RPA Coding Standards

Section	Category	Description	Comment
Activity Packages	Activity	Do not use non-UiPath Activity packages unless there is a no way to accomplish an action without using non-sanctioned activities	Written consent must be sought from the Lead Architect for the use of any non-sanctioned activity or custom packages
Arguments	Code Readability	All Arguments must start with either in_, out_, or io_	
Annotations, comments and descriptive names	Code Readability	Always use logical Log Messages and annotate your workflows with clear, easy to understand descriptions on what's about to happen in the workflow	Always consider: What would you want to see in the code if you have to make changes/troubleshoot code that you did not write?
		Use the comment activity whenever some part of the process might be hard to understand or when some hack has been done in order to succeed so that when you come back to the code or someone else picks it up later it is easy to understand what happened	
		Always use descriptive names for workflows, activities, variables, attributes and sequences. When you use an activity it might have a name like "tblCst23" <- This is not descriptive and should be changed to something more appropriate	
Attach Browser/ Attach Window	Activity	Only include in the scope of the Attached Browser/Window the activities that interact with the browser or window being attached to. If attaching to a different Browser / Window then end the one scope and begin the other	Avoid nested Attach Browser/Attach Window
Cleaning Up	Coding	Delete all locally created files	Maintain disk space by removing files created during

			the automation run
		Close open applications	Leave the workstation tidy for the next automation
Default values	Coding	No default values to be captured in the Variables and Arguments section	Use an Initialise Variables sequence and explicitly indicate what values are being used to initialise the variables
Email Groups/Distribution Lists	Best Practice	An Email Group (Distribution List) must be created and managed by the business unit receiving emails from the automation. This allows the business unit to add/remove recipients from the Distribution List without any automation code having to be amended	Email Groups can be created by following this url: Requests - OIT Services - New Item (vodafone.com)
Email Handling	Best Practice	The automations are not to send emails to individuals. Email notifications from automations must go to an email group	
Excel	Excel Activities	Disable popup requesting whether you want to automatically update external links. This is done in Excel -> Options -> Advanced	An unattended robot would pause if this popup is displayed. It does not throw an exception or time out
		Only do Excel activities within an Excel Application Scope and only one or two activities at a time	
Excel Scope	Excel Activities	Only include in the scope of the Excel Scope the activities directly relating to the scope e.g. Read Range/Write Range	Any activity included in the Excel Scope keeps the Excel file open (even if Visible is set to False) which consumes computing resources
Exception Handling	Best Practice	Throw Activities must start with the word HANDLED: <exception message>	This is to clearly distinguish between handled and unhandled exceptions. E.g. HANDLED: SAP did not open
		Outlook: Recalled emails cause an exception	Consider creating a mail rule in Outlook that deletes/moves Recall emails prior to the bot trying to process them
Framework	REFramework	Both Dispatcher and Performer are to be built within the REFramework	Refer to REFramework worksheet for high-level architecture

		For some processes the Dispatcher should only add items to the queue if there are no existing Transaction Items to be processed	This would often apply to processes that have daily source data updates
Hard-coding Values	Coding	Never hard code any values inside the code. Use the Config file or Orchestrator Assets	This prevents the code from having to be redeployed if something changes e.g. VAT changes from 15% to 16%
IF Statements	Coding	Avoid nested IF Statements. If you have to nest an IF Statement more than once it indicates that your code should be in a flow chart	
Image Recognition	Activity	Only to be used if everything else has been tried but failed. If image recognition is used e.g. click image	Ensure that the Robot in Orchestrator is configured to the same resolution as the resolution of the computer that was used to capture the image
		Capture more than you need and use Cursor Position in the activity's properties to position the click to where it needs to be	The smaller the image captured the longer it will take the bot to find it on the screen
Log Message	Activity	To track the progress of the automation use the Log Message activity. Most progress tracking should be set to Trace level. Use Error, Warn, Fatal levels where appropriate.	Do not default all Log Messages to Info
Logic conditions	Logic Conditions	A and B and C means every part (A&B&C) has to be true for the condition to be True A and B or C equates to (B or C) and A • this should usually be written as (A and B) or C meaning A or B must be true or C must be true for the entire condition to be True • OR A and (B or C) meaning B or C must be true or A must be true for the entire condition to be True	Ensure your logic conditions do what you think they will do e.g. when used in IF statements
Naming Conventions	Code Readability	Process names must be in the format: Business Unit + Process Name as agreed with	e.g. FINOPS_Debit_Order_Reinstatement_Dispatcher

