ENSDF Nuclear Data Evaluation Instructions

⚠ CRITICAL WORKFLOW REMINDER ⚠

ALWAYS START WITH: `git status`

- Before any "What changed?" workflow
- Before any change detection or documentation
- This ensures ALL modified files are identified and processed
- Missing this step = incomplete change tracking!

CRITICAL FORMATTING RULE: ALL ENSDF values AND uncertainties MUST be LEFT-JUSTIFIED in their fields!

- Energy values, RI values, half-lives, J-π, AND their uncertainties (DE, DRI, DT, etc.)
- Special markers (GT, LT) within uncertainty fields are also left-justified
- NEVER right-justify or center ANY ENSDF field content!

🎯 80-Column Alignment Debugging Protocol

TRIGGER PHRASES: "not aligned", "wrong columns", "header formatting", "80 characters"

IMMEDIATE RESPONSE:

1. Run `python .github/column_calibrate.py "filename" --header`
2. Use visual ruler technique for manual verification
3. Compare with reference ENSDF files
4. Apply ENSDF manual field specifications:
    - Cols 1-5: NUCID
    - Cols 6-9: Must be blank
    - Cols 10-39: DSID
    - Cols 40-65: DSREF
    - Cols 66-74: PUB
    - Cols 75-80: DATE

Never claim alignment is correct without running the calibration tool first!

Command Triggers

"Self-Calibrate Columns"

Execute column validation on current ENSDF file:

- **PowerShell**: `.\column-calibrate.ps1 "currentfile.ens"` (add `-Detailed` for character mapping)
- **Python**: `python .github/column_calibrate.py "currentfile.ens"` (add `--detailed` for character mapping)
- **Quick Header Check**: `python .github/column_calibrate.py "currentfile.ens" --header`

CRITICAL 80-Column Debugging Technique: When dealing with ENSDF alignment issues, ALWAYS use the visual ruler method:

```
python -c "
header='[paste actual header line here]'
print('ENSDF 80-Column Ruler:')
print('Ones:
12345678901234567890123456789012345678901234567890123456789012345678901234567890')
print('Tens:
11111111112222222222333333333344444444445555555555666666666677777777778888888899
9')
print('Header:', header)
print('Length:', len(header))
"
```

Process: Display 80-char ruler → Extract L/G records → Validate against ENSDF Manual → Report issues

"Debug Header Alignment"

IMMEDIATE ACTION: When header alignment issues are suspected:

1. Run `python .github/column_calibrate.py "filename" --header`
2. Compare with working reference files
3. Use the visual ruler technique to spot misalignments
4. Check ENSDF manual field positions (1-5, 6-9, 10-39, 40-65, 66-74, 75-80)

"What changed?"

MANDATORY FIRST STEP: Always run `git status` to identify ALL modified files.

Execute comprehensive change detection and documentation:

1. **FIRST**: Run `git status` to list all modified files
2. **Verify completeness**: Run `git diff --name-only HEAD` for cross-verification
3. **Check untracked files**: Run `git ls-files --others --exclude-standard`
4. **For each modified file**: Run `git diff HEAD~1 "filename"` to see what changed
5. **For moved files**: Use `git show HEAD~1:old/path/file` to examine previous content
6. **Update change.log** with evidence-based entries (never assume changes)
7. **Document with**:
    - Line numbers where changes occurred
    - Before/after content for significant changes
    - Scientific/technical context and rationale
    - File movement/reorganization details

PowerShell Considerations: Use `Select-Object -First N` instead of `head` for output limiting.

Remember: Git status MUST be the first step - missing files means incomplete documentation! Always cross-verify with multiple git commands to ensure complete coverage.

"Fix format!"

