Version: 1.0.0 **Target:** 0.134.0-Beta

Implementation Plan - Standardize Text Report Headers (Phase 1)

File Identification:

1. `contest_tools/reports/text_rate_sheet_comparison.py` (Baseline: 0.116.0-Beta)
2. `contest_tools/reports/text_missed_multipliers.py` (Baseline: 0.113.0-Beta)
3. `contest_tools/reports/text_qso_comparison.py` (Baseline: 0.90.1-Beta)

Surgical Changes:

1. **`contest_tools/reports/text_rate_sheet_comparison.py`**:
   - **Imports**: Add `format_text_header`, `get_cty_metadata`, `get_standard_title_lines` from `._report_utils`.
   - **Refactor `generate`**:
     - Calculate `modes_present` for Smart Scoping.
     - Generate Block 1 ("Overall Summary") first using a simplified section title (e.g., "Overall Summary") instead of the full contest string.
     - Measure the width of Block 1 (using the separator line).
     - Generate the Standard 3-Line Header using `format_text_header` and the measured width.
     - Prepend the Header to the report blocks.
   - **Refactor `_build_comparison_block`**: No internal changes needed, as we will control the `title` passed to it.
2. **`contest_tools/reports/text_missed_multipliers.py`**:
   - **Imports**: Add `format_text_header`, `get_cty_metadata`, `get_standard_title_lines`.
   - **Refactor `generate`**:
     - Locate the `dummy_table` width calculation (approx line 1238).
     - Remove the manual `title1`, `title2`, `report_lines` centering logic.
     - Construct `title_lines` using `get_standard_title_lines`.
     - Construct `meta_lines` using `get_cty_metadata`.
     - Generate header using `format_text_header` and `max_line_width`.
     - Replace the manual header lines with this new block.
3. **`contest_tools/reports/text_qso_comparison.py`**:

- **Imports**: Add `format_text_header`, `get_cty_metadata`, `get_standard_title_lines`.
- **Refactor `generate`**:
  - Locate the manual header line `report_lines.append(f"QSO Comparison: ...\n" + "="*86)`.
  - Calculate `modes_present`.
  - Define table width (approx 86 chars based on the separator line used in the code).
  - Generate standard header.
  - Replace manual header with standard header.

Surgical Change Verification (`diff`):

CODE_BLOCK diff --- contest_tools/reports/text_rate_sheet_comparison.py +++ contest_tools/reports/text_rate_sheet_comparison.py @@ -29,3 +29,3 @@ from ..data_aggregators.time_series import TimeSeriesAggregator -from ._report_utils import _sanitize_filename_part +from ._report_utils import _sanitize_filename_part, format_text_header, get_cty_metadata, get_standard_title_lines

@@ -82,6 +82,9 @@ contest_name = first_log.get_metadata().get('ContestName', 'UnknownContest') year = first_log.get_processed_data()['Date'].dropna().iloc[0].split('-')[0] if not first_log.get_processed_data().empty else "----"

- `title_main = f"{year} {contest_name} - Comparison (All Bands)"`
-
- `# Calculate modes present for smart scoping`
- `modes_present = set(available_modes)`

`block1 = self._build_comparison_block(`

- `title=title_main,`
- `title="Overall Summary",`
  `col_defs=col_defs,`

@@ -93,2 +96,16 @@ report_blocks.append(block1) +

- `# --- Generate Standard Header ---`
- `# Measure width from the first block (separator line is index 3: Title, Blank, Heade`
- `block1_lines = block1.split('\n')`
- `table_width = len(block1_lines[3]) if len(block1_lines) > 3 else 80`
-
- `title_lines = get_standard_title_lines(self.report_name, self.logs, "All Bands", Non`
- `meta_lines = ["Contest Log Analytics by KD4D", get_cty_metadata(self.logs)]`

-
- ```
      header_block = format_text_header(table_width, title_lines, meta_lines)
  ```
-
- ```
      # Prepend header to blocks
  ```
- ```
      full_content = "\n".join(header_block) + "\n\n" + "\n\n".join(report_blocks) + "\n"
  ```

```
      # --- BLOCKS 2+: Band Details ---
```

