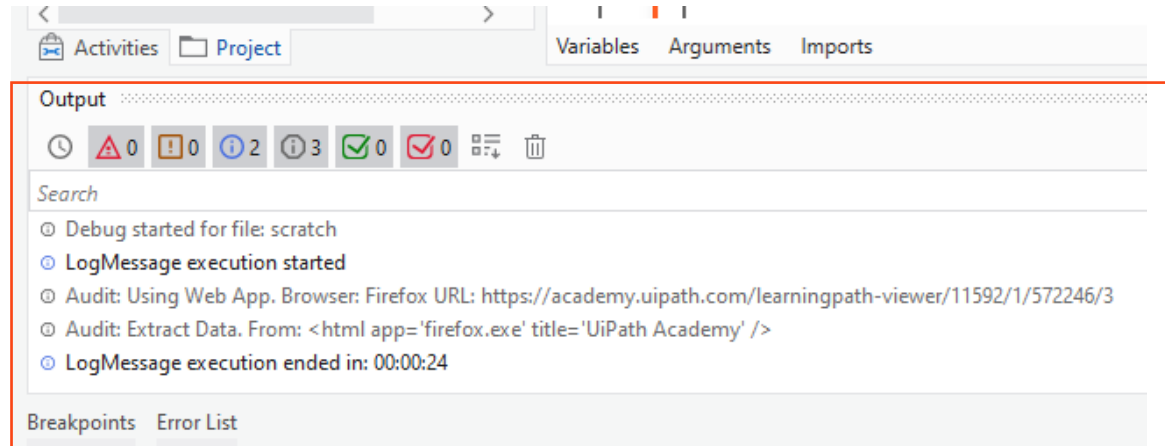
Accessing and Reading Robot Execution Logs



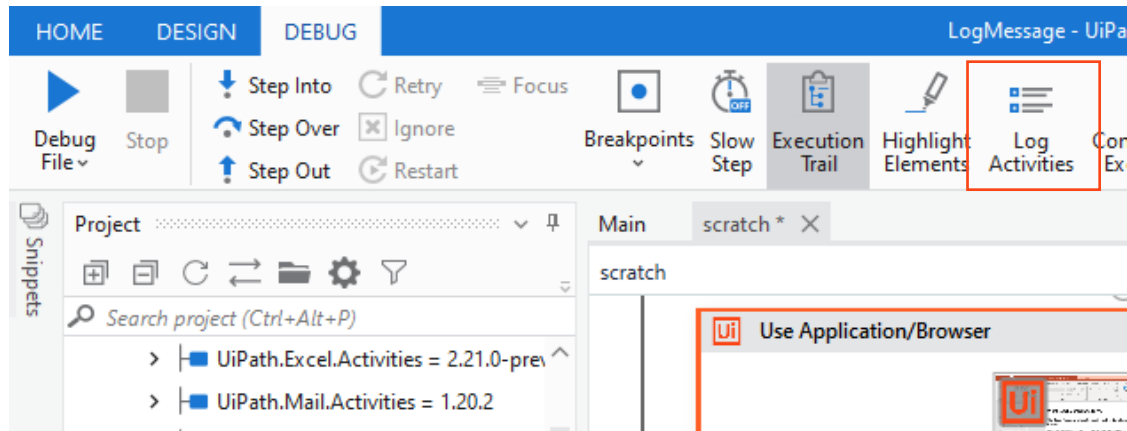
There are several places where you can access Robot Execution Logs:

Logs Location	Details
Output Panel in UiPath Studio	In the Output Panel in UiPath Studio for the previous process execution from Studio
%localappdata%\UiPath\Logs\<shortdate>_Execution.log file	For all processes ran on the machine from UiPath Studio. Logs are generated at Trace level and above or Verbose level and above depending on whether the Verbose level is activated or not
%localappdata%\UiPath\Logs\<shortdate>_Execution.log file	For all processes ran on the machine from UiPath Assistant. The logs are generated at the level defined in UiPath Assistant and above
Orchestrator	in the Logs section when running processes while connected to Orchestrator. The logs are generated at the defined level and above

Accessing Local Logs



View the logs generated by process execution, in the **UiPath Studio Output Panel**



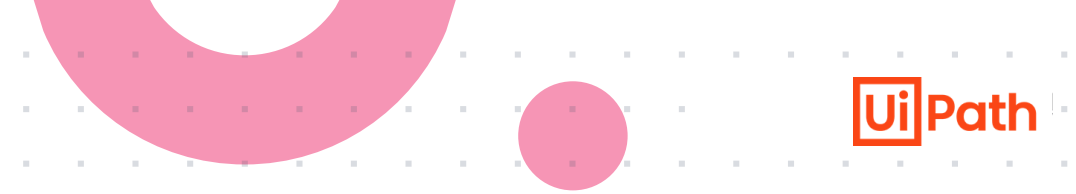
Access local Robot Execution log files from **Debug Ribbon -> Open Logs option**

The Anatomy of a Log Entry

Structure of log entries. These entries can be very helpful when checking execution logs to analyze behavior or figure out what caused an exception. Logs are in the form of a JSON, pretty much just a key-value pair ("field1:value1," "field2:value2").

The **default log fields** are present in all logs:

Key	Description
Message	The log message
Level	The log severity of the log message
Timestamp	The exact date and time the action was performed
FileName	The name of the .xaml file being "executed"
JobId	The key of the job running the process
ProcessName	The name of the process that triggered the logging
ProcessVersion	The version number of the process
WindowsIdentity	The name of the user that performed the action that was logged
RobotName	The name of the robot (as defined in Orchestrator)



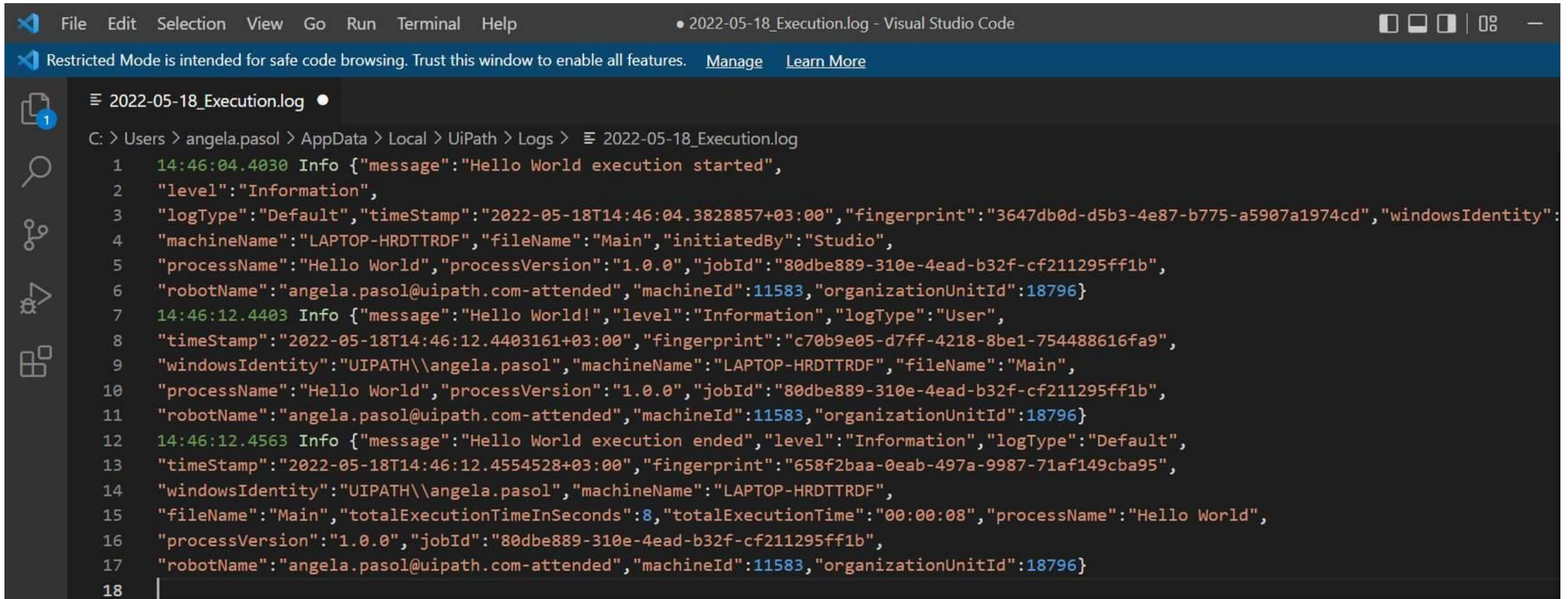
The Anatomy of a Log Entry

Aside from default fields, logs can also contain **type-specific fields** and **user-defined fields**.

- ❖ **Type-specific fields** are present depending on the log type, like `totalExecutionTimeInSeconds` and `totalExecutionTime` for Execution End.
- ❖ **User-defined fields** are defined in Studio (by using the Add Log Fields Activity) and appear in all resulting logs after the activity is generated unless they are (programmatically) removed by the activity Remove Log Fields.