Auto-convert text to proper ENSDF notation:

- Greek letters: `35S` → `{+35}S`, `α` → `|a`, `β` → `|b`, etc.
- Math symbols: `×` → `|*`, `≈` → `|?`, `±` → `|+`, etc.
- Superscripts/subscripts: Use `{+n}` and `{-n}` format

"Convert ENSDF to PDF"

Natural language request processing for ENSDF-to-PDF conversion using the enhanced `ens2pdf.py` script:

Example requests:

- "Convert S35_24mg_14n_3pg.ens to PDF"
- "Generate PDF from the adopted file"
- "Make PDF for the current ENSDF file"
- "ens2pdf for the current ens"
- "Convert Si35 files to PDF and open them"

Process: Automatically locates the specified .ens file, runs the Java conversion tool, and opens the resulting PDF

Script Usage:

```
# Convert single file by name
python ens2pdf.py Si35_adopted

# Convert with full file path
python ens2pdf.py "finished/Si35/new/Si35_adopted.ens"

# Convert all files for an element
python ens2pdf.py Si

# Convert files matching pattern
python ens2pdf.py "Si35_*sig"

# Convert and open in VS Code (default)
python ens2pdf.py Si35_adopted --open

# Convert and open in system viewer
python ens2pdf.py Si35_adopted --open --system
```

Features:

- **Smart PDF Opening**: Tries VS Code first, falls back to system viewer gracefully
- **Full Path Support**: Handles both relative names and complete file paths
- **Pattern Matching**: Use wildcards to convert multiple files
- **Cross-Platform**: Works on Windows, macOS, and Linux
- **Error Handling**: Graceful fallback when VS Code CLI tools aren't available
- **User Feedback**: Clear messages about conversion status and where PDF opened

PDF Location: All PDFs are generated in `D:/X/ND/Files/` directory

ENSDF Column Format Standards (CRITICAL - NO MISTAKES ALLOWED)

L-Record Format (Energy Levels):

```
Columns:
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Format:  35XX  L EEEE.E   DE  JP              T        DT        L        S
DS C
Example: 35P   L 1572.0    1  1/2+            2.29 PS  14        2        1.23
45
```

Field	Columns	Required	Description
NUCID	1-5	✓	Nucleus (e.g., "35P ")
CONT	6		Continuation flag
BLANK	7	✓	Must be blank
TYPE	8	✓	"L"
BLANK	9	✓	Must be blank
E	10-19	✓	Level energy (LEFT-JUSTIFIED)
DE	20-21		Energy uncertainty (LEFT-JUSTIFIED)
SPACE	22	✓	Readability space
J	23-39		Spin-parity (LEFT-JUSTIFIED at col 23)
T	40-49		Half-life with units (LEFT-JUSTIFIED)
DT	50-55		Half-life uncertainty (LEFT-JUSTIFIED)
L	56-64		Angular momentum transfer
S	65-74		Spectroscopic strength
DS	75-76		Uncertainty in S (LEFT-JUSTIFIED)
C	77		Comment flag

G-Record Format (Gamma Transitions):

```
Columns:
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Format:  35XX  G EEEE.E   DE  II.I   DI  [M]      MR     DMR   CC     DCC TI
DTI C
Example: 35P   G 1572.0    1  100.0  4   [E2]     1.23   0.45  0.0368 8   1.23
45
```

Field	Columns	Required	Description
NUCID	1-5	✓	Nucleus (e.g., "35P ")
CONT	6		Continuation flag
BLANK	7	✓	Must be blank
TYPE	8	✓	"G"
BLANK	9	✓	Must be blank
E	10-19	✓	Gamma energy (LEFT-JUSTIFIED)
DE	20-21		Energy uncertainty (LEFT-JUSTIFIED)
SPACE	22	✓	Readability space
RI	23-29		Relative photon intensity (LEFT-JUSTIFIED at col 23)
DRI	30-31		Uncertainty in RI (LEFT-JUSTIFIED, including GT, LT markers)
M	32-41		Multipolarity
MR	42-49		Mixing ratio
DMR	50-55		Uncertainty in MR (LEFT-JUSTIFIED)
CC	56-62		Conversion coefficient
DCC	63-64		Uncertainty in CC (LEFT-JUSTIFIED)
TI	65-74		Total transition intensity
DTI	75-76		Uncertainty in TI (LEFT-JUSTIFIED)
C	77		Comment flag

Critical: ENSDF files are parsed by automated systems requiring exact positions. One column off = data rejection.

UNCERTAINTY LEFT-JUSTIFICATION RULE: ALL uncertainties (DE, DRI, DMR, DCC, DTI, DT, DS, etc.) MUST be left-justified in their respective fields, just like the values themselves. Special markers (GT, LT) within uncertainty fields are also left-justified.

