**L3 Engine User Manual**

Comprehensive Guide for Production Issue Resolution

Author -

Shivansh Singh Bhadouria

Location - Delhi GCC

Date - 26th November 2025

CONTENTS

1. Summary

2. System Overview

3. Getting Started

4. Module 1: Log Extraction Engine

5. Module 2: Log Parser Engine

6. Module 3: RCA Engine (Root Cause Analysis)

7. Data Flow & Integration

8. Troubleshooting Guide

9. Best Practices

10. Appendices

1. Summary

L3 Engine is your one-stop solution for investigating and resolving airline production issues. Whether you're dealing with missing passengers, data discrepancies, or system failures, this tool streamlines your investigation process from start to finish.

What You'll Accomplish With L3 Engine

As an Ops / L2 Analyst, you will:

* Quickly extract flight logs without manual searches
* Identify passenger data issues within minutes
* Generate comprehensive reports for L3 escalation
* Resolve 80% of common data discrepancy issues independently

As an L3 Analyst, you will:

* Perform deep-dive analysis with complete data visibility
* Compare multiple data sources to find root causes
* Generate executive-level reports with detailed findings
* Reduce investigation time from hours to minutes

1. System Overview

Your Investigation Journey

When you encounter a production issue, L3 Engine guides you through a logical, step-by-step process:

❶INCIDENT REPORTED →

Flight WF123

has missing

passengers

❷ EXTRACT LOGS →

Get all system

logs for this

flight

❸ PARSE DATA →

Find passenger

messages in

the logs

❹ ANALYZE ISSUES →

Compare input

vs output data

automatically

❺ GENERATE REPORT

Create Excel

report with

findings

What Each Step Accomplishes

## Step 1: Log Extraction Engine

Your Problem: "I need logs for flight WF123 but don't know where they are"

L3 Engine Solution: Automatically pulls ALL relevant logs from all available nodes (Prod/QA/Dev)

using Azure Devops.

Result: Complete log package ready for analysis in 5 minutes

## Step 2: Log Parser Engine

Your Problem: "These logs are huge and I can't find the passenger data"

L3 Engine Solution: Extracts only API/PNR messages relevant to your investigation

Result: Clean, structured passenger data files ready for comparison

## Step 3: RCA Engine

Your Problem: "I need to compare input vs output data to find what went wrong"

L3 Engine Solution: Automatically compares data sources and highlights discrepancies

Result: Clear identification of missing passengers, duplicates, and processing errors

Common Scenarios L3 Engine Handles

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Traditional Approach** | **L3 Engine Approach** | **Time Saved** |
| Missing Passengers | Manual log search (2-3 hours) | Automated extraction & analysis (15 minutes) | 85% |
| DataDiscrepancies | Manual file comparison (1-2 hours) | Automated comparison with reports(10 minutes) | 90% |
| System Issues | Multiple tool investigation (3-4 hours) | Unified analysis workflow (30 minutes) | 85% |
| ManagementReports | Manual Excel creation (1hour) | Auto-generated professional reports(2 minutes) | 95% |

Data Types You'll Work With

**API Data**: Passenger manifest information for security and immigration.

**PNR Data**: Passenger Booking and reservation details.

**API Crew** Data: Crew manifest information.

1. Quick Start Guide

## First Time Setup (One-time only)

## Step 1: Check Your Computer

Before using L3 Engine, ensure you have:

* Windows 10 or Windows 11
* At least 4GB RAM (8GB recommended for large investigations)
* 2GB free disk space
* Network access to Azure DevOps (ask your IT team if unsure)

## Step 2: Launch the Application

# **Log Extraction Engine**: Extract logs from Azure pipelines

# **Log Parser Engine**: Parse messages from existing log files

# **RCA Engine: Analyse** : Passenger data discrepancies

Choose the module you need for your current investigation task.

## Step 3: First-Time Configuration - Log Extraction Engine

Click the "⚙ Configure Azure" button to set up your Azure DevOps connection:

1. Enter your organization URL (example: https://dev.azure.com/YourCompany)
2. Enter your project name
3. Create and enter a Personal Access Token (PAT)

(Azure URL - https://sita-pse.visualstudio.com/\_usersSettings/tokens)

1. Test the connection
2. Save your settings

💡 Tip: Contact your Azure DevOps administrator if you need help with these settings.

## 3.2 Your First Investigation

## Scenario: You've been told "Flight WF123 on November 20th has missing passengers"

What you'll do:

**1. Launch Log Extraction Engine**

**2. Start with Log Extraction:**

* Enter flight number: WF123
* Select date: November 20, 2024
* Click "Extract Logs"
* Wait for completion (usually 1-2 minutes)

**3. Launch Log Parser Engine:**

* Select the extracted logs folder
* Enter same flight details
* Choose data type API or PNR
* Click "Process Logs"

**4. Launch RCA Engine:**

* Select the parsed data folder
* Choose data type API or PNR
* Click "Process"
* Review results and generate Excel report

