

Style Guide

Moore

Notes

This guide serves as a foundation and should be adapted to the specific project. Regularly reviewing and updating this style guide will be done by Nick Moore to ensures it stays relevant and effective.

Contact Nick if you have any questions: nmoore@msiconsulting.com

Content

1 - 2: Naming Conventions

3 - 5: Workflow Design

6: General Best Practices

7: Code Review

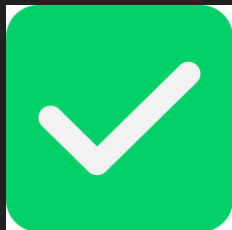
8: Standards

9: Checklist

Naming Conventions

- PascalCase (FirstName)
- Meaningful names
 - 3-4 words max to describe the variable, argument, workflow
- Applies to:
 - Variables, Arguments, Workflows
- Prefix:
 - DataTables (dt_CustomerInfo), Argument Direction (in_, out_, io_)
 - Consider prefixing variable types if similar name (StrCustomerAge, IntCustomerAge)

Naming Convention (Examples)



dt_CustomerInfo
out_CalculationResult
FirstName
LastName
in_LeftOperand
StrCustomerAge
IntCustomerAge
AddNumbers.xaml
CalculateInterest.xaml



customerinfodt
info_DT
x
str
argument1
first-name
last_name
Add-nums.xaml
CALCULATE.xaml

Workflow Design

- Invoke Workflow File activity - Use it!
- Break Workflows into smaller, atomic-ish, workflows
 - The breaking down of workflows is something that will change dynamically based on the project
 - Consider logical places to break workflow files down
 - Think using different container activities in different workflows
- This is important to consider when developing workflows, large workflows may need more memory than is available to them at runtime

Atomic Value

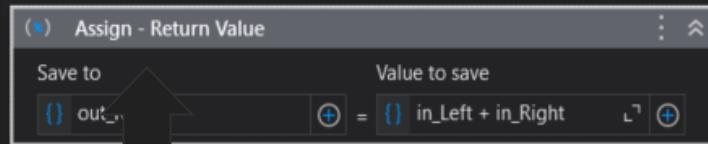
FirstName	LastName
Nick	Moore

Not Atomic Value

Name
Nick Moore

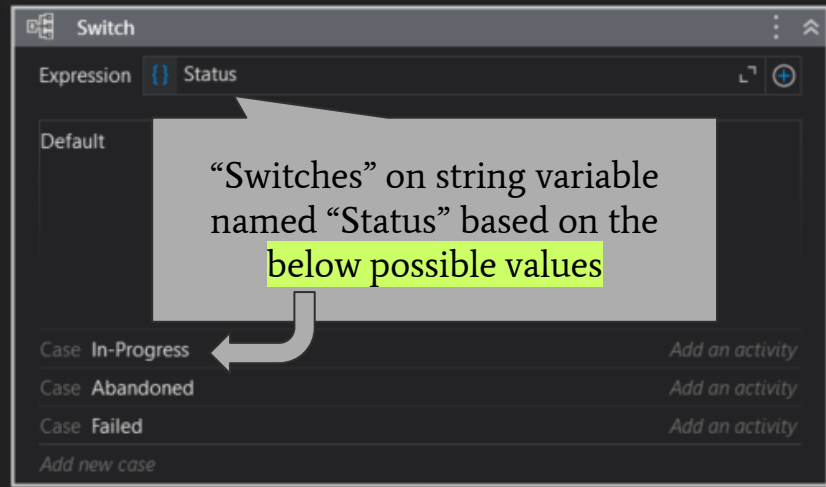
Workflow Design

- Name activities ({Activity Name} - {Description})
 - (Log Message - Entering workflow)
- Annotate where appropriate, like container activities
 - Should be a brief description of the purpose of activities
- Log Information
 - Entering/Exiting workflows, interacting with vital outside information
- Error Handling
 - Use TryCatch around operations that may fail, consider a RetryScope activity in the Catch block



Workflow Design

- Sequences
 - For linear processes
- Flowcharts
 - For branching logic
 - Avoid too much nesting of Flowcharts
- Switch Activity
 - Strings do not use quotation marks for cases
 - Most useful when there are more than two options (If & Else If)
- Invoke Code
 - Be sure to document this activity



General Best Practices

- File Management
 - Keep files separated in appropriate folders (Configuration/Data/Workflows)
- Configuration Files
 - Inside of appropriate folder
 - Provide client with guide to create this file
 - Guide to provide insight on naming conventions and any other possible issues
- Credential Handling
 - Do not hardcode credentials, use Orchestrator to manage sensitive information
- Performance
 - Avoid unnecessary loops
 - Use parallel activities if/when possible

Code Review

- Readability: Is the workflow easy to understand?
- Reusability: Are there any components that can be modularized?
- Error Handling: Is there adequate error handling?
- Performance: Is the workflow optimized for performance?
- Security: Are credentials and sensitive data handled securely?
- Logging: Are there sufficient logs for monitoring and debugging?

Standards

- Consistent formatting
 - Spacing & Alignment
- Reusability
 - Leverage libraries where appropriate to reuse workflow parts
- Documentation
 - Add appropriate activity names and annotations
- Over do it!
 - Unsure? It is almost always better to have more information than not enough information

Checklist

- Use PascalCase (UpperCamelCase) with meaningful words
- Breakdown large workflows
- Document your project
- Use the appropriate activities
- Manage files and credentials appropriately
- Be consistent & proud of your work