EventFilter

+ event_type : str
+ count : bool
+ level : Optional[str]
+ pattern_str : Optional[str]
+ pattern : Optional[Pattern]

+ __init__(event_type: str, count: bool = False, level: Optional[str] = None,
pattern: Optional[str] = None)

+ matches(log: Dict[str, Any]) : bool

+ description() : str

parse_events_file(events_file: str) -> List[EventFilter]

Parse the events configuration file and return a list of EventFilter objects. Ignores comments and blank lines. Handles parsing errors gracefully.

parse_log_line(line: str) -> Optional[Dict[str, Any]]:

Parse a single log line into its components. Returns a dict with timestamp, level, event_type, message, and raw line. Returns None if the line does not match the expected format.

log_files_in_dir(log_dir: str) -> Iterator[str]:

Yield all .log and .log.gz files in the given directory, but only if they are files (not directories).

log_lines_from_file(filepath: str) -> Iterator[str]:

Yield lines from a log file, supporting both plain text
and gzip-compressed files.

filter_logs(log_dir: str, from_ts: Optional[str], to_ts: Optional[str]) -> Iterator[Dict[str, Any]]:

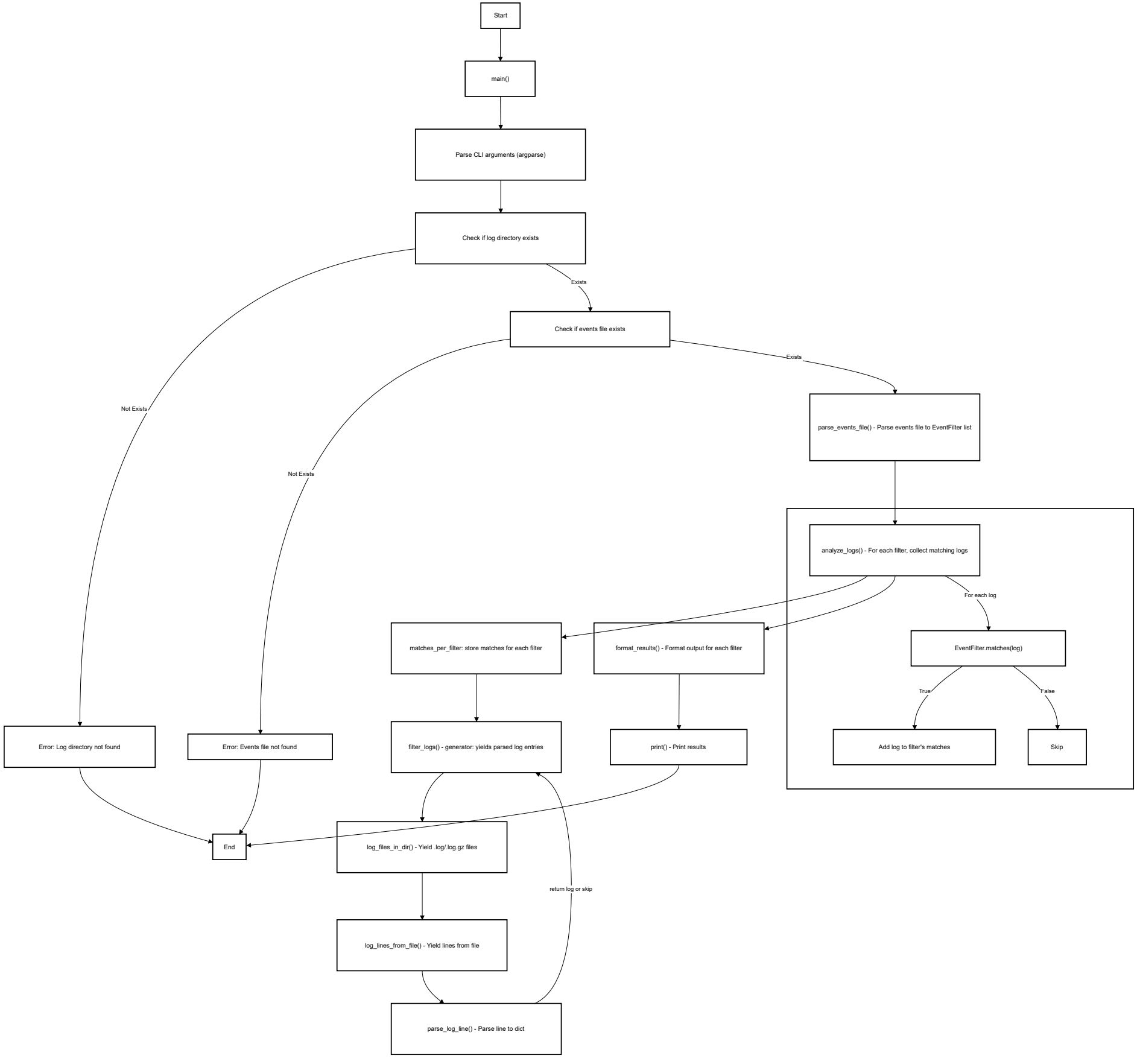
Yield lines from a log file, supporting both plain text
and gzip-compressed files.

analyze_logs(log_dir: str, event_filters: List[EventFilter], from_ts: Optional[str], to_ts: Optional[str]) -> List[Tuple[EventFilter, List[Dict[str, Any]]]]:

For each event filter, collect all matching log entries from the log files. Returns a list of (EventFilter, [matching logs]) tuples. This version processes logs in a streaming fashion for memory efficiency.

format_results(results: List[Tuple[EventFilter, List[Dict[str, Any]]]]) -> str:

Yield lines from a log file, supporting both plain text
and gzip-compressed files.


Start
main()
Parse CLI arguments (argparse)
Check if log directory exists
Exists
Not Exists
Check if events file exists
Exists
Not Exists
parse_events_file() - Parse events file to EventFilter list
analyze_logs() - For each filter, collect matching logs
For each log
EventFilter.matches(log)
True
False
Add log to filter's matches
Skip
matches_per_filter: store matches for each filter
format_results() - Format output for each filter
Error: Log directory not found
Error: Events file not found
filter_logs() - generator: yields parsed log entries
print() - Print results
End
log_files_in_dir() - Yield .log/.log.gz files
return log or skip
log_lines_from_file() - Yield lines from file
parse_log_line() - Parse line to dict