The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode

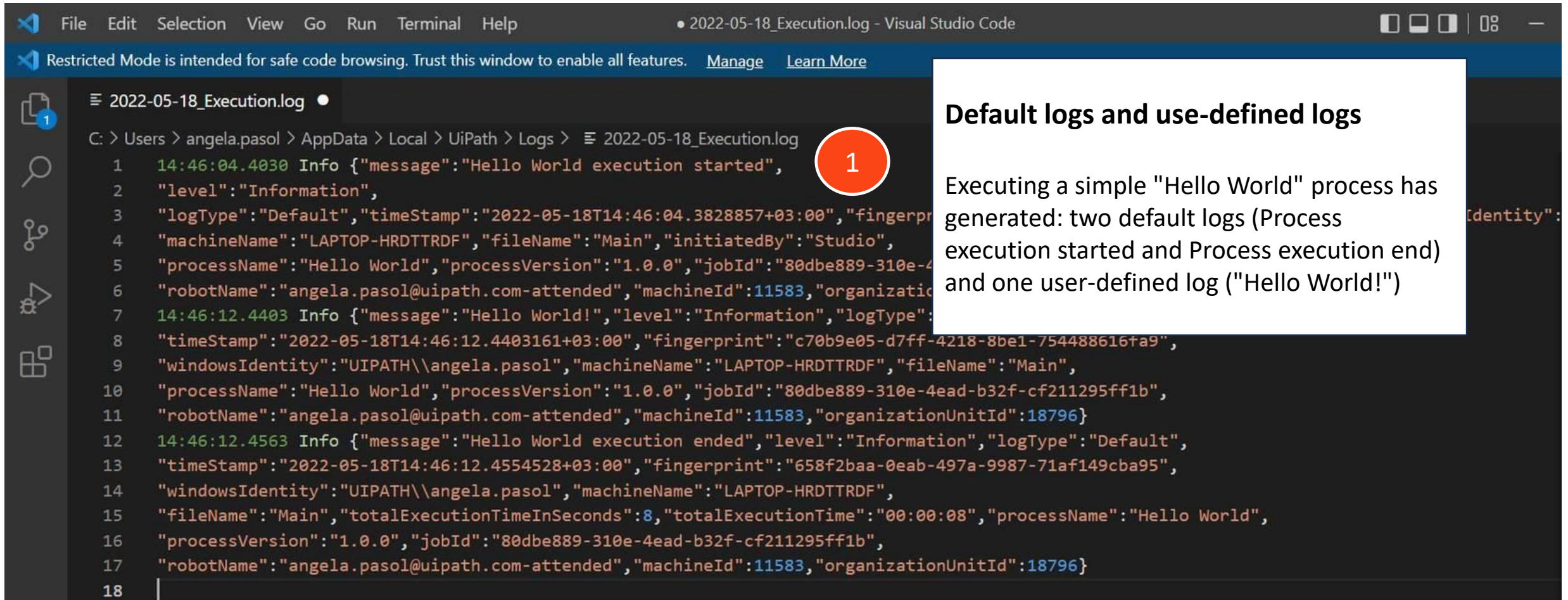


```
File Edit Selection View Go Run Terminal Help • 2022-05-18_Execution.log - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
1 14:46:04.4030 Info {"message":"Hello World execution started",
2  "level":"Information",
3  "logType":"Default","timeStamp":"2022-05-18T14:46:04.3828857+03:00","fingerprint":"3647db0d-d5b3-4e87-b775-a5907a1974cd","windowsIdentity":
4  "machineName":"LAPTOP-HRDTTRDF","fileName":"Main","initiatedBy":"Studio",
5  "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
6  "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
7 14:46:12.4403 Info {"message":"Hello World!","level":"Information","logType":"User",
8  "timeStamp":"2022-05-18T14:46:12.4403161+03:00","fingerprint":"c70b9e05-d7ff-4218-8be1-754488616fa9",
9  "windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF","fileName":"Main",
10 "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
11 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
12 14:46:12.4563 Info {"message":"Hello World execution ended","level":"Information","logType":"Default",
13 "timeStamp":"2022-05-18T14:46:12.4554528+03:00","fingerprint":"658f2baa-0eab-497a-9987-71af149cba95",
14 "windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF",
15 "fileName":"Main","totalExecutionTimeInSeconds":8,"totalExecutionTime":"00:00:08","processName":"Hello World",
16 "processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
17 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
18
```

The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode



The screenshot shows the Visual Studio Code interface with a log file open. The log file path is `C:\Users\angela.pasol\AppData\Local\UiPath\Logs\2022-05-18_Execution.log`. The log contains three entries. The first entry is an information log with the message "Hello World execution started". The second entry is an information log with the message "Hello World!". The third entry is an information log with the message "Hello World execution ended". A callout box titled "Default logs and use-defined logs" explains that executing a simple "Hello World" process generates two default logs (Process execution started and Process execution end) and one user-defined log ("Hello World!").

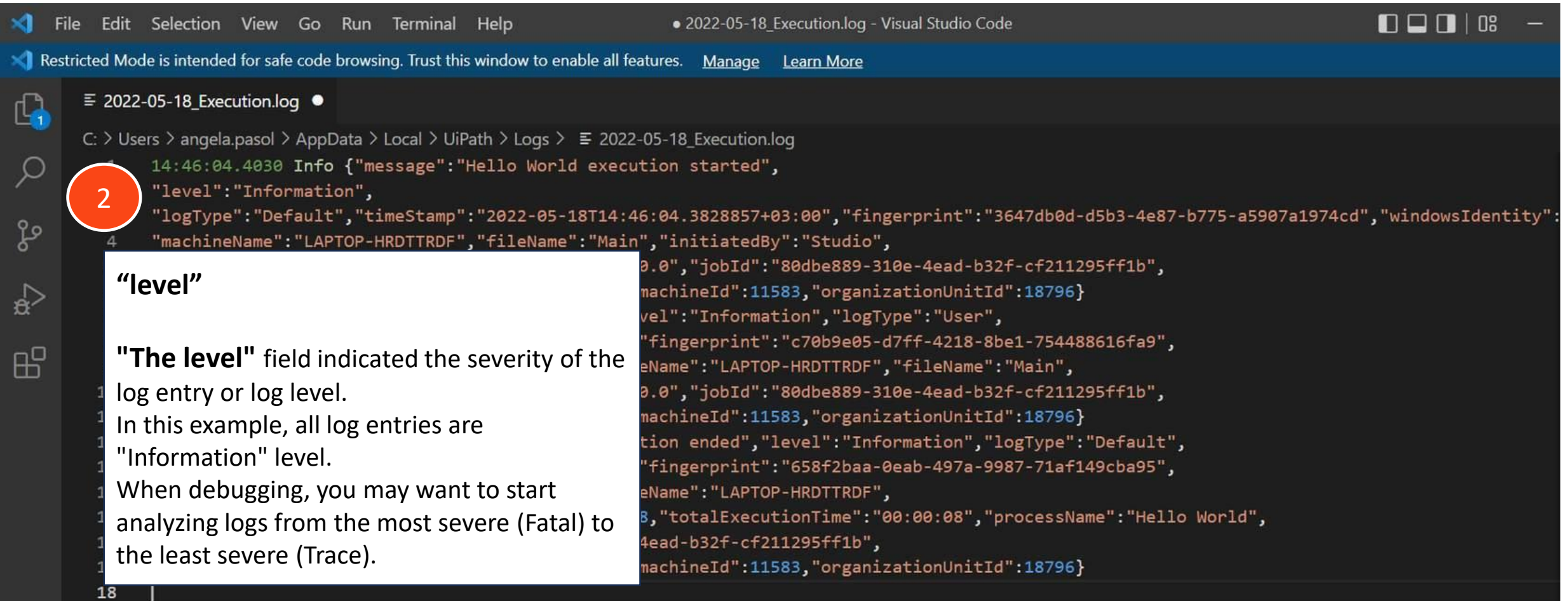
Default logs and use-defined logs

Executing a simple "Hello World" process has generated: two default logs (Process execution started and Process execution end) and one user-defined log ("Hello World!")

```
2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
1 14:46:04.4030 Info {"message":"Hello World execution started",
2 "level":"Information",
3 "logType":"Default","timeStamp":"2022-05-18T14:46:04.3828857+03:00","fingerprint":
4 "machineName":"LAPTOP-HRDTTRDF","fileName":"Main","initiatedBy":"Studio",
5 "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4
6 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
7 14:46:12.4403 Info {"message":"Hello World!","level":"Information","logType":"Default",
8 "timeStamp":"2022-05-18T14:46:12.4403161+03:00","fingerprint":"c70b9e05-d7ff-4218-8be1-754488616fa9",
9 "windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF","fileName":"Main",
10 "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
11 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
12 14:46:12.4563 Info {"message":"Hello World execution ended","level":"Information","logType":"Default",
13 "timeStamp":"2022-05-18T14:46:12.4554528+03:00","fingerprint":"658f2baa-0eab-497a-9987-71af149cba95",
14 "windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF",
15 "fileName":"Main","totalExecutionTimeInSeconds":8,"totalExecutionTime":"00:00:08","processName":"Hello World",
16 "processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
17 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
18
```


The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode



The screenshot shows the Visual Studio Code interface with a log file open. A callout box highlights the "level" field in the log entry.

“level”

"The level" field indicated the severity of the log entry or log level.

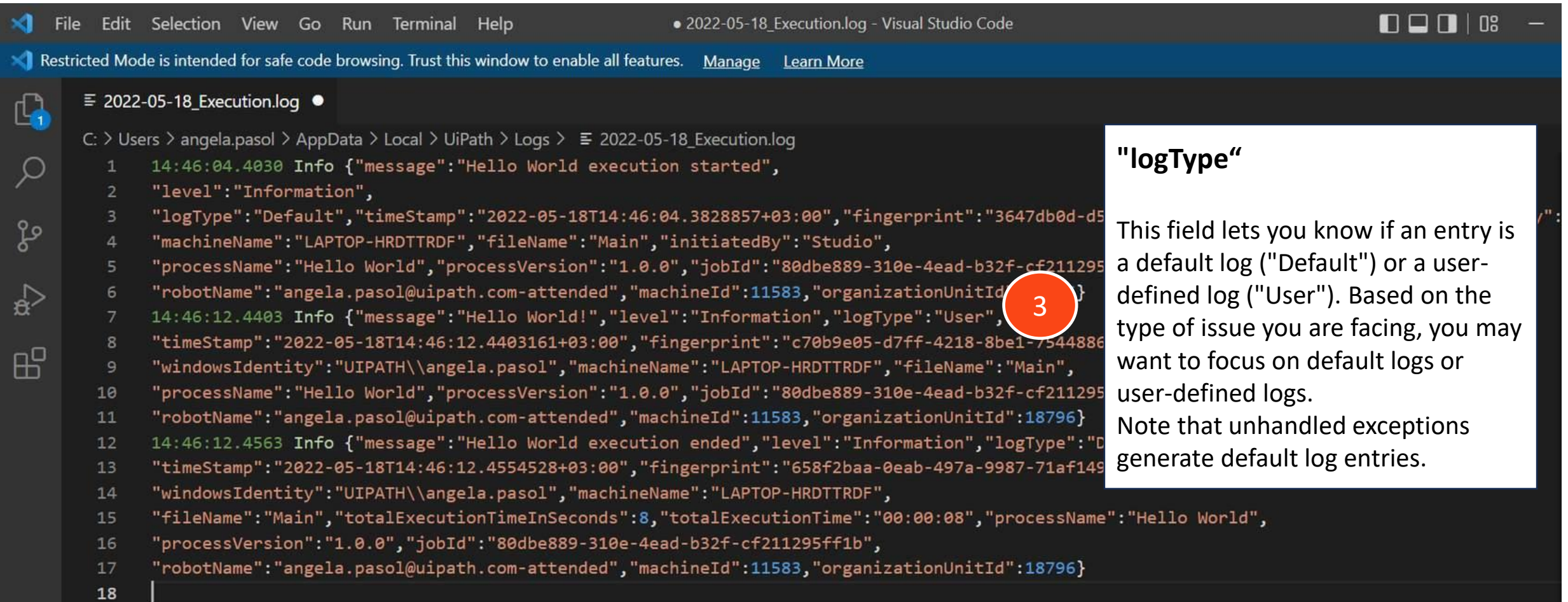
In this example, all log entries are "Information" level.