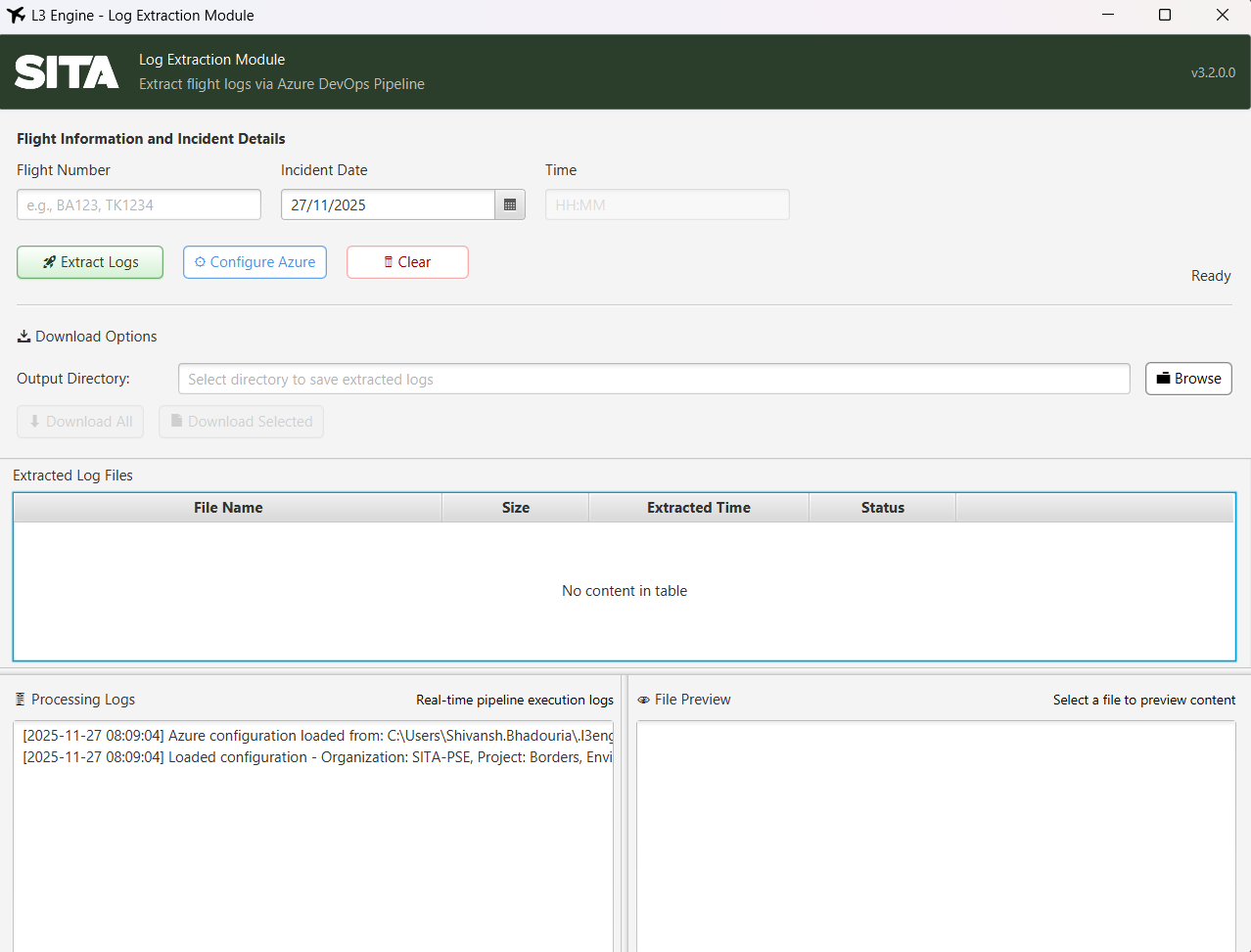
**Expected Result**: You'll have a complete analysis showing exactly which passengers are missing and why.

1. Log Extraction Engine - Getting Flight Logs
   1. What This Module Does for You

Instead of hunting through multiple systems and asking different teams for log files, the Log Extraction Engine automatically gets ALL the logs you need for a specific flight from Azure DevOps in one click.

**Before L3 Engine**: Get PAM Token → Start Jump Server → Open All nodes → Write manual commands → Copy files to separate location on node → Download to local

**With L3 Engine**: Enter flight details → click Extract → get complete log package in 2 minutes



* 1. How to Use It

**Step 1: Enter Flight Information**

**Flight Number** ✈️

* **What to enter**: The flight number exactly as it appears in the incident report
* **Examples**: WF123 , AI101 , 6E2024 , SG+8475
* ⚠️ **Important**:
  + Must be at least 3 characters
  + Can include + symbol between carrier code and flight number (like SG+8475)
  + Not case-sensitive (wf123 works same as WF123)

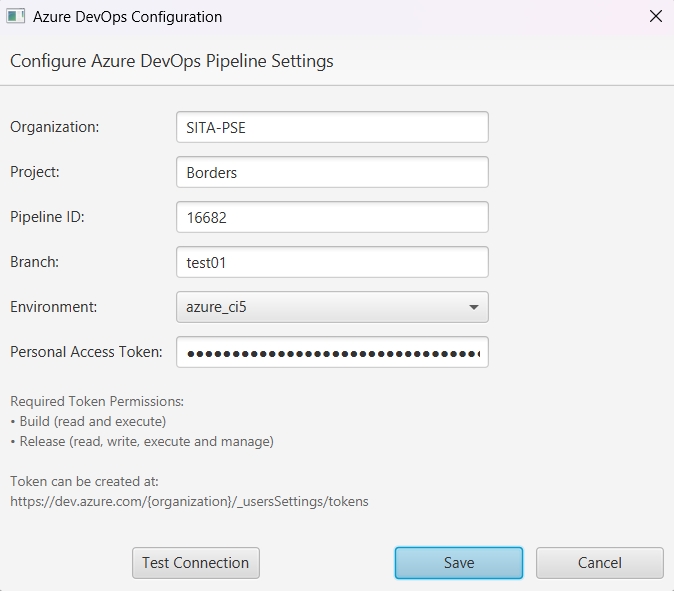
**Incident Date** 📅

* **What to enter**: The date when the data was sent to our application (not the departure date)
* **How to enter**: Click the calendar icon and select the date
* ⚠️ **Important**: You MUST select a date - the extraction engine won't work without it.

**Incident Time** ⏰ (Currently Non-Functional)

* **What to enter**: If you know the specific time the issue occurred
* **Format**: 24-hour time like 14:30 or 09:15
* **If unsure**: Leave blank (system will use 00:00)

**Step 2: Configure Azure Connection (First Time Only)**

****

Click "⚙ Configure Azure" if this is your first time or if you see connection errors.

**You'll need these details from your IT team**:

1. **Organization URL**: Usually looks like “sita-pse.visualstudio.com”
2. **Project Name**: The Azure DevOps project name (ask your team lead or L3)
3. **Personal Access Token**: A secure password for Azure access
4. **Environment**: Usually "Production" for live issues

**How to get a Personal Access Token**:

1. Go to Azure DevOps in your browser (https://sitapse.visualstudio.com/\_usersSettings/tokens)
2. Click the icon besides your profile picture → Personal Access Tokens
3. Click "New Token"
4. Give it a name like "L3Engine"
5. Set expiration to 30 days (recommended)
6. Give it all permission under "Build" and "Release" sections permissions
7. Copy the token and paste it in L3 Engine configuration wizard.

