Errors and Exception Handling

UiPath Academy

Introduction to controlled fires

Errors vs Exceptions

- Errors are events that a particular program can’t normally deal with.
  - Ex: Syntax Error
- Exceptions, aka exceptional events, are events that occur during the execution of a program that disrupts the normal flow of instructions.
  - Types:
    - System Exceptions
      - Exceptions thrown by the system that occur during runtime
      - Ex: NullReferenceException
    - Business Exceptions
      - Exceptions thrown by your program due to some data not conforming to certain application logic
      - Ex: when an item isn’t found in the product list for a vendor
      - Because we don’t have the ability to make custom exceptions in uipath, use the BusinessRuleException class to throw an application exception

Throwing Exceptions

- Obviously, exceptions tell you when something is wrong
  - You’d want to know why your program isn’t working
- Exceptions are also good for enforcing logic/rules in your program
- You throw exceptions using the Throw or Rethrow Activities
  - Throw is used to throw new exceptions
  - Rethrow is used in a try catch block to rethrow the caught exception

Handling Exceptions

- Try Catch Finally Block
- Retry Scope
- Continue on Error Property
- Global Exception Handler

Try Catch Finally

[Try Catch]

- Try
  - The activities performed which have a chance of throwing an error
- Catch
  - The activity or set of activities to be performed when an exception occurs. Please note that multiple exceptions and corresponding activities can be added to this block.
- Finally
  - The activity or set of activities to be performed after the Try and Catches blocks are executed.
  - This section is executed only when no exceptions are thrown or when an error occurs and is caught in the Catches block (without being re-thrown).

Retry Scope

Retry Scope

- The Retry Scope activity retries the contained activities as long as the condition is not met or an error is thrown.
- The activity has 2 main sections:
  - **Action** - Holds the activities we want to retry.
  - **Condition** - Holds the termination condition. The Retry Scope activity retries the activities in the Action section as long as this termination condition is not met.
- Additional properties
  - **NumberOfRetries** - The number of times that the sequence is to be retried.
  - **RetryInterval** - Specifies the amount of time (in seconds) between each retry.

Continue on Error Property

- The ContinueOnError is a property that specifies if the execution of the activity should continue even when the activity throws an error.
- If the ContinueOnError is set to True on an activity that has a scope (such as Attach Window or Attach Browser), then all the errors that occur in other activities inside that scope are also ignored.

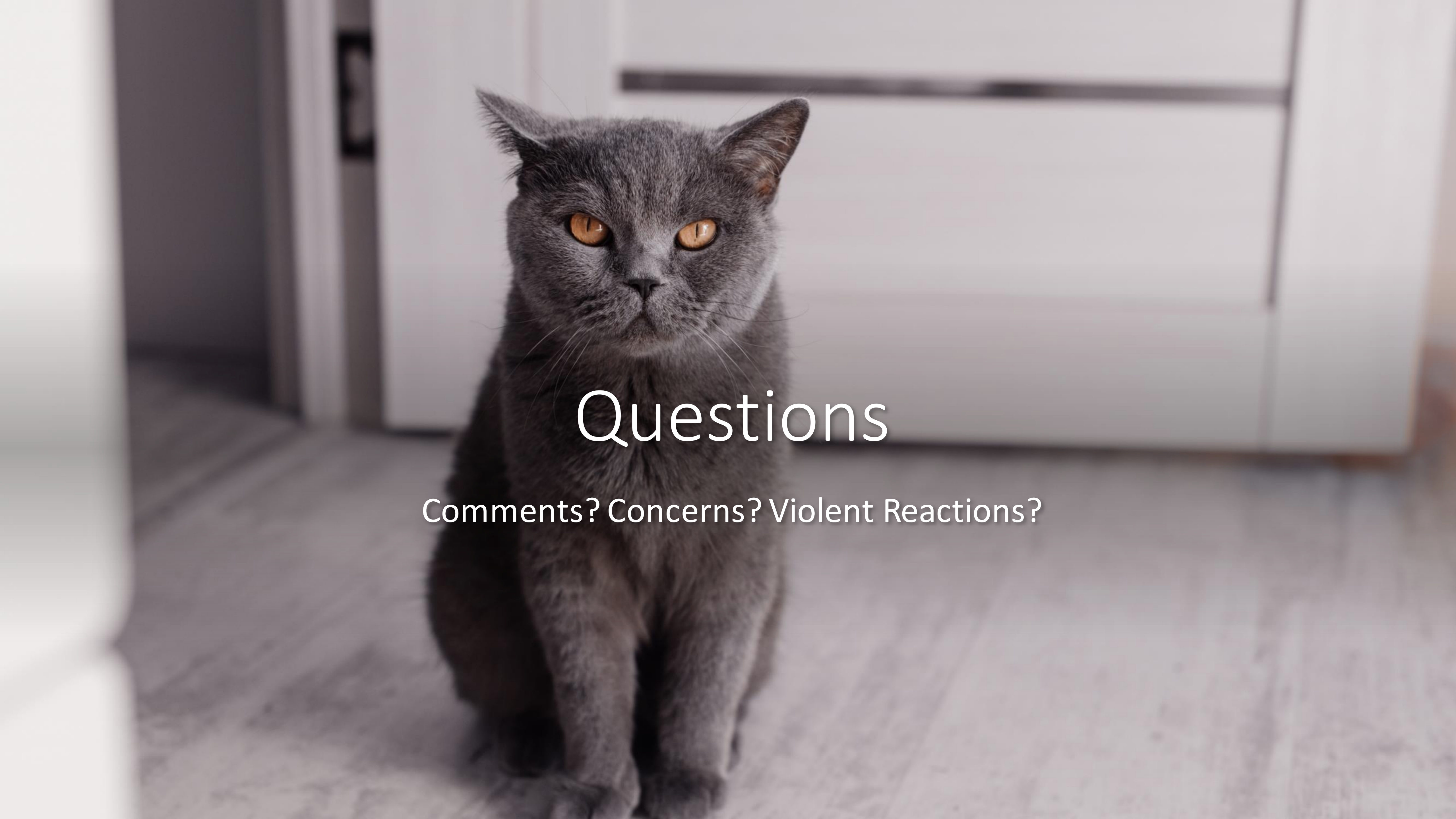
Global Exception Handler

Global Exception Handler

- A type of workflow designed to determine the process' behavior when encountering an unexpected exception.
- Only one Global Exception Handler can be set per automation project.
- In its default configuration, the Global Handler catches the exceptions thrown by any activity in the process at runtime and executes a standard response - ignore, retry, abort or continue, as predefined at design time.
- The Global Exception Handler has 2 predefined arguments, that shouldn't be removed:
  - **errorInfo** with the In direction - contains the information about the error that was thrown and the workflow that failed.
  - **result** with the Out direction - used for determining the next behavior of the process when it encounters the error.

Next behavior options for result

- **Continue** - The exception is re-thrown.
- **Ignore** - The exception is ignored, and the execution continues from the next activity.
- **Retry** - The activity which threw the exception is retried.
- **Abort** - The execution stops after running the current handler.



Questions

Comments? Concerns? Violent Reactions?

Discussion Questions

- What is the difference between errors and exceptions?
- What are the different types of exceptions?
- How can you throw exceptions?
- Why throw exceptions?
- How do you handle exceptions?
- What's the global exception handler?