When debugging, you may want to start analyzing logs from the most severe (Fatal) to the least severe (Trace).

```
2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
14:46:04.4030 Info {"message":"Hello World execution started",
"level":"Information",
"logType":"Default","timestamp":"2022-05-18T14:46:04.3828857+03:00","fingerprint":"3647db0d-d5b3-4e87-b775-a5907a1974cd","windowsIdentity":
"machineName":"LAPTOP-HRDTTRDF","fileName":"Main","initiatedBy":"Studio",
0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
machineId":11583,"organizationUnitId":18796}
vel":"Information","logType":"User",
"fingerprint":"c70b9e05-d7ff-4218-8be1-754488616fa9",
eName":"LAPTOP-HRDTTRDF","fileName":"Main",
0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
machineId":11583,"organizationUnitId":18796}
tion ended","level":"Information","logType":"Default",
"fingerprint":"658f2baa-0eab-497a-9987-71af149cba95",
eName":"LAPTOP-HRDTTRDF",
8,"totalExecutionTime":"00:00:08","processName":"Hello World",
4ead-b32f-cf211295ff1b",
machineId":11583,"organizationUnitId":18796}
```

The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode



```
File Edit Selection View Go Run Terminal Help • 2022-05-18_Execution.log - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

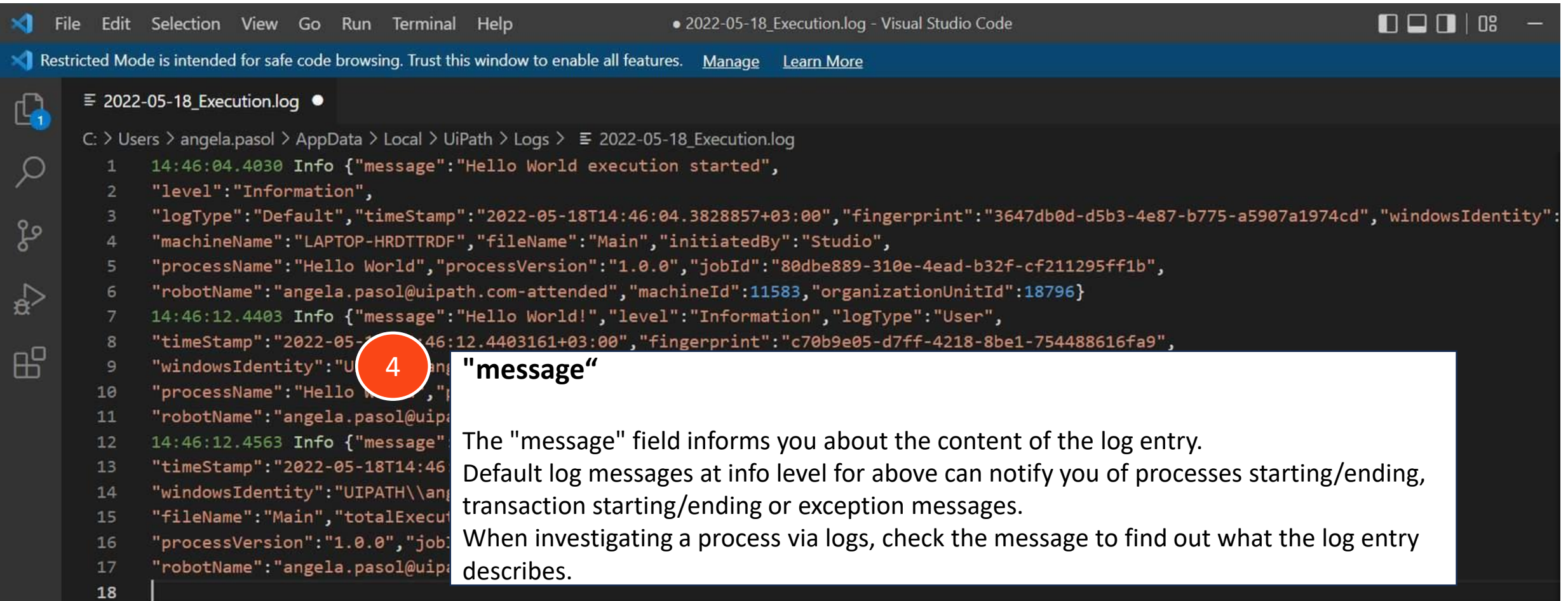
2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
1 14:46:04.4030 Info {"message":"Hello World execution started",
2 "level":"Information",
3 "logType":"Default","timestamp":"2022-05-18T14:46:04.3828857+03:00","fingerprint":"3647db0d-d5
4 "machineName":"LAPTOP-HRDTTRDF","fileName":"Main","initiatedBy":"Studio",
5 "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295
6 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
7 14:46:12.4403 Info {"message":"Hello World!","level":"Information","logType":"User",
8 "timestamp":"2022-05-18T14:46:12.4403161+03:00","fingerprint":"c70b9e05-d7ff-4218-8be1-7544886
9 "windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF","fileName":"Main",
10 "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295
11 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
12 14:46:12.4563 Info {"message":"Hello World execution ended","level":"Information","logType":"D
13 "timestamp":"2022-05-18T14:46:12.4554528+03:00","fingerprint":"658f2baa-0eab-497a-9987-71af149
14 "windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF",
15 "fileName":"Main","totalExecutionTimeInSeconds":8,"totalExecutionTime":"00:00:08","processName":"Hello World",
16 "processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
17 "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
18
```

"logType"

This field lets you know if an entry is a default log ("Default") or a user-defined log ("User"). Based on the type of issue you are facing, you may want to focus on default logs or user-defined logs. Note that unhandled exceptions generate default log entries.

The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode



The screenshot shows the Visual Studio Code interface with a log file open. The log entry at line 7 is highlighted, and a callout box points to the "message" field.

```
1 14:46:04.4030 Info {"message":"Hello World execution started",
2  "level":"Information",
3  "logType":"Default","timestamp":"2022-05-18T14:46:04.3828857+03:00","fingerprint":"3647db0d-d5b3-4e87-b775-a5907a1974cd","windowsIdentity":
4  "machineName":"LAPTOP-HRDTTRDF","fileName":"Main","initiatedBy":"Studio",
5  "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
6  "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
7 14:46:12.4403 Info {"message":"Hello World!","level":"Information","logType":"User",
8  "timestamp":"2022-05-18T14:46:12.4403161+03:00","fingerprint":"c70b9e05-d7ff-4218-8be1-754488616fa9",
9  "windowsIdentity":"UIPATH\\angela.pasol",
10 "processName":"Hello World",
11 "robotName":"angela.pasol@uipath.com-attended"}
12 14:46:12.4563 Info {"message":"Transaction started","level":"Information","logType":"User",
13 "timestamp":"2022-05-18T14:46:12.4563161+03:00","fingerprint":"c70b9e05-d7ff-4218-8be1-754488616fa9",
14 "windowsIdentity":"UIPATH\\angela.pasol",
15 "fileName":"Main","totalExecutionTime":0,
16 "processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
17 "robotName":"angela.pasol@uipath.com-attended"}
18
```

4 "message"

The "message" field informs you about the content of the log entry. Default log messages at info level for above can notify you of processes starting/ending, transaction starting/ending or exception messages. When investigating a process via logs, check the message to find out what the log entry describes.

The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode

"processName"

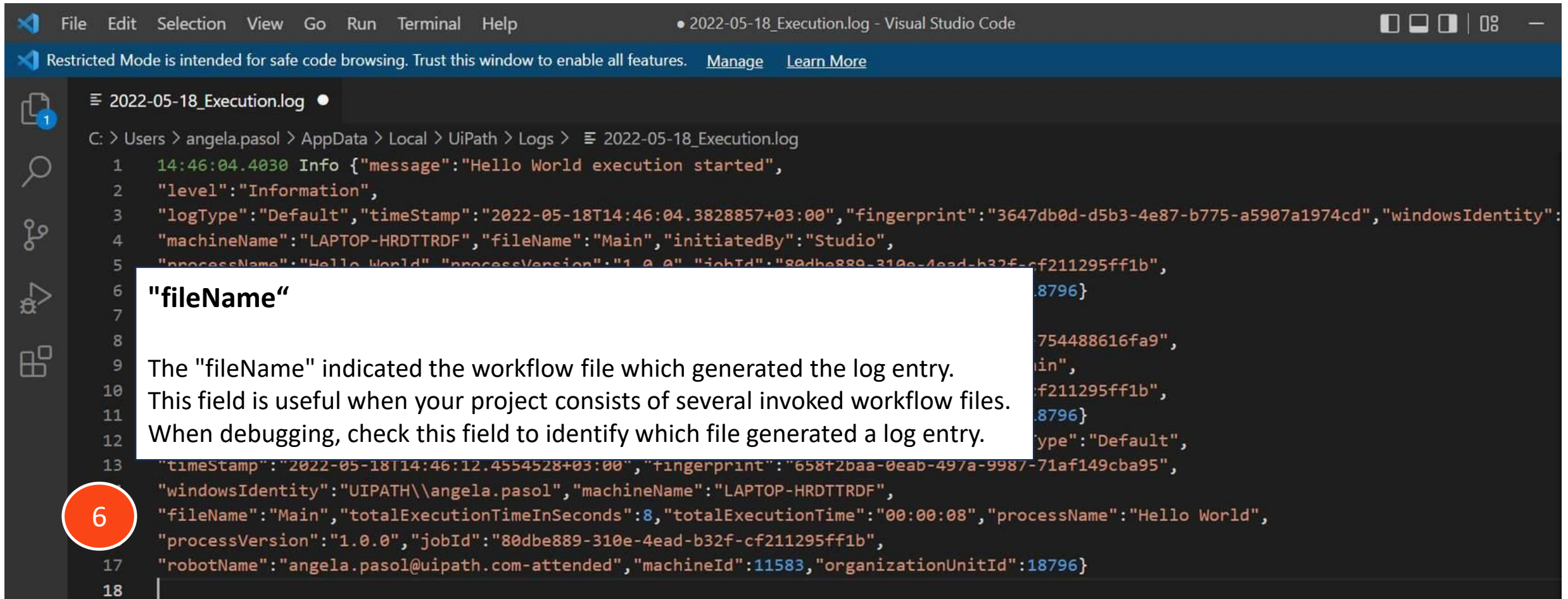
This field indicates the name of the process which generated the log entry. As UiPath Studio will create a single log file for all process executed in Studio in one day, filtering entries by "processName" will let you focus on the processes you are interested in.