**Step 3: Extract the Logs**

1. Click the **"Extract Logs"** button
2. Watch the progress bar and status messages
3. **Typical wait time**: 1-2 minutes depending on log size
4. When complete, you'll see a list of extracted files

**What you'll see during extraction**:

✅ "Starting pipeline..." - System is connecting to Azure

✅ "Pipeline running..." - Logs are being collected

✅ "Downloading files..." - Files are being transferred to your computer

✅ "Extraction complete" - Ready for next step

**Step 4: Download Your Files**

You'll see a table with all the log files found:

* **File Name**: The name of each log file
* **Size**: How big each file is
* **Time**: When it was extracted
* **Status**: Whether the file is ready

Download Options:

* **Download All**: Gets everything (recommended for thorough investigation)
* **Download Selected**: Click specific files first, then download only those.

**NOTE:** The user will be able to preview files only once they have been downloaded

* 1. What Can Go Wrong and How to Fix It

❌ **"Configuration Error - Please configure Azure settings"**

**What happened**: Azure connection isn't set up

**How to fix**:

1. Click "⚙ Configure Azure"
2. Enter your Azure details (see Step 2 above)
3. Click "Test Connection"
4. If it fails, double-check your organization URL and token

❌ **"Validation Error - Please enter a flight number"**

**What happened**: Flight number field is empty

**How to fix**: Enter at least 3 characters in the flight number field

❌ **"Validation Error - Flight number must be at least 3 characters"**

**What happened**: Flight number is too short

**How to fix**: Check the flight number - it should be like WF123, not just WF

❌ **"Pipeline execution failed"**

**What happened**: Something went wrong in Azure

**How to fix**:

1. Check if the flight data was actually received on that date.
2. Try again in a few minutes (might be temporary).
3. Regenerate expired PAT token.
4. If it keeps failing, contact your Azure DevOps team.

❌ **"No logs found for flight"**

**What happened**: System couldn't find any logs for your flight

**Possible reasons**:

* Flight number spelled wrong
* Wrong date selected
* Flight didn't operate that day
* Logs not available yet (for very recent flights)

**How to fix**:

1. Double-check flight number spelling
2. Verify the flight date with operations
3. Try a slightly different date if unsure
4. For recent flights, wait a few hours and try again
   1. Tips for Success

✅ **Double-check flight details**: Most errors come from typos in flight numbers or wrong dates

✅ **Use the preview feature**: Look at file contents by downloading individual files.

✅ **Save your work**: The system remembers your Azure configuration, so you only need to set it up once

✅ **Organize your files**: Downloaded files go to your downloads folder by default, but you can choose a different location

✅ **Be patient**: Large investigations can take up to 10 minutes to extract all logs

* 1. What You'll Get

After successful extraction, you'll have:

* **Application Logs**: Detailed system operation logs
* **Error Logs**: Any errors that occurred during flight processing
* **Performance Logs**: System timing and performance data
* **Debug Logs**: (If turned ON) Detailed troubleshooting information

These files are automatically organized by date and flight number, ready for the next step: Log Parsing.

5. Log Parser Engine - Finding Passenger Data in Log

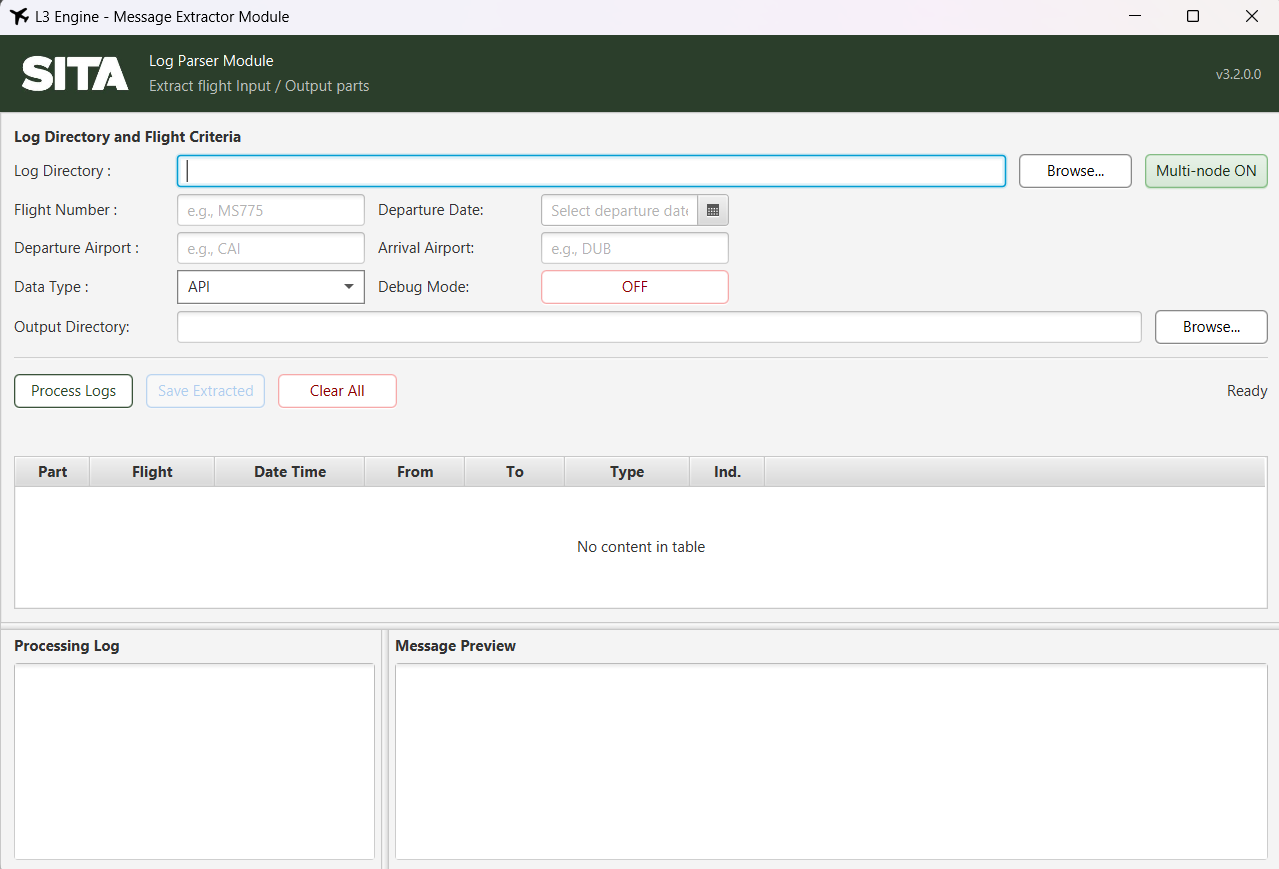
Files

* 1. What This Module Does for You

After you've extracted logs, they're usually huge files full of technical information. The Log Parser Engine finds and extracts just the passenger-related messages (API or PNR data) that you need for your investigation.