LEFT-JUSTIFICATION RULE: ALL values AND uncertainties MUST be left-justified within their respective fields. This includes:

- Energy values (E field) and their uncertainties (DE field)
- J-π values (spin-parity) and any associated uncertainties
- Half-life values (T field) and their uncertainties (DT field)
- RI values (relative intensity) and their uncertainties (DRI field)
- Mixing ratios (MR field) and their uncertainties (DMR field)
- Conversion coefficients (CC field) and their uncertainties (DCC field)
- All numerical and text values AND their uncertainties

Never right-justify or center ANY values OR uncertainties in ENSDF records!

Essential Rules

File Protection

- **NEVER** edit `.old` files (reference files from previous evaluation rounds)
- **NEVER** modify first/last line indentation or spacing in .ens files
- **ALWAYS** preserve "PN" line with its numeric value
- Make all edits between first and last line boundaries only

Column Positioning

- **J-π placement**: Always start at column 23, LEFT-JUSTIFIED (never add spaces that shift uncertainties)
- **Energy values**: LEFT-JUSTIFIED in their designated columns (10-19)
- **RI values**: Start at column 23, **LEFT-JUSTIFIED** in 7-char field (23-29)
- **DRI values**: Position at columns 30-31 (including special markers like GT, LT)
- **Half-life values**: LEFT-JUSTIFIED in T field (columns 40-49)
- **BR values**: Position at column 32 (N-records), LEFT-JUSTIFIED
- **NR values**: Columns 11-15 (N-records), LEFT-JUSTIFIED

CRITICAL: ALL values must be LEFT-JUSTIFIED within their respective fields - never right-justified or centered!

⚠ **CRITICAL COLUMN RULE**: When fixing a quantity's position to the correct columns, NEVER shift other field values to wrong columns!

- L-transfer values: Must stay in columns 56-64
- Spectroscopic factors: Must stay in columns 65-74
- Comment flags: Must stay in column 77
- Only adjust spacing between fields - never move field data to incorrect columns!

NSR Keynumber Formatting

- **In comments/records**: Second letter lowercase (`2023Bo17`, `2021Wa16`)
- **In headers/Q-records**: All uppercase (`2023BO17`, `2021WA16`)

Change Tracking

- **Always** update `.github/change.log` after significant changes
- **Never** create duplicate change.log files
- Use evidence-based documentation with specific line numbers
- **Never** document assumed changes - always verify with tools

ENSDF Special Characters

Superscripts/Subscripts

- `{+n}` → superscript (e.g., `{+35}Ar` → ^{35}Ar)
- `{-n}` → subscript (e.g., `T{-1/2}` → $T_{1/2}$)
- `{+-n}` → negative superscript (e.g., `{+-4}` → $^{-4}$)

Greek Letters

Lowercase: |a → α, |b → β, |g → γ, |d → δ, |e → ε, |l → λ, |m → μ, |n → ν, |p → π, |r → ρ, |s → σ, |t → τ, |w → ω **Uppercase**: |D → Δ, |G → Γ, |L → Λ, |P → Π, |S → Σ, |W → Ω

Mathematical Symbols

- |* → × (times), |? → ≈ (approx), |+ → ± (plus-minus), |- → ∓ (minus-plus)
- |< → ≤, |> → ≥, |' → °, |= → ≠, |@ → ∞
- |^ → ↑, |_ → ↓, |(→ ←, |) → →, |. → ∝, || → |

Important: Use |? for approximate values, never standalone ~ (except in names/mass notation)

Common Examples

- %(|e+|b{++})p → %(ε+β^+)p
- {+208}Pb({+36}S,{+35}S) → ^{208}Pb(^{36}S,^{35}S)
- |s(E({+3}He),|q) → σ(E(^{3}He),θ)

Academic Standards

Citation Tense

Use PAST tense for all references to completed studies:

- ✓ "Authors stated...", "1994FO04 measured...", "Previous evaluators concluded..."
- ✗ "Authors state...", "1994FO04 measures..."

