# **Task Report Generation**

## **System Overview**

**The Task Usage Reporting System is an automated data pipeline that collects, processes, and publishes task usage metrics from multiple countries/regions to Confluence. The system supports both database connections and CSV file imports with automatic aggregation and visualization.**

### **Key Features**

* **Multi-Country Support** - Process data from multiple countries in a single run
* **Dual Data Sources** - Support for both Oracle database connections and CSV file imports
* **Auto-Discovery** - Automatic detection of CSV files following naming conventions
* **Flexible Scheduling** - Monthly and Daily (YTD) reporting modes
* **Automated Publishing** - Direct publishing to Confluence with charts and tables
* **Secure Credentials** - Encrypted password storage for database and Confluence access

## **1. System Architecture**

### **Core Components**

| **Component** | **File** | **Purpose** |
| --- | --- | --- |
| Main Orchestrator | main.py | Entry point, coordinates workflow execution, handles date ranges |
| Workflow Engine | src/workflow.py | Manages database connections, query execution, data extraction |
| CSV Processor | lib/csv\_processor.py | Aggregates multiple country CSVs into unified reports |
| Confluence Publisher | lib/confluence\_publisher.py | Creates and publishes HTML reports with charts |
| Security Manager | lib/secure\_config.py | Handles encryption/decryption of credentials |

### **Configuration Files**

| **File** | **Purpose** |
| --- | --- |
| config/config.json | Confluence settings, baseline values, CSV paths |
| config/config.ini | Database connection parameters (single country) |
| config/countries.json | Multi-country configuration with DB details |
| config/query.sql | Base SQL query template |
| config/HK\_config.ini, UK\_config.ini | Country-specific DB configurations |

## **2. Operating Modes**

### **Monthly Mode (--monthly)**

* **Purpose:** Generate report for the previous completed month
* **Date Range:** First day to last day of previous month
* **Page Title:** "Report Title - [Month Name]"
* **Use Case:** Regular monthly reporting after month close
* **Example:** Running in September generates August report

### **Daily/YTD Mode (--daily)**

* **Purpose:** Generate Year-to-Date report through current date
* **Date Range:** January 1st to current date
* **Page Title:** "Report Title - [Current Month] YTD"
* **CSV Handling:** Aggregates all monthly CSV files from January to current month
* **Use Case:** Daily monitoring, mid-month reviews

### **Test Mode (--test)**

* **Purpose:** Validate workflow with synthetic data
* **Behavior:** Generates dummy data instead of querying databases
* **Page Title:** Appends "-TEST" to prevent overwriting production

## **3. Data Flow**

main.py (Orchestrator)

├── Database Countries (workflow.py)

│ └── Oracle DB → Execute SQL → Extract Data

├── CSV Countries (Auto-Discovery)

│ └── CSV Files → Pattern: CC\_USAGEREPORT\_YYYYMM.csv

↓

CSV Processor (Aggregation)

• Aggregate by date

• Add country column

• Create reports

↓

Confluence Publisher

• Generate charts

• Create HTML tables

• Publish to Space

## **4. Execution Commands**

### **Single Country Mode**

bash

*# Previous month report*

python main.py --monthly

*# Year-to-date report*

python main.py --daily

*# Test mode*

python main.py --test

### **Multi-Country Mode**

bash

*# Monthly report for all countries*

python main.py --monthly --countries-config config/countries.json

*# YTD report for all countries*

python main.py --daily --countries-config config/countries.json

*# Skip Confluence publishing*

python main.py --monthly --countries-config config/countries.json --no-publish

### **CSV Operations**

bash

*# List available CSV countries*

python main.py --list-csv-countries

*# Process only CSV countries*

python main.py --csv-countries-only --monthly

*# Specify target month*

python main.py --monthly --target-month 202508

## **5. Configuration Examples**

### **Database Countries (countries.json)**

json

{

"countries": [

{

"name": "HK",

"config\_file": "config/HK\_config.ini",

"query\_file": "config/query.sql"

},

{

"name": "UK",

"config\_file": "config/UK\_config.ini",

"query\_file": "config/query.sql"

}

]

}

### **CSV Configuration (config.json)**

json

{

"CSV\_COUNTRIES\_PATH": "C:/path/to/csv/files",

"BASELINE": 1899206,

"CONFLUENCE\_URL": "https://confluence.company.com",

"SPACE\_KEY": "TECH",

"PAGE\_TITLE": "Task Usage Report"