**Before L3 Engine**: Open massive log files → search manually for hours → copy/paste relevant sections → miss important data

**With L3 Engine**: Point to log folder → select data type → get clean passenger data files in minutes

****

* 1. How to Use It

**Step 1: Select Your Log Files**

**Log Directory** 📁

* **What to select**: The folder containing the log files you extracted (or received from someone else)
* **How to select**: Click "Browse" and navigate to your log folder
* ⚠️ **Important**: The folder must exist and you must have permission to read it

💡 **Tip**: If you just used the Log Extraction Engine, browse to your extraction/downloads folder and provide that path to the Log Parser Engine.

**Step 2: Enter Flight Details**

Flight Number ✈️

* **What to enter**: Same flight number as your investigation
* **Why needed**: Filters out messages from other flights in the logs

Departure Date 📅

* **What to enter**: The flight's departure date
* **Why needed**: Logs might contain multiple days of data

Departure Airport 🛫 (Optional but strictly recommended)

* **What to enter**: 3-letter airport code like DEL , BOM , BLR
* **Why helpful**: Further narrows down the data to your specific flight.

Arrival Airport 🛬 (Optional but strictly recommended)

* **What to enter**: 3-letter destination airport code
* **Why helpful**: Confirms you're getting the right flight data

**Step 3: Choose Your Data Type**

**A screenshot of a computer

AI-generated content may be incorrect.**

**API (Advanced Passenger Information)**

* **What it is**: Passenger manifest data with names, passport numbers, nationalities
* **When to use**:
  + Error reported in the data.
  + Missing passenger investigations
  + Complete message drops.
* **What you'll get**: Clean list of all passengers with their details

**PNR (Passenger Name Record)**

* **What it is**: Booking and reservation data with ticket information
* **When to use**:
  + Error reported in the data.
  + Missing passenger investigations
  + Complete message drops.
* **What you'll get**: Passenger manifest with booking details and ticket information

**Step 4: Advanced Options (Usually Not Needed)**

**Multi-Node Mode**

* **What it does**: Combines logs from distributed systems (n1, n2, n3 folders)
* **When to use**: Only if your log folder has subfolders named n1/, n2/, n3/
* **Default**: ON.

**Debug Mode**

**NOTE** - DO NOT TURN ON. STRICTLY FOR TROUBLESHOOTING PURPOSES.

* **What it does**: Shows detailed processing information
* **When to use**: Only if parsing fails and you need to troubleshoot
* **Warning**: Makes processing slower and creates more log messages

**Step 5: Process Your Logs**

1. Click **"Process Logs"**
2. Watch the progress bar and status messages
3. **Typical wait time**: 2-5 minutes for normal-sized logs
4. When complete, you'll see extracted messages in the results table

**What you'll see during processing**:

* "Scanning log files..." - Finding relevant files
* "Extracting messages..." - Pulling out passenger data
* "Processing complete" - Ready to review results

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Understanding Your Results

**Results Table**

**A screenshot of a computer

AI-generated content may be incorrect.**

**What each column means**:

* **Part**: Some messages are split into parts (1, 2, 3, etc.)
* **Flight**: Confirms this is your flight
* **Date**: When the flight was scheduled
* **Departure**: Origin airport (should match what you expected)
* **Arrival**: Destination airport
* **Type**: API or PNR (what you selected)
* **Indicator**: C for middle/initial parts. F for last part.

**Message Preview**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **Click any row** in the results table to see the full message
* **Raw format**: This is the actual airline industry standard format
* **Don't worry**: You don't need to understand this format - the RCA Engine will process it
  1. Save Your Work

**Output Options**

1. **Choose output folder**: Click "Browse Output Directory"
2. **Click "Save"**: Saves all extracted messages as text files
3. **File names**: Automatically named like WF123\_2024-11-20\_API.txt

💡 **Tip**: Save to a folder you'll remember - you'll need these files for the RCA Engine

* 1. What Can Go Wrong and How to Fix It

❌ **"No messages extracted from logs"**

**What happened**: System couldn't find any passenger data

**Possible causes**:

* Wrong data type selected (try switching API ↔ PNR)
* Flight number typo
* Wrong date
* Logs don't contain passenger data

*How to fix*:

1. Double-check flight number spelling
2. Try the other data type (API vs PNR)
3. Check if logs actually contain passenger data (ask the person who gave you the logs)

❌ **"Directory not found" or "Access denied"**

**What happened**: Can't read the log folder

*How to fix*:

1. Check folder path is correct
2. Make sure folder exists
3. Check you have permission to read the folder

❌ **"Multi-node consolidation failed"**

**What happened**: System expected n1/, n2/, n3/ folders but didn't find them

*How to fix*:

1. Turn OFF Multi-node mode if you don't have these subfolders
2. If you do have n1/, n2/, n3/ folders, check they contain log files

❌ **"Message parsing errors"**

**What happened**: A bug in the application.

*How to fix*:

1. A possible edge case that needs to be informed to Devs to include in next release.
   1. Tips for Success

✅ **Start with API data type**: It covers most passenger investigation needs

✅ **Check the results table**: Make sure you see messages for your flight before saving

✅ **Use the preview**: Click on messages to verify they contain the data you expect

✅ **Save your work**: Always save the extracted messages - you'll need them for analysis

✅ **Organize files**: Keep extracted messages in a folder you'll remember

* 1. What You'll Get

After successful processing and saving:

* **Clean message files**: Just the passenger data, no system noise
* **Structured format**: Ready for analysis in the RCA Engine
* **Flight-specific data**: Only messages relevant to your investigation
* **Time-stamped files**: Easy to track what you processed when

These files are now ready for the final step: Root Cause Analysis in the RCA Engine.

6. RCA Engine - Finding What Went Wrong with

Passenger Data

6.1 What This Module Does for You

The RCA Engine is where you find answers. It compares different files of passenger data to show you exactly what went wrong - missing passengers, duplicates, data corruption, or processing failures.

**Your typical investigation**:

* "Flight WF123 should have 150 passengers, but only 147 were processed. Where are the missing 3?"
* "Passenger John Smith appears twice in the manifest. Why?"
* "The input file had 200 passengers, but output shows 205. What happened?"

