



Registration Quality Alert Resolution

Process Definition Document

Purpose: The Process Definition Document (PDD) formalizes the final design of an automation solution. It functions as an approval artifact for automation and operations teams, enabling the transition from design to development once approved.

TABLE OF CONTENTS

I. Automation Overview	3
1.1 Objectives	3
1.2 Operational metrics	3
1.3 Design Team	3
II. Automation Details	3
2.1 Automation Overview	3
2.2 Applications	3
2.3 Process Map	10
2.4 Toolkit documentation	10
2.5 Out of scope	10
III. Approval	11
3.1 Approval	11
3.2 Future Initiatives	11
3.3 Revisions	12

I. AUTOMATION OVERVIEW

1.1 Objectives

The automation will achieve

- Autonomous resolution of 5 high volume alerts to ensure the appropriate payers and payer details are on the accounts in Cerner. The alerts are 52,53,16,1309,1461.

1.2 Operational metrics

The automation will impact the following operational metrics

- Number of alerts resolved
- Potentially eligibility denials avoided

1.3 Design Team

The following individuals contributed to the automation design, must be informed of changes, and agree to support the maintenance of this automation

Name	Role / Title	Contact
Manal AlMaghafi	Operational Owner	
Savanna Bollitier	Operational SME(s)	
Sean Furlong	Solution Architect	
Snehitha Mamidi	Developer	

II. AUTOMATION DETAILS

2.1 Automation Overview

Item	Description
Area / Department	FCC
Automation Name	Registration Eligibility Alert Resolution
Automation Description	See Below

Automation Workflow Prose

Dispatcher

1. Network File Approach
 - a. Open File Explorer and Navigate to the following file path
 - i. \\mhcetnas01\MHC_CBS\CBS Automation\Cerner Inbound

- b. Retrieve the file that matches the corresponding naming convention
 - i. NEED TO CONFIRM FILE NAME(s)
 - ii. One file should be all account data and no alerts
 - iii. One file should be just account number and alert data
 - iv. One file for Account Number, Health Plan, Bill status
 - v. One file for payer mapping
- c. If ANY of the files are NOT present
 - i. Wait 30 minutes and retry, max attempts = 3
- d. If max attempts exceeded?
 - i. Send email with smtp to cbs_automation@mclaren.org
 - ii. Subject:
 - 1. RQA Alerts Doesn't Have Necessary Input Files
 - iii. Body:
 - iv. Dolly Darling doesn't have all the necessary input files to start today's automation run of RQA alerts. Please check
 \mhcetnas01\MHC_CBS\CBS Automation\Cerner Inbound and restart job when all files are present
- e. Process Files for Scope
 - i. Map Alert Company Id to health plan
 - ii. Map all Payer Mnemonics to a health plan
 - iii. If any mapping doesn't have a health plan:
 - 1. Log Exception
 - 2. Next Account Number
 - iv. Does Account Number NOT have at least one in scope alert?
 - 1. In Scope Alerts = 52,53,16,1461,1309
 - 2. Log Exception
 - 3. Next Account Number
 - v. For the In Scope Alerts check status to make sure the status = "Assign to User"
 - vi. If no alerts have a status of "Assign to User":
 - 1. Log Exception
 - 2. Next Account Number
 - 3. Note: If at least one alert DOES have a status of "Assign to User" then we can proceed with just the Alerts with that status
 - vii. If Account Number has a Bill Status of Billed
 - 1. Log Exception
 - 2. Next Account Number
 - viii. If an account Number's service date is more than 30 days into the future
 - 1. Log Exception
 - 2. Next Account Number
 - ix. Create Queue Variables
 - x. Add Account Number to UiPath Queue

Performer

1. Get next queue transaction from UiPath
2. Query Data Fabric to get historical processing results for the account number
 - a. Need to count successes and failures
3. If an Account Number has been successfully worked before:
 - a. Log Exception
 - b. Next Transaction
4. If an Account Number has failed more than 3 times, then we log an exception and move to the next transaction
 - a. Log Exception
 - b. Next Transaction
5. If an Account Numbers Service date is older than automation run date – 3 days
 - a. Search for Account Number in Cerner
 - i. If failure to find Account Number in Cerner:
 1. Log Exception
 - ii. Move to next Transaction in UiPath Queue
 - b. Write Encounter Comment
 - c. Log Output, move to next Transaction
6. We now need to loop over all the in-scope alert list from the UiPath Queue
 - a. In scope alerts are 52,53,16,1309 and 1461
 - b. The order of the alerts we work with matters, ideally, we work the alerts in the following order
 - i. 1461,1309,52,53,16
 - c. Is Alert 1461?
 - i. If the bot has NOT already read the alert health plan full response:
 1. Fast Client Switch to St Luke's.docx
 2. Open Enterprise Alerts Queue.docx
 3. Search for Account Number in eCare
 - a. If failure to find account number, retry up to a max of 3 attempts
 - b. If after max retry still getting failure:
 - i. Log Exception
 - c. Next Transaction
 4. Read eCare Full Response
 5. Extract Subscriber ID
 - a. If Available also extract Group Number and Medicare HIC Number
 - ii. If the bot has already read the alert health plan full response:
 1. Extract Subscriber ID
 - a. If Available also extract Group Number and Medicare HIC Number