}

## **6. Output Structure**

### **Directory Organization**

reports/

├── 2025-09-17/ # First run of the day

│ ├── data\_hk.csv # Individual country data

│ ├── data\_uk.csv

│ ├── data\_mx.csv

│ └── task\_usage\_report\_by\_region.csv # Aggregated

├── 2025-09-17\_2/ # Second run of the day

└── 2025-09-17\_3/ # Third run of the day

### **Generated Reports**

| **File** | **Description** |
| --- | --- |
| data\_{country}.csv | Individual country data with NET\_DATE and TOTAL\_JOBS |
| task\_usage\_report\_by\_region.csv | Aggregated data with DATE, COUNTRY, TOTAL\_JOBS |
| task\_usage\_report.csv | Detailed report with all metrics |

## **7. Security Setup**

### **Database Credentials**

bash

*# Setup encrypted config*

python setup\_secure\_config.py

*# Or use environment variable*

export DB\_PASSWORD="your\_password"

### **Confluence Authentication**

bash

*# Environment variable (recommended)*

export CONFLUENCE\_PASSWORD="your\_api\_token"

*# Or encrypted in config.json*

{

"PASSWORD\_ENCRYPTED": "encrypted\_string"

}

## **8. Troubleshooting**

### **Common Issues**

| **Issue** | **Cause** | **Solution** |
| --- | --- | --- |
| No CSV countries found | Wrong path or file naming | Check CSV\_COUNTRIES\_PATH and file pattern |
| Query date range not found | SQL format mismatch | Verify BETWEEN TO\_DATE pattern |
| datetime has no attribute timedelta | Import missing | Add from datetime import timedelta |
| Folder already exists | Multiple runs same day | System auto-creates numbered folders |
| Authentication failed | Invalid credentials | Run setup\_secure\_config.py |
| No data for month | Date filtering issue | Check target\_month format (YYYYMM) |

### **Log Files**

* **Location:** workflow\_YYYYMMDD\_HHMMSS.log
* **Content:** Detailed execution trace
* **Check for:** ERROR and WARNING messages

### **Validation Checklist**

□ Database connectivity (tnsping) □ CSV files follow naming convention  
 □ Confluence space and page permissions □ Date ranges in SQL query □ config.json has all required keys □ Environment variables set (if used)

## **9. CSV File Requirements**

### **Naming Convention**

* **Pattern:** {COUNTRY\_CODE}*USAGEREPORT*{YYYYMM}.csv
* **Example:** MX\_USAGEREPORT\_202509.csv

### **Required Columns**

* Date column: NET\_DATE, DATE, DAY, or REPORT\_DATE
* Jobs column: JOBS, JOBS\_PER\_DAY, TOTAL\_JOBS, JOB\_COUNT, or FVALUE

### **Supported Date Formats**

* 01/08/2025 08:00 (DD/MM/YYYY HH:MM)
* 2025-08-01 08:00:00 (YYYY-MM-DD HH:MM:SS)
* 08/01/2025 08:00 (MM/DD/YYYY HH:MM)

## **10. Report Features**

### **Jobs Data Table**

* Top 4 peak days
* Baseline comparison
* Variation analysis (green/red indicators)

### **Charts Generated**

* Peak comparison bar chart (3D)
* Monthly usage line chart
* Country breakdown (YTD mode)
* Country summary with percentages

### **Page Updates**

* Automatic title updates based on mode
* Timestamp and user tracking
* Data coverage period display

## **11. Maintenance**

### **Regular Tasks**

| **Frequency** | **Task** |
| --- | --- |
| Weekly | Review log files for warnings |
| Monthly | Archive old report folders |
| Quarterly | Update baseline if needed |
| Yearly | Clean up backup query files (.bak) |

### **Performance Tips**

* Use connection pooling for multiple DB queries
* Implement parallel processing for countries
* Archive historical reports
* Index Confluence pages for faster updates

## **12. Quick Reference**

### **Most Common Commands**

bash

*# Generate previous month report for all countries*

python main.py --monthly --countries-config config/countries.json

*# Generate YTD report for all countries*

python main.py --daily --countries-config config/countries.json

*# Test the system*

python main.py --test --countries-config config/countries.json

*# Check available CSV files*

python main.py --list-csv-countries

**Document Version:** 1.0  
 **Last Updated:** September 2025  
 **System Version:** Task Usage Reporting v2.0  
 **Contact:** Technical Team

*Note: This document provides comprehensive technical documentation for the Task Usage Reporting System. For additional support, review log files and test with --test mode before production runs.*