**A screenshot of a computer

AI-generated content may be incorrect.**

6.2 How to Use It

**Step 1: Select Your Data Folder**

**What folder to choose**:

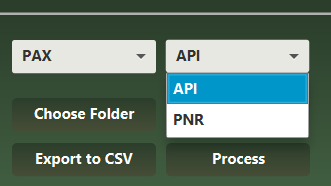
* The folder containing passenger data files
* Usually from the Log Parser Engine output
* Or provided by operations team

**Expected file names for API & PNR data**:

* input\*.txt - Original API passenger data sent to the system
* output\*.txt - API passenger data after processing

💡 **Tip**: Don't worry about exact file names - the system will find passenger data files automatically based on your data type selection

**Step 2: Choose What Type of Data to Analyse**

****

A screenshot of a computer

AI-generated content may be incorrect.

**Data Type Options**:

* **API**: Passenger manifest data with security/immigration focus
  + Use for: Missing passengers, document issues, security screening problems
  + Contains: Names, passport numbers, nationalities, document details
* **PNR**: Booking and reservation data with commercial focus
  + Use for: Booking system issues, ticketing problems, seat assignments
  + Contains: Booking references, ticket numbers, seat assignments, payment info
* **Crew**: Flight crew data analysis
  + Use for: Crew scheduling issues, crew manifest problems

**Record Type Options**:

* **Input**: Analyze only the original data that went into the system
* **Output**: Analyze only the final processed data

🎯 **For most investigations**: Choose **API** for data type, and select **Input** first to understand what data came into the system, then run **Output** separately to see what was processed.

**Step 3: Process and Analyse**

1. Click **"Process"**
2. Wait for analysis to complete (usually 1-2 minutes)
3. Review the results in the summary panel and data tables

6.3 Understanding Your Results

**Summary Statistics - Your Investigation Dashboard**

**A green screen with white text

AI-generated content may be incorrect.**

**What each number tells you**:

**Total Input Passengers**: How many passengers were originally supposed to be processed

* *Example: 150 passengers booked on the flight*

**Total Unique Input Passengers**: Same as above, but removing any duplicates from the input

* *Example: 148 unique passengers (2 were duplicates in the booking system)*

**Total Output Passengers**: How many passengers actually got processed

* *Example: 147 passengers processed*

**Dropped Passengers**: Passengers in input but missing from output

* *Example: 1 passenger missing = Your investigation focus*

**Duplicate Passengers**: Passengers appearing multiple times in the data

* *Example: 2 duplicates = Data quality issue to investigate*

**New PNRs**: Passengers in output but not in input (rare but important) Only in PNR data

* *Example: 0 new passengers (good) or 3 new passengers (needs investigation)*

**Flight Information**

**A green background with white text and a plane

AI-generated content may be incorrect.**

Automatically extracted and displayed:

* **Flight Number**: Confirms you're analyzing the right flight
* **Departure Date**: Flight date and time
* **Route**: Origin → Destination airports

6.4 Detailed Analysis Tables

**Input Passengers Table - "What Should Have Been Processed"**

**A screenshot of a computer

AI-generated content may be incorrect.**

What you'll see:

* Every passenger from the original input data
* Names, dates of birth, passport numbers
* Source file information
* Duplicate count (if any)

How to use it:

* Review for data quality issues
* Check for obvious duplicates or errors
* Use search box to find specific passengers

**Output Passengers Table - "What Actually Got Processed"**

**A screenshot of a computer

AI-generated content may be incorrect.**

What you'll see:

* Every passenger that was successfully processed
* Same format as input table
* Shows final state after system processing

How to use it:

* Compare with input to see changes
* Verify expected passengers are present
* Look for unexpected additions

**Dropped Passengers Table - "The Missing Ones"**

****

**What you'll see**: Passengers who were in input but missing from output

**This is usually your main investigation focus**:

* Why were these passengers dropped?
* System error? Data format issue? Processing rule?
* Were they legitimately excluded (no-shows, cancellations)?

**Investigation steps**:

1. Note the passenger names and details

2. Check if there's a pattern (same nationality, document type, etc.)

3. Look for error logs around the time these should have been processed

**Duplicate Passengers Table - "The Repeats"**

**A screenshot of a computer

AI-generated content may be incorrect.**

**What you'll see**: Passengers appearing multiple times

**Common causes**:

* Booking system errors
* Multiple ticket versions
* Data synchronization issues
* Name variations (JOHN SMITH vs SMITH JOHN)

6.5 Search and Investigation

**Smart Search Function**

**How to search**:

* Type in the search box above any table
* Search works across all columns
* Not case-sensitive
* Finds partial matches

**What to search for**:

* Passenger names
* Passport numbers
* Specific dates
* Document types

6.6 Generate Your Investigation Report

**Excel Report Creation**

**Click "Export to Excel"** buttonto generate a professional investigation report.

**What you'll get**:

* **Summary sheet**: All key statistics and findings
* **Input Passengers**: Complete input data
* **Output Passengers**: Complete output data
* **Dropped Passengers**: Your main investigation findings
* **Duplicate Passengers**: Data quality issues
* **Charts**: Visual representation of key metrics

**The report is ready for**:

* Management presentation
* L3 analyst handoff
* Incident documentation
* Process improvement discussions

6.7 Common Investigation Scenarios

**Scenario 1: Missing Passengers**

**Situation**: "Flight should have 150 passengers, only shows 147"

**What RCA Engine shows you**:

* Summary: 150 input → 147 output = 3 dropped
* Dropped Passengers table: Shows exactly which 3 passengers
* Pattern analysis: Same nationality? Same document type?

**Your next steps**:

* Check error logs for these specific passengers
* Look for system processing rules that might exclude them
* Verify if they were legitimately removed (no-shows, etc.)