- iii. If Bot did not extract at least Subscriber ID value:
 - 1. Log Exception
 - 2. Next Alert
- iv. Search for Account Number in Cerner
 - 1. If failure to find Account Number in Cerner:
 - a. Log Exception
 - 2. Move to next Transaction in UiPath Queue
- v. Locate Alert Payer in Cerner
 - 1. If Payer not present on Account:
 - a. Log Exception
 - 2. Next Alert
- vi. Change Health Plan Details and Remove Old Payer
 - 1. Note we need to save the health plan name that is originally there for a comment later in the flow
- vii. Write Encounter Comment
- viii. Initiate Eligibility and Complete Account
- ix. Next Alert or Log Output
- x. Log Details:
 - 1. Note: We only want to log details once per account number, so we will need to combine outcome values for all the in scope alerts that were resolved. The In Scope and Result values should be the same, but the outcome can vary based on the alerts worked.
 - 2. In Scope = Yes
 - 3. Result = Success
 - 4. Outcome = Resolved Alert 1461, changed health plan from {ORIGINAL HEALTH PLAN} to {NEW HEALTH PLAN}
- d. Is Alert 1309?
 - i. Search for Account Number in Cerner
 - 1. If failure to find Account Number in Cerner:
 - a. Log Exception
 - 2. Move to next Transaction in UiPath Queue
 - ii. Read Medical Service
 - iii. Is Medical Service NOT Emergency Medicine?
 - 1. Log Exception
 - 2. Next Alert
 - iv. Locate Alert Payer in Cerner
 - 1. If Payer not present on Account:
 - a. Log Exception
 - 2. Next Alert
 - v. If more than one payer on the account
 - 1. Remove Alert Payer
 - vi. If only one payer on the account

1. Change Payer to Self Pay
- vii. Write Encounter Comment
- viii. Initiate Eligibility and Complete Account
- ix. Next Alert or Log Output
- x. Log Details:
 1. Note: We only want to log details once per account number, so we will need to combine outcome values for all the in scope alerts that were resolved. The In Scope and Result values should be the same, but the outcome can vary based on the alerts worked.
 2. In Scope = Yes
 3. Result = Success
 4. Outcome = Resolved Alert 1309, Removed {ORIGINAL HEALTH PLAN}
- e. If Alert is 52:
 - i. If the bot has NOT already read the alert health plan full response:
 1. Fast Client Switch to St Luke's.docx
 2. Open Enterprise Alerts Queue.docx
 3. Search for Account Number in eCare
 - a. If failure to find account number, retry up to a max of 3 attempts
 - b. If after max retry still getting failure:
 - i. Log Exception
 - c. Next Transaction
 4. Read eCare Full Response
 5. Extract Member ID Number
 - ii. If the bot has already read the alert health plan full response:
 1. Extract Member ID Number
 - iii. If the bot failed to read a Member ID Number:
 1. Log Exception
 2. Move to Next Alert (if applicable)
 - iv. Search for Account Number in Cerner
 1. If failure to find Account Number in Cerner:
 - a. Log Exception
 2. Move to next Transaction in UiPath Queue
 - v. Locate Alert Payer in Cerner
 1. If Payer not present on Account:
 - a. Log Exception
 2. Next Alert
 - vi. Update Subscriber Member Number Field
 1. Note we need to save the original value so that it can be used in a comment later in the flow
 - vii. Write Encounter Comment

- viii. Initiate Eligibility and Complete Account
- ix. Next Alert or Log Output
- x. Log Details:
 - 1. Note: We only want to log details once per account number, so we will need to combine outcome values for all the in scope alerts that were resolved. The In Scope and Result values should be the same, but the outcome can vary based on the alerts worked.
 - 2. In Scope = Yes
 - 3. Result = Success
 - 4. Outcome = Resolved Alert 52, changed subscriber id from {ORIGINAL VALUE} to {NEW VALUE}
- f. If Alert is 53:
 - i. If the bot has NOT already read the alert health plan full response:
 - 1. Fast Client Switch to St Luke's.docx
 - 2. Open Enterprise Alerts Queue.docx
 - 3. Search for Account Number in eCare
 - a. If failure to find account number, retry up to a max of 3 attempts
 - b. If after max retry still getting failure:
 - i. Log Exception
 - c. Next Transaction
 - 4. Read eCare Full Response
 - 5. Extract Group Number
 - ii. If the bot has already read the alert health plan full response:
 - 1. Extract Group Number
 - iii. If the bot failed to read a Group Number:
 - 1. Log Exception:
 - 2. Move to Next Alert (if applicable)
 - iv. Search for Account Number in Cerner
 - 1. If failure to find Account Number in Cerner:
 - a. Log Exception
 - 2. Move to next Transaction in UiPath Queue
 - v. Locate Alert Payer in Cerner
 - 1. If Payer not present on Account:
 - a. Log Exception
 - 2. Next Alert
 - vi. Update Group Number Field
 - 1. Note we need to save the original value so that it can be used in a comment later in the flow
 - vii. Write Encounter Comment
 - viii. Initiate Eligibility and Complete Account
 - ix. Next Alert or Log Output