Grammar Fixes

Common corrections: "stoped"→"stopped", "usign"→"using", "coeffcients"→"coefficients"

Nuclear Data Evaluation

General Comment Ordering (adopted.ens files)

1. **Isotope discovery** (reference): experimental details
2. **{+A}X production**: production methods and studies
3. **{+A}X decay measurements**: half-life, decay modes
4. **{+A}X radius measurement**: nuclear radius determinations
5. **{+A}X mass measurements**: mass spectrometry, Q-values
6. **Theoretical calculations**: models, predictions (always last)

L-Transfer Rules for J-π Assignment

- L=0 → J-π: 1/2+
- L=1 → J-π: 1/2-,3/2-
- L=2 → J-π: 3/2+,5/2+
- L=3 → J-π: 5/2-,7/2-

Note: Always confirm with experimental data; never enter L-values in J-π column.

Tools and Workflows

PDF Generation

```
# Single element
Set-Location "D:\X\ND\Files"
$element = "Al"
Get-ChildItem "D:\X\ND\A35\finished\${element}35\new\*.ens" | ForEach-Object {
    java -jar "D:\X\ND\McMaster-MSU-Java-
NDS\McMaster_MSU_JAVA_NDS_v3.0_01May2025.jar" $_.FullName "$($_.BaseName).pdf"
}

# All elements
$elements = @("Al", "Ar", "Ca", "K", "Mg", "Na", "Ne", "P", "Si")
foreach ($element in $elements) {
    Get-ChildItem "D:\X\ND\A35\finished\${element}35\new\*adopted.ens" | ForEach-
Object {
        java -jar "D:\X\ND\McMaster-MSU-Java-
NDS\McMaster_MSU_JAVA_NDS_v3.0_01May2025.jar" $_.FullName "$($_.BaseName).pdf"
    }
}
```

Change Detection Process

1. **Pre-work (MANDATORY)**: `git status`, `git diff --name-only HEAD`
2. **During work**: Track file modifications systematically
3. **Post-work**: Use all detection tools on ALL files from git status
4. **Documentation**: Evidence-based change.log entries with line numbers

CRITICAL REMINDER: Always start with `git status` - this shows the complete picture!

File Categories to Track

- **ENSDF source files**: *.ens files (most important)
- **Generated PDFs**: *.pdf files (expected to change when source changes)
- **Processing artifacts**: temp/. files (expected, document but don't commit)
- **Tools and scripts**: .github/. files (important for tooling changes)
- **Documentation**: README.md, change.log, etc.

Evidence-Based Documentation Rules

Every change log entry should be backed by:

- Specific file diffs from `git diff HEAD~1 "filename"`
- Line numbers where changes occurred
- Actual before/after content when significant
- Explanation of why the change was made

Key principle: Use multiple detection methods and always cross-verify. If git shows a file changed, dig deeper with git diff. If you modified an ENSDF file, expect to see corresponding PDF changes.

Verification Checklist

- [ ] **FIRST**: `git status` - identify ALL modified files (MANDATORY)
- [ ] `git diff --name-only HEAD` - complete list verification
- [ ] `git ls-files --others --exclude-standard` - untracked files
- [ ] `git diff HEAD~1 "filename"` on each modified file from git status
- [ ] For moved files: `git show HEAD~1:old/path/file | Select-Object -First 20` (PowerShell)
- [ ] For large outputs: Use `Select-Object -First N` to limit output in PowerShell
- [ ] Update `change.log` with evidence-based entries
- [ ] Document file movements/reorganizations with full context
- [ ] Comprehensive commit message
- [ ] Cross-check: did any ENSDF changes result in expected PDF updates?

Remember: Start every workflow with git status and use PowerShell-compatible commands!

Git Commit Template

```
Title: Brief description of main changes

Summary:
- Enhanced/improved/fixed major components
- Scientific content updates in specific files

ENSDF Tools:
- tool_name.py: Specific improvements and validation results

Scientific Content:
- file_name.ens: Changes with line numbers and rationale

Processing Artifacts:
- PDF files: Regenerated files listed
- Temp files: Expected analysis output updates

Files changed: X modified, Y untracked
Brief scope and impact summary
```