**Scenario 2: Duplicate Passengers**

**Situation**: "Passenger John Smith appears twice in the manifest"

**What RCA Engine shows you**:

* Duplicate Passengers table: Shows John Smith with count = 2
* Details: Shows both entries with their differences
* Source: Shows which input files contained the duplicates

**Your next steps**:

* Check booking system for duplicate reservations
* Look for data synchronization issues
* Verify if this is the same person with multiple bookings

**Scenario 3: Data Corruption**

**Situation**: "Names are showing up garbled or incomplete"

**What RCA Engine shows you**:

* Input vs Output comparison showing data changes
* Unusual characters or missing data in passenger names
* Pattern of corruption (specific fields affected)

**Your next steps**:

* Check data encoding issues
* Look for system processing errors
* Verify data transformation rules

6.8 What Can Go Wrong and How to Fix It

❌ **"No passenger data found"**

**What happened**: System couldn't find expected passenger files

**How to fix**:

1. Check folder contains files like api\_input.txt, api\_output.txt
2. Try different Data Type (API → Passenger → Crew)
3. Verify file permissions
4. Look at "Files Processed" list to see what was found

❌ **"Excel export failed"**

**What happened**: Can't create the Excel report

**How to fix**:

1. Close any open Excel files with the same name
2. Check you have permission to write to the output folder
3. Make sure there's enough disk space
4. Try a different output folder

❌ **"High duplicate count but no obvious duplicates"**

**What happened**: System is detecting duplicates you don't see

**How to investigate**:

1. Look at the Duplicate Passengers table carefully
2. Check for subtle name variations (JOHN vs JOHN.)
3. Look for same passenger with different document numbers
4. Check date format differences

6.9 Tips for Successful Investigation

* **Focus on the Summary Statistics first**: They tell you the big picture
* **Use the Dropped Passengers table as your starting point**: This is usually where the answers are
* **Search is your friend**: Use it to quickly find specific passengers or patterns
* **Export to Excel**: Even if you solve the issue, export the report for documentation
* **Look for patterns**: If 5 passengers are missing, what do they have in common?

6.10 What You'll Accomplish

After using the RCA Engine, you'll have:

* **Exact count**: Precise numbers of missing/duplicate passengers
* **Specific names**: Identity of problematic passenger records
* **Pattern analysis**: Understanding of what types of issues occurred
* **Professional report**: Excel document ready for stakeholders
* **Investigation leads**: Clear next steps for deeper analysis

This gives you everything needed to either resolve the issue or escalate with complete information to L3 analysts or development teams.

7. Data Flow & Integration

* 1. End-to-End Workflow

Azure DevOps → Log Extraction → Raw Logs → Log Parser → Structured Data →

RCA Engine → Analysis Reports

* 1. Data Format Standards
     1. Log File Formats
* **Source**: Various application log formats
* **Structure**: Time-stamped entries with message content
* **Encoding**: UTF-8 text encoding
* **Size**: Can range from KB’s to MB’s depending on flight complexity
  + 1. Extracted Message Formats
* **API Messages**: PAXLST (EDIFACT) format for passenger information
* **PNR Messages**: PNRGOV (EDIFACT) format for booking data
* **Structure**: Segmented messages with defined IATA syntax
  + 1. Analysis Data Formats
* **Input Format**: Structured text files with passenger records
* **Processing Format**: Normalized internal representation
* **Output Format**: Excel workbooks with multiple sheets
  1. File Organization Best Practices
     1. Directory Structure

Investigation\_Root/

├── Logs/

│ ├── Flight\_WF123\_2024-11-20/

│ │ ├── Raw\_Logs/

│ │ └── Extracted\_Messages/

├── Analysis/

│ ├── WF123\_API\_Analysis/

│ └── WF123\_PNR\_Analysis/

└── Reports/

├── WF123\_Investigation\_Report.xlsx

└── WF123\_Summary.pdf

7.3.2 File Naming Conventions

* **Flight Logs**: FlightNumber\_Date\_LogType.txt
* **Extracted Messages**: FlightNumber\_Date\_MessageType.txt
* **Analysis Reports**: FlightNumber\_AnalysisType\_Timestamp.xlsx

1. When Things Don't Work - Your Troubleshooting Guide
   1. "Help! Something's Not Working" - Quick Diagnosis

First, check these basics:

1. Is your computer connected to the network?
2. Did you fill in all required fields (marked with red \*)?
3. Are you using the correct flight number and date?
4. Do you have permission to access the files/folders you selected?
   1. Log Extraction Problems

* **"Configuration Error - Please configure Azure settings"**

**What you see**: Error message when you click "Extract Logs"

**What it means**: The system can't connect to Azure DevOps

**How to fix it**:

1. Click "⚙ Configure Azure" button
2. Fill in all the required fields:
   1. Organization URL (get from your IT team)
   2. Project Name (ask your team lead)
   3. Personal Access Token (create in Azure DevOps)
   4. Environment (usually "Production")
3. Click "Test Connection" - it should say "Success"
4. Click "Save"
5. Try extracting logs again

* **"Validation Error - Please enter a flight number"**

**What you see**: Error when flight number field is empty

**How to fix it**: Type your flight number (like WF123, AI101, etc.)

**Remember**: Must be at least 3 characters

* **"Pipeline execution failed"**

**What you see**: Extraction starts but then fails

**Most common causes**:

* + Flight didn't actually operate on that date
  + You don't have permission to access that flight's data
  + Temporary Azure system issue

**How to fix it**:

1. **Double-check the flight details** with operations team
2. **Verify the flight date** - did it actually fly?
3. **Wait 5 minutes and try again** (might be temporary)
4. **Contact your Azure admin** if it keeps failing

* 🚨 **"No logs found for flight"**

**What you see**: Extraction completes but no files found

**What it means**: Azure didn't find any logs for your flight

**Possible reasons**:

* + Flight number typed wrong (WF123 vs WF 123 vs WF-123)
  + Wrong date selected
  + Flight was cancelled/didn't operate
  + Logs not available yet (very recent flights)

**How to fix it**:

1. **Check flight number spelling** - try different formats
2. **Verify flight date** with operations
3. **Try ±1 day** if you're unsure of exact date
4. **For very recent flights**: Wait a few hours and try again
   1. Log Parser Problems

* **"No messages extracted from logs"**

**What you see**: Parsing completes but results table is empty

**Most common cause**: Wrong data type selected

**How to fix it**:

1. **Try the other data type**: If you used API, try PNR (or vice versa)

2. **Check the log files**: Open one in Notepad - do you see passenger names?

3. **Verify flight details**: Make sure flight number/date match the logs

4. **Turn on Debug mode**: Click the Debug toggle to see detailed messages

* **"Directory not found" or "Access denied"**

**What you see**: Error when you click "Process Logs"

**What it means**: System can't read your log folder

**How to fix it**:

1. **Check the path**: Click "Browse" and reselect your log folder

2. **Verify folder exists**: Browse to it in Windows Explorer

3. **Check permissions**: Can you open files in that folder normally?