- x. Log Details:
 - 1. Note: We only want to log details once per account number, so we will need to combine outcome values for all the in scope alerts that were resolved. The In Scope and Result values should be the same, but the outcome can vary based on the alerts worked.
 - 2. In Scope = Yes
 - 3. Result = Success
 - 4. Outcome = Resolved Alert 53, changed group number from {ORIGINAL VALUE} to {NEW VALUE}

- g. Is Alert 16?
 - i. Search for Account Number in Cerner
 - 1. If failure to find Account Number in Cerner:
 - a. Log Exception
 - 2. Move to next Transaction in UiPath Queue
 - ii. Read Patient Relationship to Guarantor
 - 1. Note we need to save the original value so that we can use it later for a comment in the flow
 - iii. If Relationship in ('Guardian','Other','Ward','Unknown'):
 - 1. Log Exception
 - 2. Next Alert
 - iv. Change Patient Relationship to Guarantor to "Self"
 - v. Write Encounter Comment
 - vi. Initiate Eligibility and Complete Account
 - vii. Next Alert or Log Output
 - viii. Log Details:
 - 1. Note: We only want to log details once per account number, so we will need to combine outcome values for all the in scope alerts that were resolved. The In Scope and Result values should be the same, but the outcome can vary based on the alerts worked.
 - 2. In Scope = Yes
 - 3. Result = Success
 - 4. Outcome = Resolved Alert 16, changed patient's relationship to guarantor from {ORIGINAL VALUE} to {NEW VALUE}

2.2 Applications

The table below lists the applications being utilized by the automation

Application	Comments / Access Needs
-------------	-------------------------

Access Management Office	
eCare Next	

2.3 Process Map

The diagram below contains the steps the automation will take (the 'to-be' process map)

2.4 Toolkit documentation

The table below provides a list of toolkits and links to the completed ROI, planning toolkits, and any additional design documentation

Toolkit	Link to documentation	Comments
Workflow Visio	<u>Registration Eligibility Alert Resolution.vsd</u>	
Plan Tool Kit	<u>Registration Eligibility Alert Resolution Plan Toolkit.xlsx</u>	

2.5 Out of scope

The table below provides a list of automation scenarios or account populations that are out of scope or unobtainable

Scenario / Population	Rationale
Not All Health Plans Mapped	We map company id and payer mnemonics to a health plan list, if we don't have all values mapped we can't do the automation
No In Scope Alerts	We only are programming 5 alerts for resolution if we don't have one of those five at least we can't run the automation
Bill Status Out of Scope	We only want to resolve RQA alerts when the account/health plan doesn't have a claim sent to the payer. We will do this by getting a new export file on the network drive and check the account/health plan.
Alert Status Check	We need to work only unresolved alerts, so if we don't have any inscope alerts with a status of "Assign to User" then the account is out of scope

Service Date > Automation Run + 30	We don't want to work on any account where the scheduled service date is greater than 30 days into the future. These will be marked as "Skip" in the log output, field = result.
Service Date < Automation Run - 3	We will be placing a note on accounts that meet this criteria, but they don't really count as an In Scope Volume.
Account Number worked Successfully on a previous run	If an account number has been worked successfully on a different automation run, then we don't want to work it again, so its out of scope
Account Number Failed >= 3 times	There will always be transient errors, so if an account number has been worked and failed up to three times, the next time we see its out of scope
Alert 1309, Medical Service not in scope	The Medical Service for an account with alert 1309 needs to be Emergency Medicine, if its any other value the account is out of scope
Alert 16, Patient/Guarantor Relationship	If the patient's relationship to guarantor is one of the following the account is out of scope. Guardian, Unknown, Other, Ward

III. APPROVAL

3.1 Approval

Upon review and approval of this Process Definition Document (PDD), I, [ENTER Process Owner], on [ENTER today's date], hereby approve this plan for development.

I acknowledge that while this automation will experience downtime, the ultimate responsibility for the process remains with my department. I commit to collaborating with the automation team to maintain the automation and meet our uptime goal. This commitment extends to ongoing responsibilities, including but not limited to: testing, implementing enhancements, handling break-fix scenarios, managing application upgrades, and supporting process changes or improvements as required.

3.2 Future Initiatives

Below is a list of future operational, technical, or known application changes that may impact the automation

- ENTER
- ENTER

3.3 Downtime Plans

Below are the actions the operational team will take in the event of automation downtime

Scenario	Action (taken by operations)	Contact
Level 1		

<i>Downtime = 1-2 days</i>		
Level 2 <i>Downtime = 2-7 days</i>		
Level 3 <i>Downtime = 7-14 days</i>		
Level 4 <i>Downtime = Indefinitely</i>		

3.4 Revisions

Complete the table below for any document revisions

Rev. #	Date	Section/page	Revision summary	Author