5

```
"timestamp":"2022-05-18T14:46:12.4403161+03:00","fingerprint":"c70b9e05-d7ff-4218-8be1-754488616fa9",
"windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF","fileName":"Main",
"processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
"robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
14:46:12.4563 Info {"message":"Hello World execution ended","level":"Information","logType":"Default",
"timestamp":"2022-05-18T14:46:12.4554528+03:00","fingerprint":"658f2baa-0eab-497a-9987-71af149cba95",
"windowsIdentity":"UIPATH\\angela.pasol","machineName":"LAPTOP-HRDTTRDF",
"fileName":"Main","totalExecutionTimeInSeconds":8,"totalExecutionTime":"00:00:08","processName":"Hello World",
"processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
"robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
```

The Anatomy of a Log Entry

Key log fields generated at the **Trace level and above** when executing a project in run mode



The screenshot shows the Visual Studio Code interface with a log file open. A callout box highlights the 'fileName' field in the log entry. The log entry is as follows:

```
1 14:46:04.4030 Info {"message":"Hello World execution started",
2  "level":"Information",
3  "logType":"Default","timestamp":"2022-05-18T14:46:04.3828857+03:00","fingerprint":"3647db0d-d5b3-4e87-b775-a5907a1974cd","windowsIdentity":
4  "machineName":"LAPTOP-HRDTTRDF","fileName":"Main","initiatedBy":"Studio",
5  "processName":"Hello World","processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
6  "fileName":"Main","totalExecutionTimeInSeconds":8,"totalExecutionTime":"00:00:08","processName":"Hello World",
7  "processVersion":"1.0.0","jobId":"80dbe889-310e-4ead-b32f-cf211295ff1b",
8  "robotName":"angela.pasol@uipath.com-attended","machineId":11583,"organizationUnitId":18796}
```

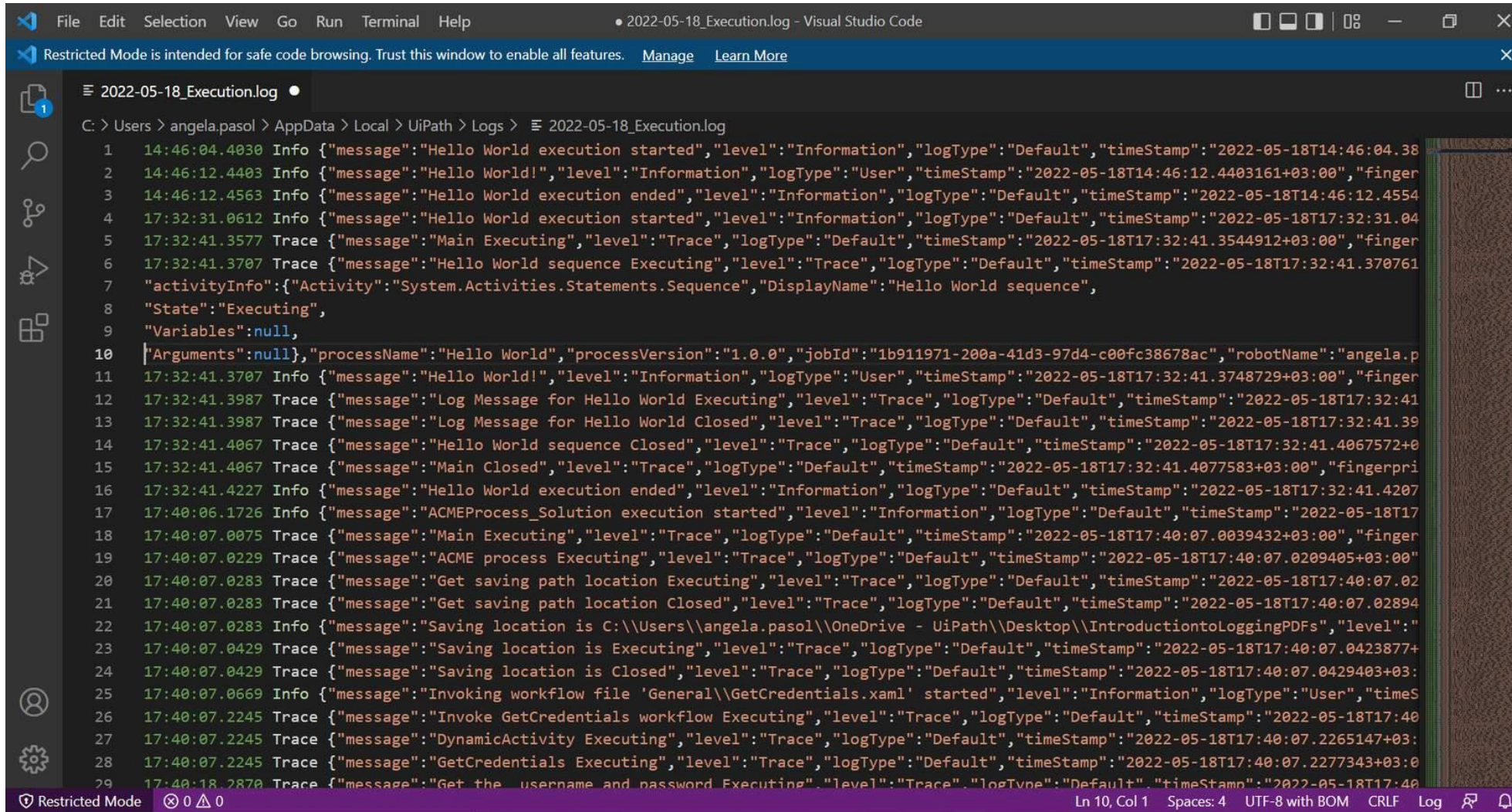
"fileName"

The "fileName" indicated the workflow file which generated the log entry. This field is useful when your project consists of several invoked workflow files. When debugging, check this field to identify which file generated a log entry.

6

The Anatomy of a Log Entry

Log entries generated at the **Verbose level** and their key fields. These are the types of files generated when running a project **locally in debug mode with Log Activities option enabled**

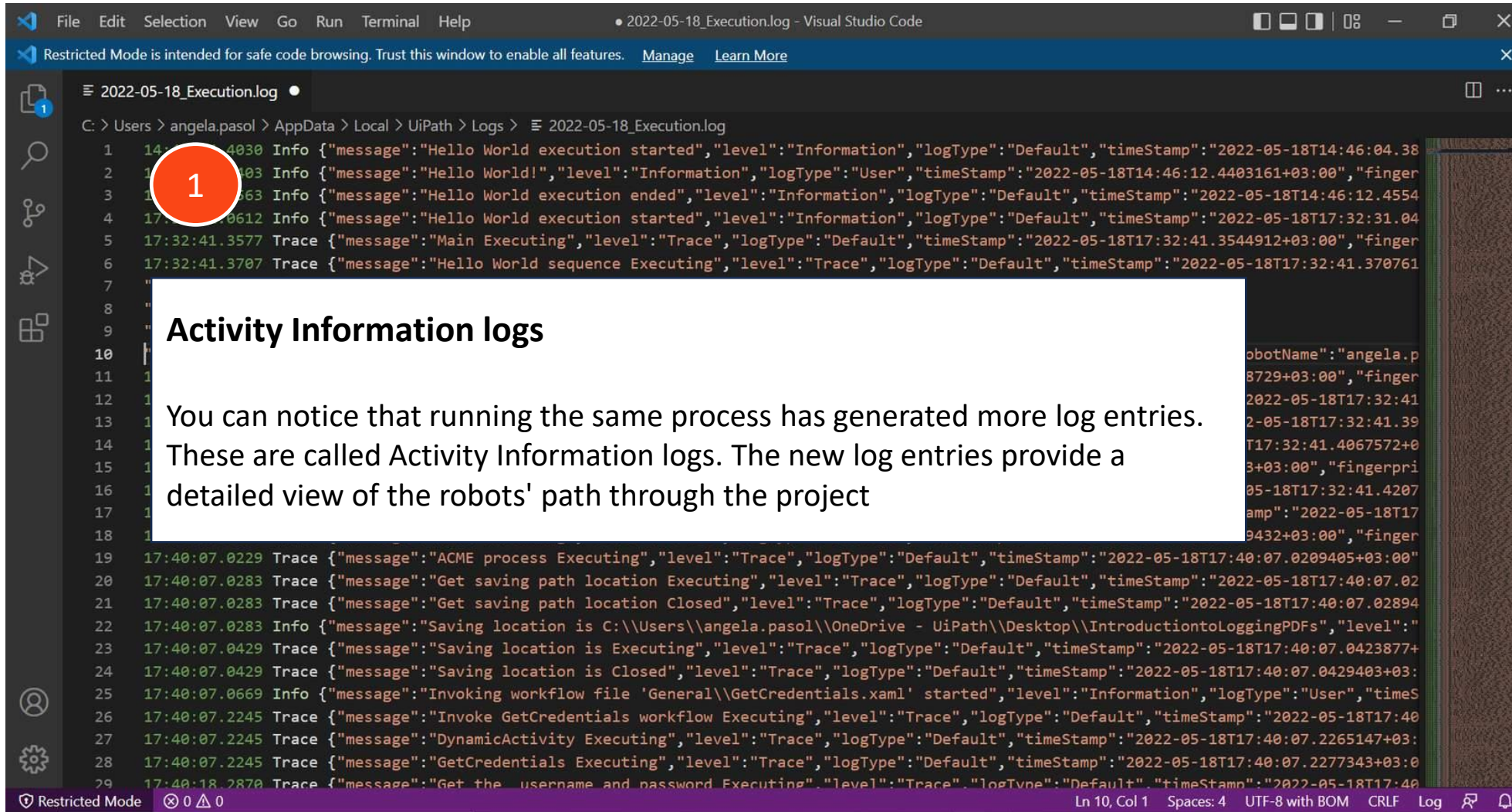


```
File Edit Selection View Go Run Terminal Help • 2022-05-18_Execution.log - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
1 14:46:04.4030 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:04.38
2 14:46:12.4403 Info {"message":"Hello World!", "level":"Information","logType":"User","timeStamp":"2022-05-18T14:46:12.4403161+03:00","finger
3 14:46:12.4553 Info {"message":"Hello World execution ended","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:12.4554
4 17:32:31.0612 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T17:32:31.04
5 17:32:41.3577 Trace {"message":"Main Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.3544912+03:00","finger
6 17:32:41.3707 Trace {"message":"Hello World sequence Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.370761
7 "activityInfo":{"Activity":"System.Activities.Statements.Sequence","DisplayName":"Hello World sequence",
8 "State":"Executing",
9 "Variables":null,
10 "Arguments":null},"processName":"Hello World","processVersion":"1.0.0","jobId":"1b911971-200a-41d3-97d4-c00fc38678ac","robotName":"angela.p
11 17:32:41.3707 Info {"message":"Hello World!", "level":"Information","logType":"User","timeStamp":"2022-05-18T17:32:41.3748729+03:00","finger
12 17:32:41.3987 Trace {"message":"Log Message for Hello World Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41
13 17:32:41.3987 Trace {"message":"Log Message for Hello World Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.39
14 17:32:41.4067 Trace {"message":"Hello World sequence Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.4067572+0
15 17:32:41.4067 Trace {"message":"Main Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.4077583+03:00","fingerpri
16 17:32:41.4227 Info {"message":"Hello World execution ended","level":"Information","logType":"Default","timeStamp":"2022-05-18T17:32:41.4207
17 17:40:06.1726 Info {"message":"ACMEProcess_Solution execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T17
18 17:40:07.0075 Trace {"message":"Main Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0039432+03:00","finger
19 17:40:07.0229 Trace {"message":"ACME process Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0209405+03:00"
20 17:40:07.0283 Trace {"message":"Get saving path location Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.02
21 17:40:07.0283 Trace {"message":"Get saving path location Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.02894
22 17:40:07.0283 Info {"message":"Saving location is C:\\Users\\angela.pasol\\OneDrive - UiPath\\Desktop\\IntroductiontoLoggingPDFs","level":"
23 17:40:07.0429 Trace {"message":"Saving location is Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0423877+
24 17:40:07.0429 Trace {"message":"Saving location is Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0429403+03:
25 17:40:07.0669 Info {"message":"Invoking workflow file 'General\\GetCredentials.xml' started","level":"Information","logType":"User","times
26 17:40:07.2245 Trace {"message":"Invoke GetCredentials workflow Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
27 17:40:07.2245 Trace {"message":"DynamicActivity Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2265147+03:
28 17:40:07.2245 Trace {"message":"GetCredentials Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2277343+03:0
29 17:40:18.2870 Trace {"message":"Get the username and password Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
```


