

RPA Oracle Invoice Processing - Technical Documentation

1. Overview

This document explains the Python-based RPA script used to automate **Transfer Out / Invoice processing** using:
- Screen automation (keyboard & image-based actions)
- OCR validation using Tesseract
- Oracle Database integration for data fetch and logging

The script continuously fetches pending MR (Material Receipt) numbers from Oracle DB, processes them in an ERP screen, validates values using OCR, and logs successful transactions back into the database.

2. Technology Stack

Component	Purpose
rpa (TagUI for Python)	UI automation (keyboard, image detection)
pyautogui	Screen capture & mouse actions
pytesseract	OCR text recognition
OpenCV (cv2)	Image preprocessing
oracledb	Oracle DB connectivity
logging	Runtime logging
traceback	Error diagnostics

3. Configuration Section

3.1 OCR & File Paths

```
TESSERACT_PATH = "C:\\\\Program Files\\\\Tesseract-OCR\\\\tesseract.exe"
TEMP_IMAGE_PATH = "D:\\\\python\\\\temp_area.png"
OCR_REGION = (1711,354,205,60)
```

- **OCR_REGION**: Screen coordinates where stock quantity is validated - **TEMP_IMAGE_PATH**: Temporary image used for OCR processing

3.2 Wait Timings

```
WAIT_SHORT = 1  
WAIT_LONG = 3
```

Controls keyboard and screen wait timings for stability.

3.3 Logging

```
LOG_FILE = "rpa_log.txt"
```

Logs all processing steps, warnings, and errors.

4. Oracle Database Configuration

```
DB_USER = "RAYMEDI_RAMRAJ"  
DB_PASS = "raymedi_hq"  
DB_DSN = "ramraj-qa.cugyvyz68ru0.ap-south-1.rds.amazonaws.com:1521/ramrajqa"
```

Used for: - Fetching pending MR numbers - Logging completed RPA transactions

5. Logging Setup

```
logging.basicConfig(  
    level=logging.INFO,  
    format='%(asctime)s [%(levelname)s] %(message)s',  
    handlers=[logging.FileHandler(LOG_FILE), logging.StreamHandler(sys.stdout)]  
)
```

Ensures logs are written to both console and file.

6. Utility Functions

6.1 OCR Read Function

```
def read_text_from_screen(region):
```

Purpose: - Takes screenshot of given screen region - Extracts text using Tesseract OCR

6.2 Error Handler

```
def handle_failure(error_message):
```

Purpose: - Logs detailed traceback - Safely closes RPA - Exits script to avoid partial processing

6.3 Stock Not Available Handler

```
def stock_not():
```

Purpose: Handles ERP navigation when "Stock Not Available" popup appears using keyboard automation.

7. Database Operations

7.1 Fetch Pending Invoices

```
def fetch_outlet_invoice():
```

Functionality: - Fetches Outlet Name & MR Number - Filters already processed records - Returns list of `(outlet, invoice)` tuples

8. Main Processing Logic

8.1 `process_data_from_db()`

Workflow:

1. Connect to Oracle DB
2. Fetch pending MR records
3. Loop through each record
4. Navigate ERP UI using keyboard
5. Enter outlet and invoice number

6. Validate invoice using OCR
 7. Perform stock validation using OCR region
 8. Save transaction
 9. Insert log into Oracle table `TRANSFER_OUT_RPA_LOG`
-

8.2 Invoice OCR Validation

```
custom_config = r'--oem 3 --psm 7 -c tessedit_char_whitelist=MR0123456789'
```

- Restricts OCR to valid MR characters - Prevents mismatch errors

If OCR result != DB invoice → record is skipped safely.

8.3 Stock Validation Loop

```
for attempt in range(22):
```

- Repeats OCR validation until non-zero stock is detected - Handles `stock_not.png` condition - Fails after max retries
-

8.4 Database Insert on Success

```
INSERT INTO TRANSFER_OUT_RPA_LOG  
(OUTLET_NAME, MR_NO, CREATED_DATE, CREATED_TIME, STATUS)
```

Records each successful automation entry.

9. Continuous Execution (Main Loop)

```
while True:
```

- Script runs every 10 minutes (`sleep(600)`) - Ideal for background RPA service - Safely restarts after each cycle
-

10. Error Handling Strategy

Scenario	Action
OCR mismatch	Skip record
Stock not found	Handle popup & retry
Any exception	Log error & exit

11. Output Artifacts

- `rpa_log.txt` → Execution logs
 - `invoice.png` → OCR validation snapshot
 - `temp_area.png` → Stock OCR image
-

12. Best Practices & Notes

- Screen resolution must remain constant
 - Image assets (`ok.png`, `no.png`, `stock_not.png`) must be accurate
 - OCR regions may need tuning per system
 - Database credentials should ideally be moved to environment variables
-

13. Conclusion

This RPA script ensures reliable, automated processing of MR invoices with strong validation using OCR and database tracking. It is suitable for unattended execution in production environments with minimal manual intervention.

Document Prepared For: RPA / ERP Automation Support Team