4. **Try copying files**: Copy logs to your Documents folder and try again

* **"Multi-node consolidation failed"**

**What you see**: Error about n1/n2/n3 folders

**What it means**: System expected distributed logs but found different structure

**How to fix it**:

1. **Turn OFF Multi-node mode**: Uncheck the Multi-node toggle

2. **Check folder structure**: Do you actually have n1/, n2/, n3/ subfolders?

3. **If yes**: Make sure each subfolder contains log files

* 1. RCA Engine Problems
* **"No passenger data found"**

**What you see**: Processing completes but no statistics shown

**What it means**: System didn't find expected passenger data files

**How to fix it**:

1. **Check your folder**: Look for files named like:

* + Input\*.txt
  + Output\*.txt
  + passenger\_api\_input.txt
  + Any files with passenger data

2. **Try different Data Type**: Switch between API/Passenger/Crew

3. **Check "Files Processed" list**: See what files the system found

4. **Verify file content**: Open a file in Notepad - does it have passenger names?

* **"Excel export failed"**

**What you see**: Error when clicking "Export to Excel"

**Most common causes**: File permissions or Excel conflicts

**How to fix it**:

1. **Close Excel**: If you have Excel open with a file named the same way

2. **Check permissions**: Can you create files in that folder normally?

3. **Try different location**: Browse to your Desktop or Documents

4. **Check disk space**: Make sure you have space for the report file

* **"All passengers showing as duplicates"**

**What you see**: Duplicate count equals total passengers

**What it means**: Data format might be causing false duplicates

**How to investigate**:

1. **Look at Duplicate Passengers table**: What do the entries look like?

2. **Check for extra spaces**: Sometimes "John Smith" vs "John Smith " (extra space)

3. **Date format issues**: Different date formats might cause duplicates

4. **Ask for help**: This might need L3 analyst or developer investigation

* 1. Performance Issues
* **"Everything is running really slowly"**

**What you see**: Operations taking much longer than expected

**Possible causes**: Large files, low memory, or background processes

**How to improve performance**:

1. **Close other applications**: Free up memory for L3 Engine

2. **Check file sizes**: Very large log files take longer to process

3. **Be patient**: Complex investigations can take 15-20 minutes

4. **Restart application**: Close and reopen L3 Engine if it seems stuck

* **"Log extraction taking over 10 minutes"**

**What to do**:

1. **Check Azure DevOps**: Maybe the pipeline is busy

2. **Don't cancel**: Unless you're sure it's stuck

3. **Try during off-peak hours**: Early morning or late evening

4. **Contact Azure admin**: If it consistently takes too long

* 1. Data Quality Red Flags
* **"Passenger names look weird"**

**Examples you might see**:

* + Names like "SMITH?JOHN" or "SM█TH JOHN"
  + Empty names or just numbers
  + Very long names that seem corrupted

**What this means**: Data encoding or corruption issues

**What to do**:

1. **Document the issue**: Take screenshots

2. **Check source data**: Look at the original input files

3. **Report to L3**: This usually needs deeper investigation

4. **Continue analysis**: Focus on passengers with normal names

* **"Impossible passenger counts"**

**Examples**:

* + Negative dropped passengers
  + More output than input (when there shouldn't be)
  + Counts that don't make logical sense

**What this means**: System processing issue or data corruption

**What to do**:

1. **Double-check your inputs**: Verify folder and settings
2. **Try different Record Type**: Use "Input only" or "Output only"
3. **Document the anomaly**: Include in your investigation report
4. **Escalate**: This needs L3 analyst attention
   1. "I'm Completely Stuck" - Escalation Guide

When you've tried everything and still can't get results:

**What Information to Gather Before Escalating**

1. **Flight Details**: Exact flight number, date, route

2. **Error Messages**: Screenshots of any error messages

3. **What You Tried**: List of troubleshooting steps attempted

4. **Files Involved**: Attach all data files used

5. **Expected vs Actual**: What you expected to happen vs what actually happened

**Who to Contact**

1. **Technical Issues**: L3 analyst or team lead

2. **Azure/Network Issues**: L3 analyst or team lead

3. **Data Analysis Issues**: L3 analyst or team lead

4. **Application Bugs**: L3 analyst or team lead

**How to Document for Escalation**

**Subject**: "L3 Engine Issue - [Flight Number] - [Brief Description]"

**Include**: All information from above

**Attach**: Screenshots, error logs, sample data files (if permitted)

**Urgency**: Clearly state if this is blocking a critical investigation

* 1. Prevention Tips
* **Best Practices to Avoid Problems**

1. **Double-check flight details**: Most issues come from typos

2. **Start with recent, successful flights**: For testing and learning

3. **Keep files organized**: Use consistent folder naming

4. **Regular Azure token updates**: Renew tokens before they expire

5. **Document your process**: Note what works for your investigations

* **When to Ask for Help Early**
  + Don't spend more than 30 minutes troubleshooting alone
  + If error messages don't make sense
  + If you're investigating a critical/urgent incident
  + If you suspect data corruption or system issues

1. How to Be Successful with L3 Engine
   1. Your Investigation Playbook

**The 15-Minute Quick Investigation (Try fast, Fail fast)**

For urgent issues where you need answers fast:

1. **Minute 1-2**: Launch Log Extraction Engine, gather flight details

2. **Minute 3-8**: Extract logs (flight number, date, click Extract)

3. **Minute 9-12**: Launch Log Parser, parse data (API type, same flight details, Process)

4. **Minute 13-15**: Launch RCA Engine, quick analysis (Process, check summary)

**Result**: You'll know if passengers are missing and have basic counts for immediate response.

**The Complete Investigation (10-20 minutes)**

For thorough analysis and documentation:

1. **Planning (5 minutes)**:

1. Write down flight details, incident description
2. Create investigation folder on your computer
3. Gather any additional context .