The Anatomy of a Log Entry

Log entries generated at the **Verbose level** and their key fields. These are the types of files generated when running a project **locally in debug mode with Log Activities option enabled**



```
File Edit Selection View Go Run Terminal Help • 2022-05-18_Execution.log - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
1 14:46:04.38 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:04.38
2 14:46:12.4403 Info {"message":"Hello World!","level":"Information","logType":"User","timeStamp":"2022-05-18T14:46:12.4403161+03:00","finger
3 14:46:12.4554 Info {"message":"Hello World execution ended","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:12.4554
4 17:32:31.04 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T17:32:31.04
5 17:32:41.3577 Trace {"message":"Main Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.3544912+03:00","finger
6 17:32:41.3707 Trace {"message":"Hello World sequence Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.370761
7
8
9
10
11
12
13
14
15
16
17
18
19 17:40:07.0229 Trace {"message":"ACME process Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0209405+03:00"
20 17:40:07.0283 Trace {"message":"Get saving path location Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.02
21 17:40:07.0283 Trace {"message":"Get saving path location Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.02894
22 17:40:07.0283 Info {"message":"Saving location is C:\\Users\\angela.pasol\\OneDrive - UiPath\\Desktop\\IntroductiontoLoggingPDFs","level":"
23 17:40:07.0429 Trace {"message":"Saving location is Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0423877+
24 17:40:07.0429 Trace {"message":"Saving location is Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0429403+03:
25 17:40:07.0669 Info {"message":"Invoking workflow file 'General\\GetCredentials.xml' started","level":"Information","logType":"User","timeS
26 17:40:07.2245 Trace {"message":"Invoke GetCredentials workflow Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
27 17:40:07.2245 Trace {"message":"DynamicActivity Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2265147+03:
28 17:40:07.2245 Trace {"message":"GetCredentials Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2277343+03:0
29 17:40:18.2870 Trace {"message":"Get the username and password Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
```

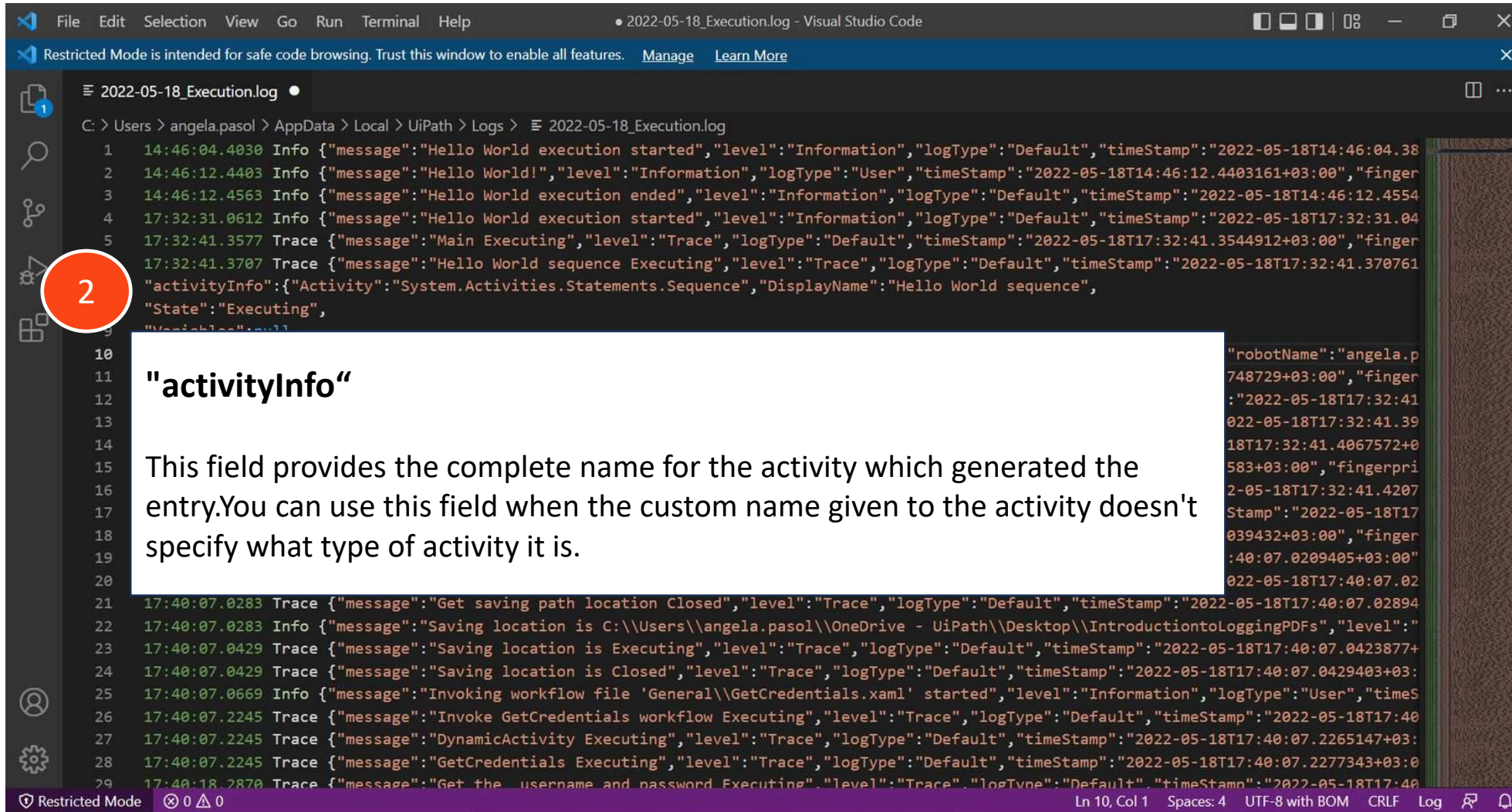
Activity Information logs

You can notice that running the same process has generated more log entries. These are called Activity Information logs. The new log entries provide a detailed view of the robots' path through the project

Ln 10, Col 1 Spaces: 4 UTF-8 with BOM CRLF Log

The Anatomy of a Log Entry

Log entries generated at the **Verbose level** and their key fields. These are the types of files generated when running a project **locally in debug mode with Log Activities option enabled**



The screenshot shows the Visual Studio Code interface with a log file open. A callout box highlights the 'activityInfo' field in a log entry. The log entry is as follows:

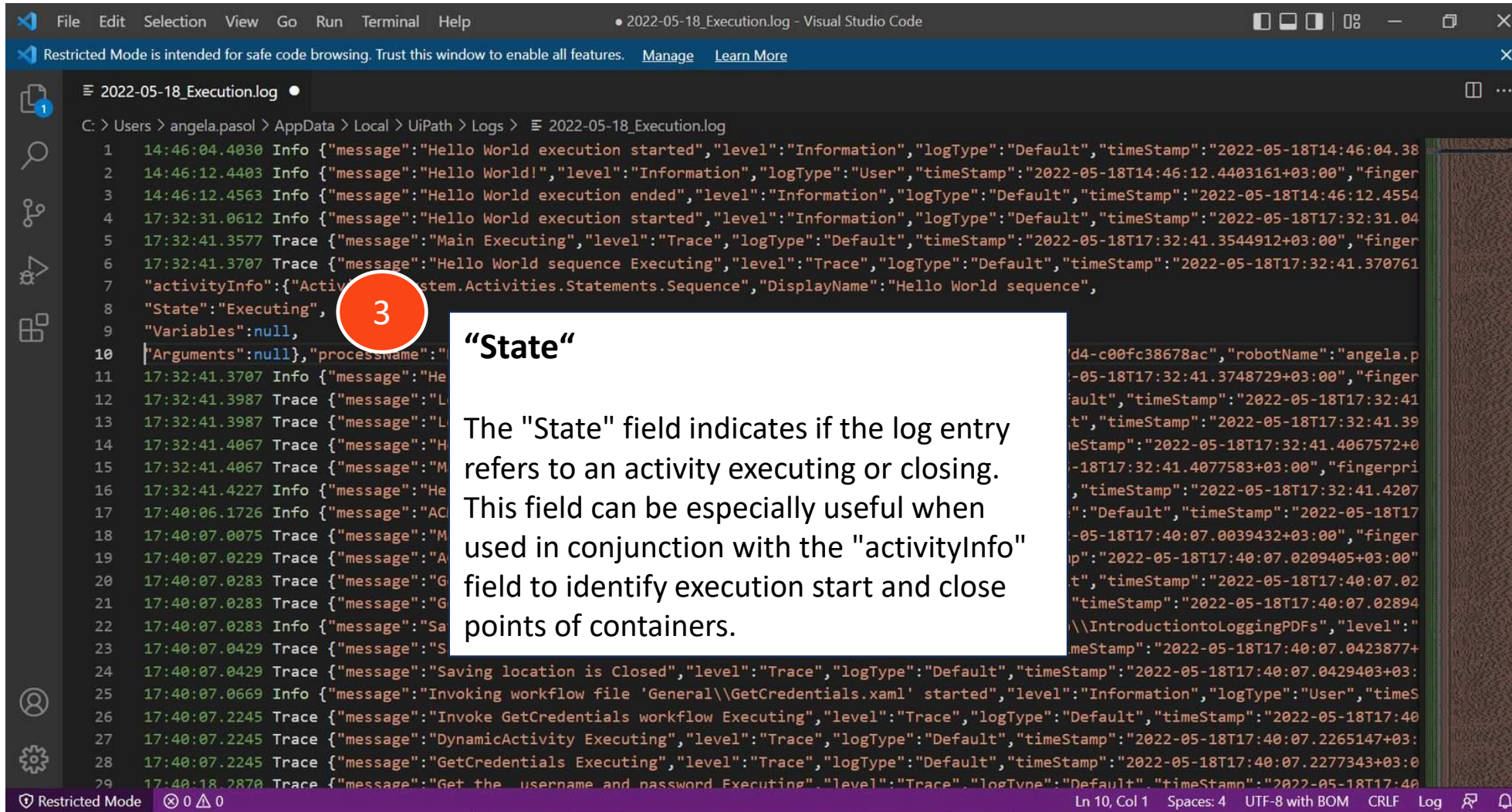
```
17:32:41.3577 Trace {"message":"Main Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.3544912+03:00","finger
17:32:41.3707 Trace {"message":"Hello World sequence Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.370761
"activityInfo":{"Activity":"System.Activities.Statements.Sequence","DisplayName":"Hello World sequence",
"State":"Executing",
"Version":"v1.1
"robotName":"angela.p
748729+03:00","finger
:"2022-05-18T17:32:41
022-05-18T17:32:41.39
18T17:32:41.4067572+0
583+03:00","fingerpri
2-05-18T17:32:41.4207
Stamp":"2022-05-18T17
039432+03:00","finger
:40:07.0209405+03:00"
022-05-18T17:40:07.02
```

"activityInfo"

This field provides the complete name for the activity which generated the entry. You can use this field when the custom name given to the activity doesn't specify what type of activity it is.

The Anatomy of a Log Entry

Log entries generated at the **Verbose level** and their key fields. These are the types of files generated when running a project **locally in debug mode with Log Activities option enabled**



The screenshot shows the Visual Studio Code interface with a log file named '2022-05-18_Execution.log' open. The log entries are displayed in a list format. A red circle with the number '3' highlights the 'State' field in the log entry at line 8. A callout box titled 'State' explains its function.

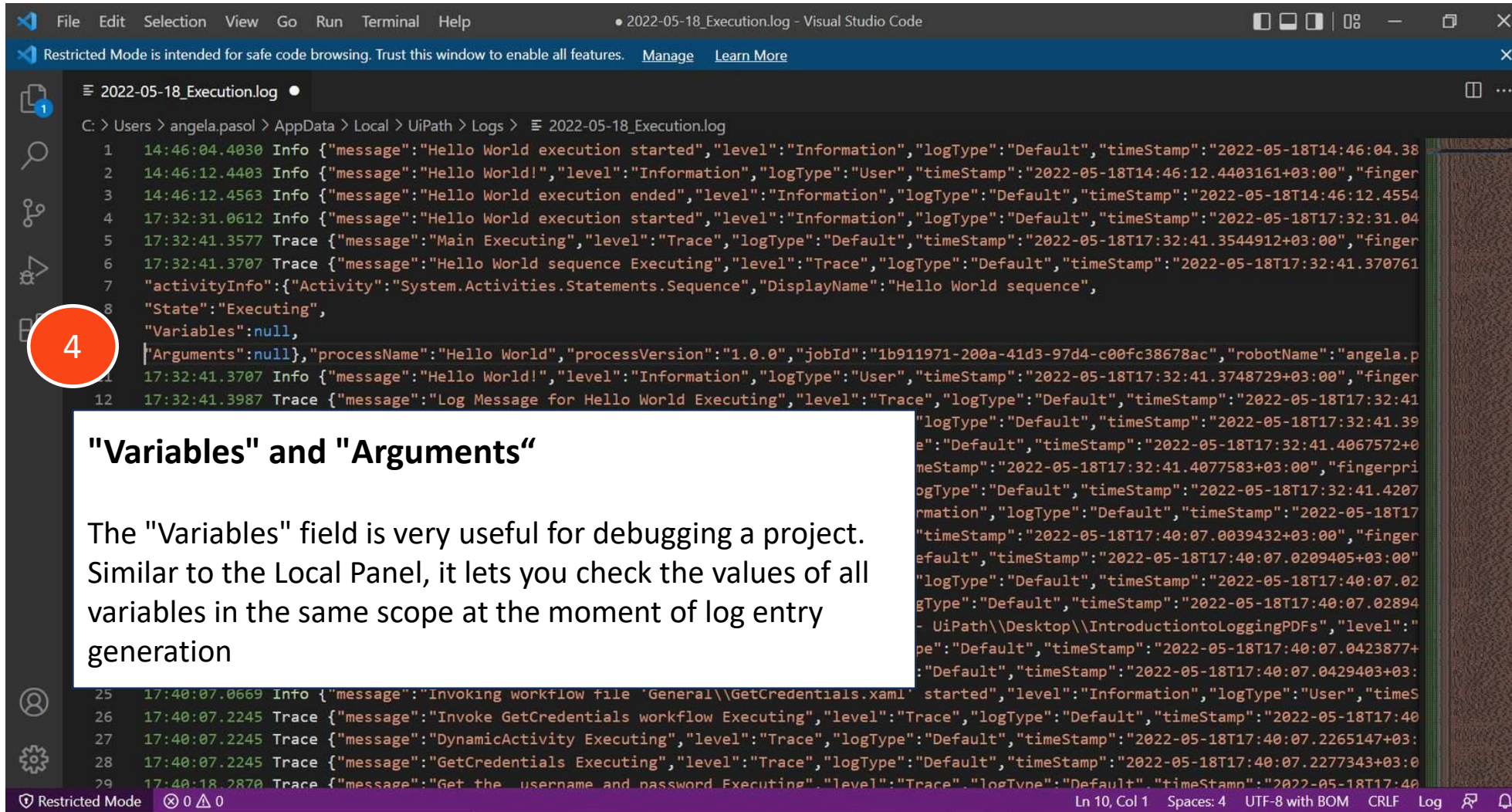
```
1 14:46:04.4030 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:04.38
2 14:46:12.4403 Info {"message":"Hello World!","level":"Information","logType":"User","timeStamp":"2022-05-18T14:46:12.4403161+03:00","finger
3 14:46:12.4554 Info {"message":"Hello World execution ended","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:12.4554
4 17:32:31.0612 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T17:32:31.04
5 17:32:41.3577 Trace {"message":"Main Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.3544912+03:00","finger
6 17:32:41.3707 Trace {"message":"Hello World sequence Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.370761
7 "activityInfo":{"Activity":"System.Activities.Statements.Sequence","DisplayName":"Hello World sequence",
8 "State":"Executing",
9 "Variables":null,
10 "Arguments":null,"processName":"
11 17:32:41.3707 Info {"message":"He
12 17:32:41.3987 Trace {"message":"L
13 17:32:41.3987 Trace {"message":"L
14 17:32:41.4067 Trace {"message":"H
15 17:32:41.4067 Trace {"message":"M
16 17:32:41.4227 Info {"message":"He
17 17:40:06.1726 Info {"message":"AC
18 17:40:07.0075 Trace {"message":"M
19 17:40:07.0229 Trace {"message":"A
20 17:40:07.0283 Trace {"message":"G
21 17:40:07.0283 Trace {"message":"G
22 17:40:07.0283 Info {"message":"Sa
23 17:40:07.0429 Trace {"message":"S
24 17:40:07.0429 Trace {"message":"Saving location is Closed","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.0429403+03:
25 17:40:07.0669 Info {"message":"Invoking workflow file 'General\\GetCredentials.xml' started","level":"Information","logType":"User","times
26 17:40:07.2245 Trace {"message":"Invoke GetCredentials workflow Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
27 17:40:07.2245 Trace {"message":"DynamicActivity Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2265147+03:
28 17:40:07.2245 Trace {"message":"GetCredentials Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2277343+03:0
29 17:40:18.2870 Trace {"message":"Get the username and password Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
```

State

The "State" field indicates if the log entry refers to an activity executing or closing. This field can be especially useful when used in conjunction with the "activityInfo" field to identify execution start and close points of containers.

