

Standard Operating Procedure: Production Reports

This document outlines the standard procedure for executing reports from the `bcsb-prod` repository.

1. Prerequisites

Before running the reports, ensure your local environment is correctly configured. This is a one-time setup and a daily check to ensure your tools are current.

- **Repository Access:** You must have `bcsb-prod` repository, which is located & maintained on Github Enterprise.
 - **Software:**
 - **Git:** Must be installed on your system.
 - **Python:** A compatible version of Python must be installed.
 - 3.11.x
 - **Terminal:** You will need a command-line interface (e.g., PowerShell, Git Bash, or Terminal). The instructions below are optimized for PowerShell but include notes for other shells.
-

2. Daily Reporting Procedure

Perform these steps each morning to generate the required reports.

Step 1: Navigate to the Project Directory

Open your terminal and navigate to the root of your local `bcsb-prod` repository.

```
cd path/to/your/bcsb-prod
```

Step 2: Update Your Local Repository

Ensure you have the latest version of the reporting script and configurations by pulling the latest changes from the `main` branch. This prevents errors caused by running outdated code.

```
git pull origin main
```

Step 3: Set the Environment Variable for Production

You must explicitly set the environment to "prod". This critical step directs the scripts to run against live production data. **Failure to do this correctly could result in running reports against the wrong environment.**

The command varies depending on your terminal:

- **Windows PowerShell (Recommended):**

```
$env:REPORT_ENV="prod"
```

- **Windows Command Prompt (CMD):**

```
set REPORT_ENV=prod
```

- **Linux / macOS / Git Bash:**

```
export REPORT_ENV="prod"
```

Note: This variable is only set for your current terminal session. You must set it every time you open a new terminal to run reports.

Step 4: Execute the Reports

Paste the following command into your terminal. It will run the four necessary morning reports sequentially. The script is configured to handle one report at a time.

Below is an example of how you'd run the daily report after setting the environment to prod:

```
python testing/run_reports.py --name "R360"; python testing/run_reports.py --name "Daily Deposit Update"; python testing/run_reports.py --name "Dealer Reserve Recon"; python testing/run_reports.py --name "Daily Mismatched Debit Card Txns"
```

The terminal will display the status of each report as it completes, indicating success or failure.

3. Verification and Troubleshooting

Confirming Success

- **Terminal Output:** Observe the output in your terminal for a "success" message after each report completes.
- **Log File:** All execution details, including any errors, are automatically logged. You can review the log file for a complete record of the process:
 - **Log File Location:** bcsb-prod/prod_execution.log

Full Reporting Schedule

Daily - Morning

Report Name	Script Command
R360	<code>python testing/run_reports.py --name "R360"</code>
Daily Deposit Update	<code>python testing/run_reports.py --name "Daily Deposit Update"</code>
Dealer Reserve Recon	<code>python testing/run_reports.py --name "Dealer Reserve Recon"</code>
Daily Mismatched Debit Card Txns	<code>python testing/run_reports.py --name "Daily Mismatched Debit Card Txns"</code>

Daily - 11:00 AM

Report Name	Script Command
Rate Scrape	<code>python testing/run_reports.py --name "Rate Scrape"</code>

Weekly

In progress

Monthly (1st of the Month for ME reporting)

In progress