Example Commit Structure

```
Title: Enhance ENSDF column calibration tools and improve Ar35 scientific content

Summary:
- Enhanced Python column calibration script with complete 80-column ENSDF format
support
- Improved scientific content and formatting in Ar35 ENSDF files
- Completed comprehensive change tracking and documentation

ENSDF Tools:
- column_calibrate.py: Extended from 41-column to complete 80-column ENSDF support
- check_averages.py: Completed and tested average calculation verification tool

Scientific Content:
```

```
- Ar35_36ar_p_d.ens: Fixed grammar in L=3 vs L=2 comparison (line 77)
- Ar35_adopted.ens: Multiple scientific and formatting enhancements

Processing Artifacts:
- PDF files: Regenerated Ar35_36ar_3he_a.pdf, Ar35_36ar_p_d.pdf, Ar35_adopted.pdf
- Temp files: Updated all analysis outputs (35.err, 35.fed, 35.fmt, etc.)

Files changed: 15 modified, 2 untracked
Completion of comprehensive ENSDF column calibration tooling and systematic
improvement of Ar35 nuclear data content.
```

Project Structure

Core Files (Most Critical)

- `finished/[Element]/new/*.ens` - Primary ENSDF source files
- `.github/change.log` - Comprehensive change tracking

Generated Files (Expected to Change)

- `finished/[Element]/pdf/*.pdf` - Generated from .ens files
- `finished/[Element]/temp/*.*` - Analysis tool artifacts

Tools

- `ens2pdf.py` - Python script for automated ENSDF to PDF conversion
- `.github/column-calibrate.ps1` - PowerShell column validator
- `.github/column_calibrate.py` - Python column validator
- `.github/check_averages.py` - Average calculation validator

Reference Files (NEVER EDIT)

- `*.old` files - Previous evaluation rounds, keep untouched

Focus Areas

Current Priority: K35 and P35 files (Ar35 completed)

Quality Assurance: Use Self-Calibrate Columns before any ENSDF edits, use What changed? after any modifications

Remember: Nuclear data accuracy is critical - when in doubt, verify with tools and cross-check against ENSDF Manual specifications.

Image Data Extraction Protocol

Level Scheme Analysis

- **Systematic scanning**: Left-to-right, top-to-bottom approach
- **Energy identification**: Clear notation for parentheses, uncertainties, tentative assignments

- **Color coding**: Black (known) vs Red (new) vs other markings
- **Special notations**: Asterisks (*), question marks (?), parentheses ()
- **Cross-verification**: Compare extracted data with tabulated lists

Spectral Analysis

- **Peak identification**: Exact energy labels, not estimates
- **Gate verification**: Check coincidence logic with nuclear structure
- **Contamination markers**: Identify non-target nuclide peaks
- **Quality indicators**: Intensity, resolution, background

Quality Control

- **Never guess or interpolate** energy values
- **Admit uncertainty** when image quality is poor
- **Section-by-section verification** before final compilation
- **Cross-check** with provided data tables

DCO Ratio and Polarization Analysis

Essential for multipolarity assignments in gamma-ray spectroscopy

DCO Ratio Rules

- **DCO(D) ≈ 1.0** → Dipole transition (M1, E1, or M1+E2 with dominant M1)
- **DCO(D) ≈ 1.6** → Quadrupole transition (E2 or M2)
- **DCO(Q) ≈ 1.0** → Quadrupole transition (E2 or M2)
- **DCO(Q) ≈ 0.6** → Dipole transition (M1, E1, or M1+E2 with dominant M1)

Polarization Rules

- **POL > 0** → Electric transition (E1, E2, etc.)
- **POL < 0** → Magnetic transition (M1, M2, etc.)
- **POL ≈ 0** → Mixed transition or measurement uncertainty

Quality Control Guidelines

- **Expected DCO ranges**: 0.4-1.4 for dipole, 0.8-1.8 for quadrupole
- **Red flags**: DCO > 2.0 or DCO < 0.3 (possible contamination or experimental issues)
- **Borderline values**: 0.8-1.2 may require additional analysis
- **Cross-verification**: Always check DCO consistency with nuclear structure logic

Systematic Analysis Protocol

1. **Extract all DCO and POL data** from experimental comments
2. **Apply rules systematically** to each transition
3. **Identify inconsistencies** between assigned multipolarity and DCO/POL
4. **Flag unusual values** (DCO > 2.0) for further investigation
5. **Document findings** with specific energy, DCO value, and recommended assignment