@@ -140,5 +157,4 @@

- ```
      # --- Output ---
  ```
- ```
      full_content = "\n\n".join(report_blocks) + "\n"
  ```
- ```
      # (Output generation moved above to inject header, variable reused)
  ```

```
      os.makedirs(output_path, exist_ok=True)
```

--- contest_tools/reports/text_missed_multipliers.py +++ contest_tools/reports/text_missed_multipliers.py @@ -28,3 +28,3 @@ from .report_interface import ContestReport -from ._report_utils import _sanitize_filename_part +from ._report_utils import _sanitize_filename_part, format_text_header, get_cty_metadata, get_standard_title_lines

@@ -227,9 +227,10 @@

- ```
      mode_title_str = f" ({mode_filter})" if mode_filter else ""
  ```
-
- ```
      title1 = f"--- {self.report_name}: {mult_name}{mode_title_str} ---"
  ```
- ```
      title2 = f"{year} {contest_name} - {', '.join(all_calls)}"
  ```
- ```
      report_lines = [title1.center(max_line_width), title2.center(max_line_width)]
  ```
- ```
      # --- Standard Header ---
  ```
- ```
      modes_present = {mode_filter} if mode_filter else set() # Approximate, or pass empty
  ```
- ```
      title_lines = get_standard_title_lines(f"{self.report_name}: {mult_name}", self.logs
  ```
- ```
      meta_lines = ["Contest Log Analytics by KD4D", get_cty_metadata(self.logs)]
  ```
-
- ```
      report_lines = format_text_header(max_line_width, title_lines, meta_lines)
  ```

```
      for band in bands_to_process:
```

--- contest_tools/reports/text_qso_comparison.py +++ contest_tools/reports/text_qso_comparison.py @@ -19,3 +19,3 @@ from .report_interface import ContestReport -from ._report_utils import get_valid_dataframe, create_output_directory +from
```

._report_utils import get_valid_dataframe, create_output_directory, format_text_header, get_cty_metadata, get_standard_title_lines import pandas as pd @@ -47,6 +47,14 @@ bands = sorted(all_bands_in_logs, key=lambda b: canonical_band_order.index(b) if b in canonical_band_order else -1)

- `    # --- Header Generation ---`
- `    table_width = 86`
- `    modes_present = set(pd.concat([df1['Mode'], df2['Mode']]).dropna().unique())`
- `    title_lines = get_standard_title_lines(self.report_name, self.logs, "All Bands", Non`
- `    meta_lines = ["Contest Log Analytics by KD4D", get_cty_metadata(self.logs)]`
- `    report_lines = []`
- `    report_lines.append(f"QSO Comparison: {call1} vs {call2}\n" + "="*86)`
- `    report_lines.extend(format_text_header(table_width, title_lines, meta_lines))`
- `    report_lines.append("="*86)`

`    # Initialize accumulators for the grand totals`

CODE_BLOCK

Affected Modules Checklist:

- `text_rate_sheet_comparison.py`
- `text_missed_multipliers.py`
- `text_qso_comparison.py`

Pre-Flight Check:

- **Inputs:** Source code for 3 text reports.
- **Expected Outcome:** Generated text reports will now feature the standard 3-line title on the left and the Branding/CTY metadata on the right, matching `text_score_report.py`.
- **Mental Walkthrough:** The `format_text_header` utility takes care of the alignment padding. I just need to provide the correct strings and the correct width. `text_rate_sheet_comparison` calculates width dynamically from the table. `text_missed_multipliers` does the same. `text_qso_comparison` uses a fixed width (86) derived from its fixed column layout.
- **State Confirmation Procedure:** Yes.
- **Backward Compatibility:** Yes.
- **Refactoring Impact Analysis:** Low. Only string formatting is changed.
- **Surgical Modification Adherence:** Yes.
- **Syntax Validation:** Checked.
- **Dependency Verification:** `_report_utils` dependencies confirmed.
- **Data Lineage Mandate:** N/A.

- **Inheritance Audit:** N/A.
- **Visual Compliance:** PASS (Adheres to CLAReportsStyleGuide v1.3.0).

Post-Generation Verification:

- Next Action Declaration: I will issue the standardized prompt for **Act as Builder**.