The Anatomy of a Log Entry

Log entries generated at the **Verbose level** and their key fields. These are the types of files generated when running a project **locally in debug mode with Log Activities option enabled**



```
File Edit Selection View Go Run Terminal Help • 2022-05-18_Execution.log - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

2022-05-18_Execution.log
C: > Users > angela.pasol > AppData > Local > UiPath > Logs > 2022-05-18_Execution.log
1 14:46:04.4030 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:04.38
2 14:46:12.4403 Info {"message":"Hello World!", "level":"Information","logType":"User","timeStamp":"2022-05-18T14:46:12.4403161+03:00","finger
3 14:46:12.4553 Info {"message":"Hello World execution ended","level":"Information","logType":"Default","timeStamp":"2022-05-18T14:46:12.4554
4 17:32:31.0612 Info {"message":"Hello World execution started","level":"Information","logType":"Default","timeStamp":"2022-05-18T17:32:31.04
5 17:32:41.3577 Trace {"message":"Main Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.3544912+03:00","finger
6 17:32:41.3707 Trace {"message":"Hello World sequence Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41.370761
7 "activityInfo":{"Activity":"System.Activities.Statements.Sequence","DisplayName":"Hello World sequence",
8 "State":"Executing",
9 "Variables":null,
10 "Arguments":null}, "processName":"Hello World", "processVersion":"1.0.0", "jobId":"1b911971-200a-41d3-97d4-c00fc38678ac", "robotName":"angela.p
11 17:32:41.3707 Info {"message":"Hello World!", "level":"Information","logType":"User","timeStamp":"2022-05-18T17:32:41.3748729+03:00","finger
12 17:32:41.3987 Trace {"message":"Log Message for Hello World Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:32:41
13 "logType":"Default","timeStamp":"2022-05-18T17:32:41.39
14 e":"Default","timeStamp":"2022-05-18T17:32:41.4067572+0
15 meStamp":"2022-05-18T17:32:41.4077583+03:00","fingerpri
16 ogType":"Default","timeStamp":"2022-05-18T17:32:41.4207
17 rmation","logType":"Default","timeStamp":"2022-05-18T17
18 "timeStamp":"2022-05-18T17:40:07.0039432+03:00","finger
19 efault","timeStamp":"2022-05-18T17:40:07.0209405+03:00"
20 "logType":"Default","timeStamp":"2022-05-18T17:40:07.02
21 gType":"Default","timeStamp":"2022-05-18T17:40:07.02894
22 - UiPath\\Desktop\\IntroductiontoLoggingPDFs", "level":":
23 pe":"Default","timeStamp":"2022-05-18T17:40:07.0423877+
24 :":"Default","timeStamp":"2022-05-18T17:40:07.0429403+03:
25 17:40:07.0669 Info {"message":"Invoking workflow file 'General\\GetCredentials.xml' started","level":"Information","logType":"User","times
26 17:40:07.2245 Trace {"message":"Invoke GetCredentials workflow Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
27 17:40:07.2245 Trace {"message":"DynamicActivity Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2265147+03:
28 17:40:07.2245 Trace {"message":"GetCredentials Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40:07.2277343+03:0
29 17:40:18.2870 Trace {"message":"Get the username and password Executing","level":"Trace","logType":"Default","timeStamp":"2022-05-18T17:40
```

"Variables" and "Arguments"

The "Variables" field is very useful for debugging a project. Similar to the Local Panel, it lets you check the values of all variables in the same scope at the moment of log entry generation

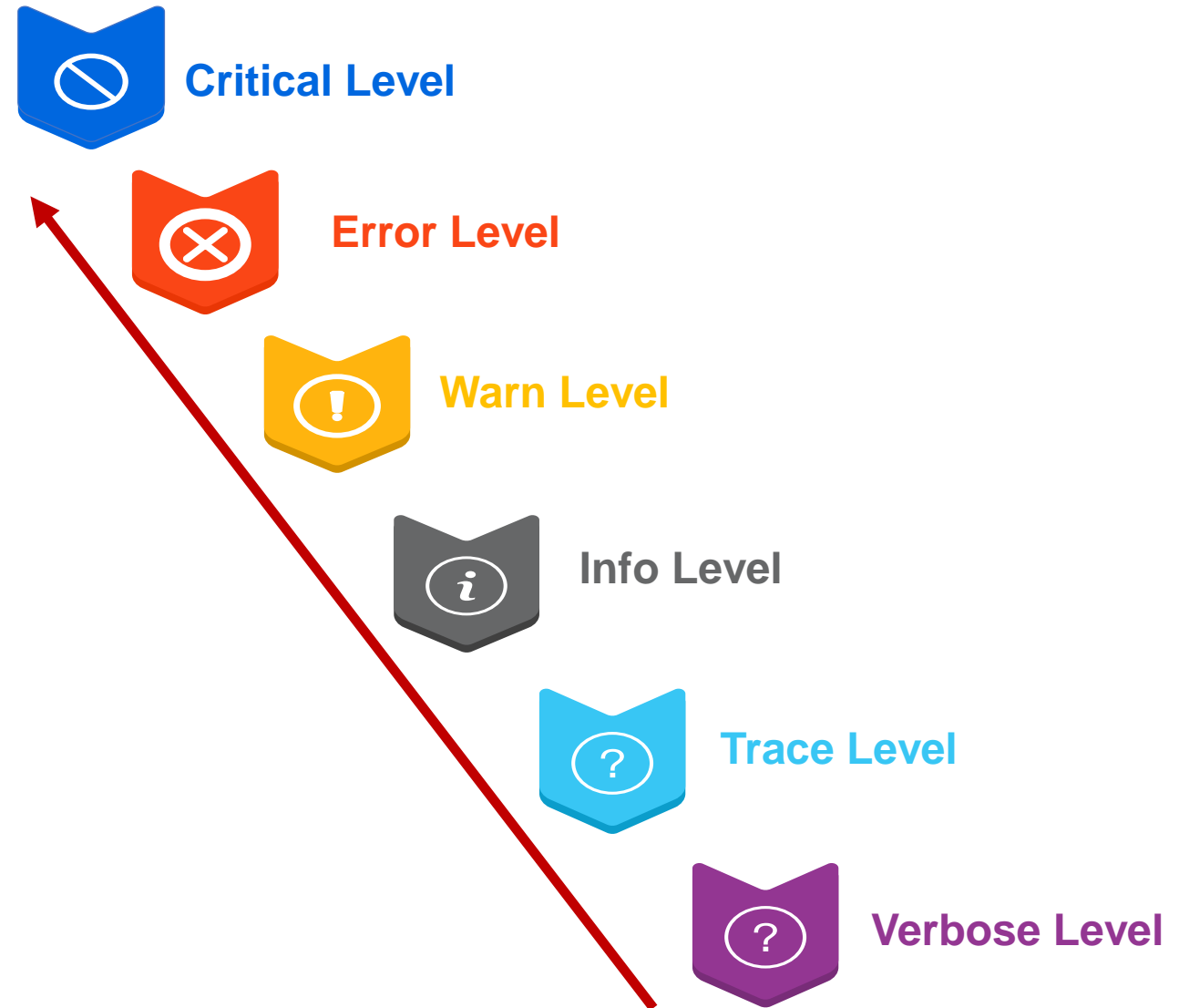
Setting the log level for UiPath Assistant

Running a process from UiPath Assistant will

- send the logs to Orchestrator if connected
- generate logs in
`%localappdata%\UiPath\Logs\<shortdate>_Execution.log`

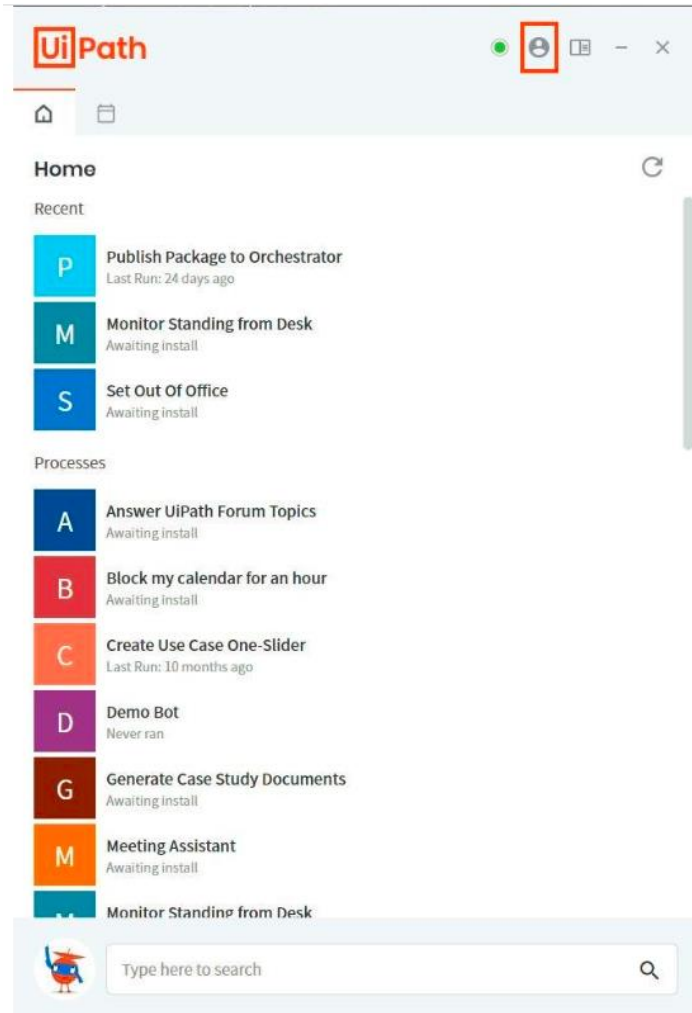
The log entries are generated at the level set in UiPath Assistant and above.

For example, if the level is set to Error, it'll only generate the logs at the Error and Fatal levels.