		Business + Dispatcher/Performer	
		Descriptive names for variables, attributes and workflows to make the code easier to read. The name should focus on the positive outcome of the logic when naming workflows.	
		Variables: Pascal case – “VariableName”	e.g. PopupWindow or popupWindow or Popup_Window or popup_Window
		Arguments: in/out prefix – “in_ArgumentName / out_ArgumentName / io_ArgumentName”	
		Workflows: Human understandable indication of what the workflow is used for	e.g. Workflow Login to SAP
OCR Activities	Activity	In most cases, ensure windows are maximised before reading	If you can't see it then the robot can't see it either
Orchestrator assets	Orchestrator	Use assets and credentials as much as possible to make your process easier to configure or update	
		Do not use windows credentials when deploying to Production	
		Name assets and credentials accordingly and make them easy to identify. E.g.. cred_Sox_Mediation or url_Remedy	Start with the type first and then the name. It makes the Asset/Credential easier to find
		Store values such as credentials, paths, URL’s, email addresses, names and anything else that may be subject to change, within an Orchestrator Asset	
Emails	Emails Outlook	Do not use Outlook as this creates a dependency on Outlook. Use the Vodacom built SendEmail package instead. Only use Outlook if the automation has to read emails that are sent to it	Reading emails from Outlook may no longer work since the Vodacom Microsoft 365 migration

		Always include in the email which bot sent the email e.g. environment.machinename & environment.username	This is to identify which bot sent the email and helps when troubleshooting
Queues	Orchestrator	Always use a Queue Reference when adding a Transaction to a Queue	This will allow someone to search for a specific Queue Item in Orchestrator - if necessary
		Never set the Unique Reference to True in Orchestrator as it can never be switched off unless the Queue is deleted and recreated which will lose all historical data in the Queue	Handle keeping Queue items unique, in the code e.g. check if it exists before adding a new one
Queue Items	Activity	Queue Items must have a Deadline set when adding the item to a Queue. By default, the deadline could be set to 24hrs after the Queue Item is added. Use the Now.AddHours(24) function, for example.	This enables the dynamic Queue management functionality of Orchestrator
Reporting & Reporting Config	Reporting	It is good practice to notify the business via email of when the process starts, when it finishes, and report on how many success and failures there were	
		Ensure your process has been captured in the Config table for reporting	This is to work out the AHT savings
Retry Scope	Activity	Wrap all Orchestrator related activities in a properly used Retry Scope e.g. Get Transaction, Set Transaction Status, Add Queue Item, etc.	This will avoid the automation failing due to an intermittent Orchestrator communication issue e.g. congested network may prevent sending or receiving to and from Orchestrator
Robot Resolution	Orchestrator	If the Robot Resolution needs to be set (e.g. for Click Image) ensure that in the Robot settings that Login to Console is switched On and the value is No	Otherwise the Robot Resolution will be set to the last resolution of the laptop that remoted on to the VDI where the process is running
Throw	Activity	Use the Throw activity where necessary to indicate handled Application or Business exceptions	This will help make the most of the REF State Machine as it is built to manage exceptions and it also helps with reporting

Try Catches	Coding	Do not abuse Try Catches - always use them sparingly	A misplaced Try Catch can mask a coding issue
Version Control	GitLab	Use GitLab to store your code and manage package versions	https://pgitap1zatcrh.vodacom.corp
		Orchestrator has built in version control for every published package	This is not a way to backup your code
		It is the developer's responsibility to ensure their code is backed up in GitLab	Losing code that has not been backed up in GitLab is negligence
		Do not indicate the version of the automation in the process' name	If you do and publish your code it will be published as a brand new package in Orchestrator
Web Browser	Vodafone Standard	Always use Microsoft Edge unless there is a justifiable reason to use a different browser	This is the Vodafone standard
Workflow structure	REFramework	InitAllApplications and Process should always start as a workflow	
		Do not duplicate code in the same project. If the same code is needed in more than one place, create a separate workflow and invoke it where needed	Use micro-journeys. These will also assist in testing individual parts of the code
		Use sequences for small process sections and flowcharts for bigger process sections	If in doubt, always use a flowchart as it makes it easier to add/remove sequences and provides the process flow
		Do not add exception handling with a blank catch	Use a Log Message activity to indicate whether the catch was triggered
		Add a comment to all delay activities to explain the reason it is in the code	Give some clarity as to why it was used. E.g. Application is slow/reliable selector not available to determine when page has loaded
		Avoid using a generic data type. If you do e.g. scraping text, then convert it to a specific datatype immediately after the activity that used the generic datatype	

		Avoid large selectors. The more complex a selector the more chance of it failing if something changes	
		All constant values to be maintained outside of the code. E.g. within a config.xlsx file or Orchestrator Asset	
		Use inner most scope for variables	
		Do not nest IF statements more than once. If you need to do so, then make use of a flowchart and a flow decision activity	Nested IF statements make it difficult to follow the logic of the flow. Flowcharts are easier to understand
		Do not use infinite loops in your code	Always have a way for a loop to eventually exit e.g. use a counter that increments within the loop that when it reaches a set threshold will allow the loop to end
		Add Should Stop before the final commit/save/submit during the process	This allows the automation to be elegantly stopped from Orchestrator without having to kill the process mid-execution
Write Line	Activity	Never use a Write Line activity	This sends unnecessary log updates to the Orchestrator database