2. **Log Extraction (5 minutes)**:

1. Extract comprehensive logs
2. Download all files to organized folder
3. Document extraction timestamp and file count

3. **Data Parsing (5 minutes)**:

1. Parse API data for passenger manifest issues
2. Parse PNR data if investigating booking/ticketing issues
3. Save extracted messages to organized folders
4. Review message preview for data quality

4. **Root Cause Analysis (5 minutes)**:

1. Run RCA analysis on data to understand what came into the system
2. Compare results to identify missing or dropped passengers
3. Review all data tables for patterns and issues
4. Generate Excel report with findings

5. **Documentation (5 minutes)**:

1. Add your analysis notes to Excel report
2. Save all files with consistent naming
3. Prepare summary for stakeholders
   1. File Organization That Works

**Folder Structure (Create This Once, Use Forever)**

L3\_Investigations/

├── 2024-11-20\_WF123\_MissingPax/

│ ├── 01\_ExtractedLogs/

│ ├── 02\_ParsedData/

│ ├── 03\_Analysis/

│ └── 04\_Reports/

├── 2024-11-21\_AI101\_Duplicates/

│ ├── 01\_ExtractedLogs/

│ └── [same structure]

└── Templates/

└── Investigation\_Template.xlsx

**File Naming Convention**

* **Investigations**: YYYY-MM-DD\_FlightNumber\_IssueType
* **Reports**: FlightNumber\_YYYYMMDD\_RCA\_Report.xlsx
* **Notes**: FlightNumber\_Investigation\_Notes.txt
  1. Working with Your Team

**L2 to L3 Escalation**

When escalating to L3 analyst:

* **Complete all L2 investigation steps** before escalating
* **Prepare executive summary**: One paragraph with key findings
* **Highlight specific concerns**: Unusual patterns, data corruption, system issues
* **Provide complete data package**: All extracted files and analysis
* **Document timeline**: When incident occurred vs when investigation started

When Reporting a Bug:

* **Prepare a folder** with all the extracted logs of that case.
* **Name the folder** - "*FlightNumber\_DepPort\_ArrPort\_DepDate*"
* **Share** the above details with L3 Devs.
  1. Advanced Investigation Techniques

**Pattern Recognition**

Look for these common patterns in your investigations:

**Missing Passenger Patterns**:

Same nationality (system processing rules)

Same document type (validation issues)

Same name format (encoding problems)

Consecutive seat numbers (batch processing failures)

**Data Quality Patterns**:

Names with special characters causing issues

Date format inconsistencies between systems

Document number validation failures

System timeout patterns during busy periods

**Cross-Investigation Analysis**

Keep a simple log of your investigations:

Flight routes with frequent issues

Time periods with system problems

Recurring patterns across multiple flights

System updates that correlate with issues

9.6 Time Management

**Priority Matrix for Incident Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **Urgency** | **Complexity** | **Investigation Time** | **Approach** |
| High | Low | 15 minutes | Quick investigation, immediate response |
| High | High | 45 minutes + L3 escalation | Thorough analysis, L3 consultation |
| Medium | Low | 30 minutes | Complete investigation, documentation |
| Medium | High | 60 minutes + team review | Comprehensive analysis, peer review |

**Batching Similar Investigations**

When you have multiple similar incidents:

1. **Extract logs for all flights first** (batch extractions operations)
2. **Parse data for all flights** (similar patterns, faster processing)
3. **Compare results across flights** (identify system-wide issues)
4. **Create combined report** (show trends and patterns)

9.7 Continuous Improvement

**After Each Investigation, Ask:**

What was the root cause, and could L3 Engine have found it faster?

What additional data would have been helpful?

What part of the process took longest, and why?

What would you do differently next time?

**Monthly Review Questions:**

* What types of issues are you investigating most often?
* Are there patterns across multiple investigations?
* What additional L3 Engine features would help your workflow?
* How can you share learnings with your team?

**Remember**: L3 Engine is a tool to help you investigate faster and more thoroughly. Your analytical skills, domain knowledge, and ability to ask the right questions are still the most important parts of being a successful analyst.

1. Appendices

**Appendix A: File Format Specifications**

A.1 EDIFACT Message Structure

UNH+MSG001+PAXLST:D:02B:UN:IATA'

BGM+745+MSG001+9'

DTM+132:20241120:102'

NAD+MS+++AIRLINE NAME'

TDT+20+WF123+1++WF:AIRLINE CODE'

LOC+125+DEL'

DTM+189:20241120:1430:203'

LOC+87+BOM'

DTM+232:20241120:1730:203'

NAD+FL+++PASSENGER NAME'

ATT+2++M'

DTM+329:19851015'

LOC+22+IND'

NAT+2+IND'

DOC+P:::::PASSPORT NUMBER'

UNT+15+MSG001'

A.2 Log File Patterns

[TIMESTAMP] [LEVEL] [COMPONENT] Message content

[2024-11-20 14:30:15.123] [INFO] [PaxProcessor] Processing passenger

manifest

[2024-11-20 14:30:15.456] [ERROR] [APIGateway] Connection timeout

**Appendix B: Error Code Reference**

B.1 Log Extraction Errors

**LE001**: Azure configuration missing

**LE002**: Invalid credentials

**LE003**: Pipeline trigger failed

**LE004**: Network connectivity error

**LE005**: Insufficient permissions

**LE006**: Required variables missing

**LE007**: Invalid directory path

B.2 Log Parser Errors

**LP001**: Invalid directory path

**LP002**: No EDIFACT messages found

**LP003**: Message parsing failed

**LP004**: Multi-node consolidation error

**LP005**: File format not supported

**LP006**: Required variables missing

B.3 RCA Engine Errors

**RCA001**: No passenger files found

**RCA002**: File parsing error

**RCA003**: Data validation failed

**RCA004**: Excel export failed

**RCA005**: Memory allocation error

**Appendix C: Configuration Templates**

C.1 Azure DevOps Configuration Template

{

"organizationUrl": "https://dev.azure.com/YourOrganization",

"projectName": "YourProject",

"personalAccessToken": "your-pat-token",

"environment": "env\_name",

"pipelineId": "your-pipeline-id"

}

C.2 Default Directory Structure

L3\_Engine\_Work/

├── Configurations/

│ └── azure\_config.json

├── Extractions/

│ └── [Flight\_Date\_folders]/

├── Parsing/

│ └── [Flight\_Date\_folders]/

└── Analysis/

└── [Flight\_Date\_folders]/

Document Information

**Document Title**: L3 Engine User Manual.

**Document Version**: 1.0.0

**Last Updated**: November 26, 2025

**Target Users**: L2 Analysts, Operations Teams.

**Classification**: Internal Use

**Authored by**:

Shivansh Singh Bhadouria

Software Developer, API PNR Gateway

*If you find errors, have suggestions, or need additional guidance not covered in this manual, please contact the author or your team lead.*

**Copyright**: © 2025 SITA. All rights reserved.