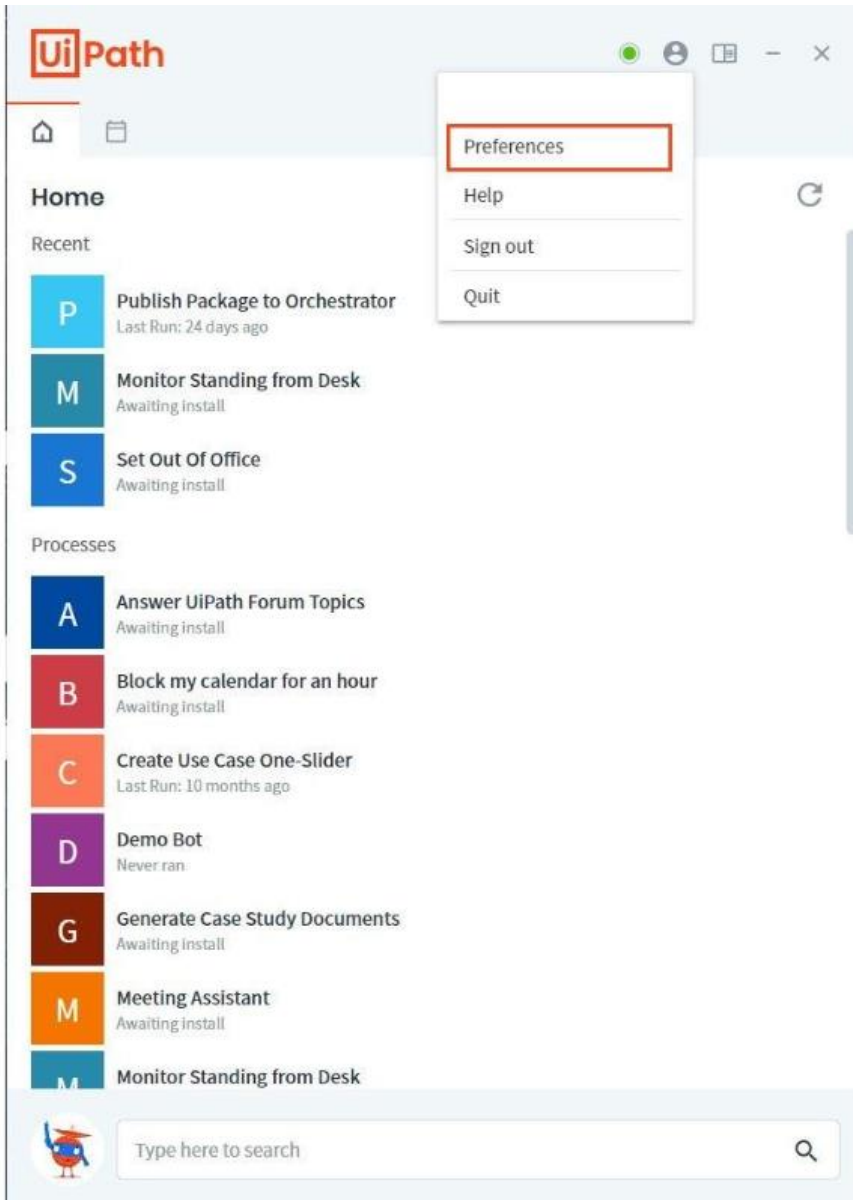
Configure UiPath Assistant to Log Events

This example shows you how to configure the UiPath Assistant to log events at the Information level and above for processes executed through it.



Step 1: Open UiPath Assistant
Click the **Preference** icon

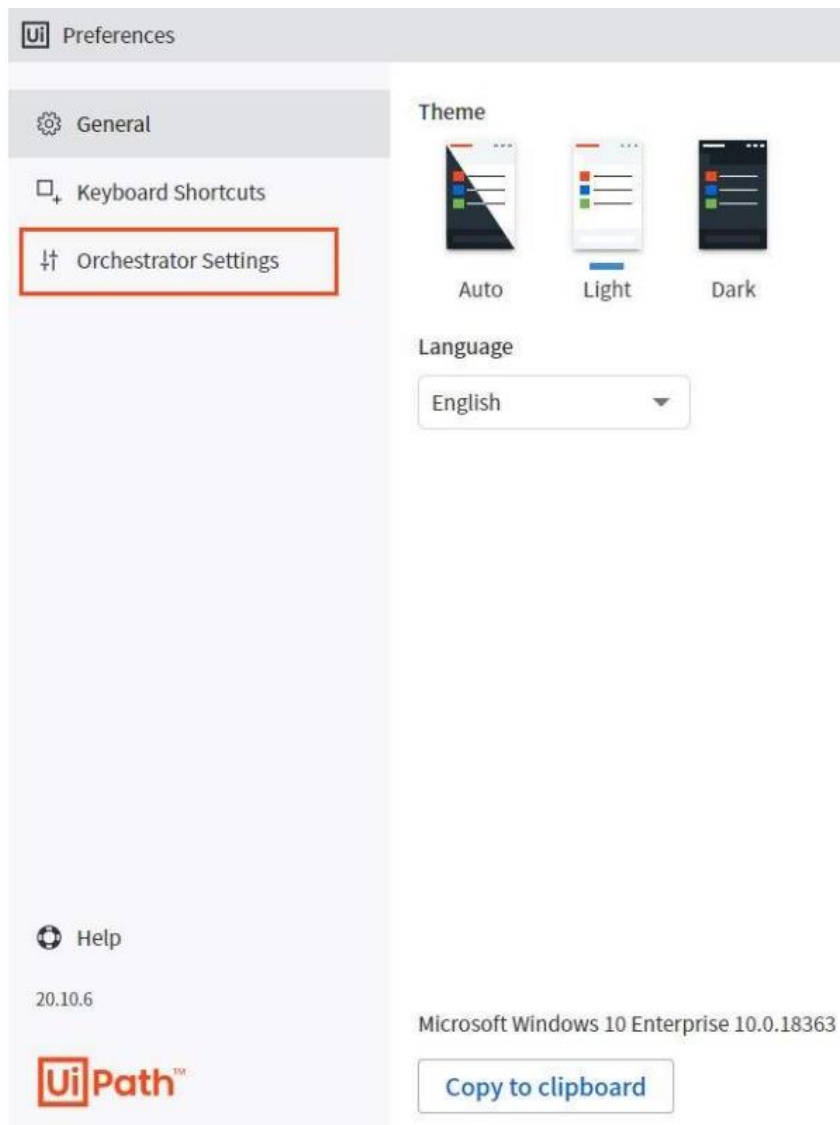
Configure UiPath Assistant to Log Events



This example shows you how to configure the UiPath Assistant to log events at the Information level and above for processes executed through it.

Step 2: Click the Preferences button

Configure UiPath Assistant to Log Events



This example shows you how to configure the UiPath Assistant to log events at the Information level and above for processes executed through it.

Step 3: Click the Orchestrator Setting option

Configure UiPath Assistant to Log Events



This example shows you how to configure the UiPath Assistant to log events at the Information level and above for processes executed through it.

Preferences

General

Keyboard Shortcuts

Orchestrator Settings

Help

20.10.6

UiPath

Orchestrator Configuration

Connection Type

Machine Key

Machine Name

Orchestrator URL

Machine Key

Disconnect

Status: Connected, Licensed

Logging

Log Level

Trace

Step 4: Click the Log Level drop-down

Configure UiPath Assistant to Log Events



This example shows you how to configure the UiPath Assistant to log events at the Information level and above for processes executed through it.

Step 5: Select the Information option

Configure UiPath Assistant to Log Events



This example shows you how to configure the UiPath Assistant to log events at the Information level and above for processes executed through it.

UiPath Preferences

General

Keyboard Shortcuts

Orchestrator Settings

Orchestrator Configuration

Connection Type

Machine Key

Machine Name

Orchestrator URL

Machine Key

Disconnect

Status: Connected, Licensed

Logging

Log Level

Information

Click Close, and you're done.

❑ Log message activities should ideally be used in the following scenarios:

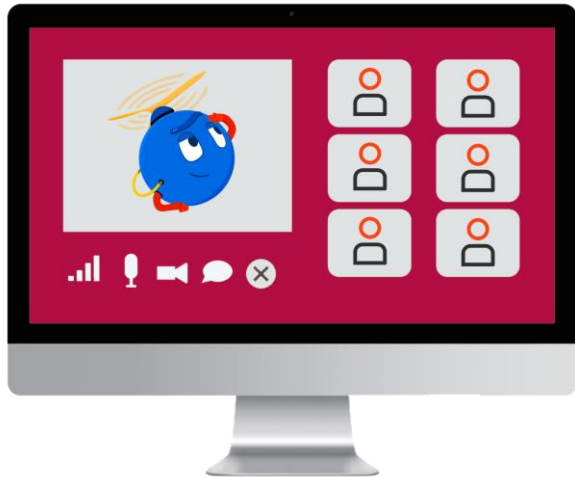
- ✓ Each time an exception is caught in a Catch block (Log level = Error)
- ✓ Each time a Business Rule Exception is thrown (Log Level = Error)
- ✓ When data is read from external sources. For example, log a message at Information level when an Excel file is read (Log Level = Information)
- ✓ In Parallel or Pick activities, log messages on every branch, to trace the branch is taken (Log Level = Information)
- ✓ In If/Flowchart Decision/Switch/Flow Switch activities

Note: There should be no more than two (maximum three) nested IF statements so it depends from workflow to workflow if you should add or not log message activities.

- ☐ When you invoke a workflow use the two properties Log Entry and Log Exit (Log level = Information). These let you specify if you want to log entry into/ exit from an invoked workflow. This is particularly useful for debugging without adding lots of Log Message activities
- ☐ Write meaningful log messages: The message should be clear enough
- ☐ Add context to your log message: Include values of meaningful variables in that context

Classroom Exercise

Demonstrate how to use **Log Message** activity of Info, Error, and Fatal level in a workflow by creating a bank transaction process that logs the responses and actions of the user.



- Display a message to the user about the available balance in the bank account and log this action at the Info level
- Ask the user to enter the amount to withdraw and log the response at the Info level
- If the requested amount is greater than the available amount, then
 - Log an error message, “User does not have enough amount.”
 - Display a message to the user, “You are trying to withdraw more than the allowed amount.”
 - Log a fatal error message, “Invalid amount entered. Process terminated.”, and terminate the process
- If the requested amount is less than the available amount, then
 - Display a message to the user, “Your remaining amount is \$x”
 - Log an Info level message, “Remaining amount displayed to the user”

The ACME project consists of several workflow files. The project navigates to ACME System 1, logs in, navigates to the Download Client and Support page, downloads all the Work Item PDF documents, and logs out.



- Apply the logging recommendations and add log messages to a workflow to make the execution easy to supervise.

Here are some hints to help you approach this exercise

- Use info level to indicate that something happened, that the process entered a certain state, for purely informative purposes.
- Use the warn level for unexpected behavior that might occur, any information that might be useful to know but is not preventing the process from being complete.
- Use trace level to log any important data that you need to stand out from the rest of the log information.
- Use fatal when a key activity might not work, and the process will not be able to complete without it.