PRT File Processor: Workover and Connection Analysis

Description

This Python script processes oil/gas well simulation output files (.PRT extension) to extract, analyze, and report on **well connection events** (opening and closing) which is considered as **workover operations**. It generates structured Excel summaries for operational planning, including drilling schedules, workover counts, and facilities payments.

Key Features

1. **Extracts and categorizes**:
    - **Closing connections** (with variable limit violations).
    - **Opening connections** (newly established well connections).
2. **Computes annual metrics**:
    - Connections (opened/closed) per well per year.
    - Workovers triggered by excessive connection changes.
3. **Applies business rules**:
    - Workover thresholds (pre-2027: >3 changes, post-2027: >2 changes).
    - Annual workover cap (max 6, with spillover to next year).
4. **Generates Excel reports** with multiple sheets for auditability:
    - Raw extracted data, aggregated counts, and final structured outputs.

Functions Overview

extract_prt_data(file_path)

- **Input**: Path to a .PRT file.
- **Output**: Two DataFrames:
    - df_closing: Contains closing events (date, well, variable limits).
    - df_opening: Contains opening events (date, well, connection ID).
- **Logic**: Parses lines with @ Closing connection or @ Opening connection.

compute_connections_per_well(df_closing, df_opening)

- **Input**: DataFrames of closing/opening events.
- **Output**: Merged DataFrame with yearly counts per well.
- **Columns**: Year, Well, Closed_Connections, Opened_Connections.

compute_workovers_per_year(df_connections)

- **Input**: Aggregated connection counts.
- **Output**: Workovers per year, filtered by threshold rules.
- **Thresholds**:
    - **<2027**: >3 total connections (opened + closed).
    - **≥2027**: >2 total connections.

enforce_max_workover(df_workovers, max_workovers=6)

- **Input**: Raw workover counts.
- **Output**: Adjusted counts (capped at 6/year, excess spills over).

generate_final_dataframe(df_adjusted_workovers, filename)

- **Input**: Adjusted workovers and filename (for scenario detection).
- **Output**: Structured DataFrame (2024–2050) with:
    - Drilling schedules (vertical/horizontal wells).
    - Workover counts (perf/shut-off, pump replacement).
    - Facilities payment schedules (if BDPRODUCERS in filename).

process_all_prt_files(root_dir)

- **Input**: Directory containing .PRT files.
- **Output**: Excel files (*_summary.xlsx) with sheets:
    1. **Final Structured Data**: High-level summary.
    2. **Raw Closing/Opening Connections**: Extracted events.
    3. **Connections per Well per Year**: Aggregated counts.
    4. **Raw/Final Workovers per Year**: Pre- and post-adjustment.

Usage Example

python

```
if __name__ == "__main__":
    folder_path = os.path.join(os.path.dirname(__file__), "prt_files")
    process_all_prt_files(folder_path)  # Processes all .PRT files in `prt_files/`
```

Output Example

Year	Drilling of Vertical Wells	Workover (Perf or Shut-off)	Facilities Payment Schedule (%)
2025	4	3	30
2026	4	6	40

Dependencies

- Python 3.7+
- Libraries: pandas, re, os.

Assumptions

- .PRT files follow the specified format (e.g., @ Closing connection (X,Y,Z)).
- Workovers are triggered by connection changes (not other events).

This documentation ensures clarity for future maintenance and collaboration. Adjust paths/filenames as needed for your project.