PDB format considerations

- **Fixed-format PDB files**
  - **Advantages**
    - **Easy & fast parsing, manual editing, visual inspection**
  - **Problem**
    - **Does not scale to larger structures**
- **Alternatives**
  - **Diverse approaches**
    - **mmCIF (but no breakthrough after 20+ years)**
    - **XML (extremely verbose, slow parsing, not human-readable)**
    - **Wide-PDB (http://biomol.dowling.edu/WPDB/)**
  - **Problem common to all approaches**
    - **Incompatible with old format -> huge investment in time**
    - **Nobody in a position to dictate that everybody spends time+money to quickly move the entire community over to something new**

Format evolution adopted in PHENIX

- We need a working format that scales to larger structures
  - Not to be confused with deposition format accepted by wwpdb.org members
- Where are the most pressing problems?
  - More than 100k atoms (27 structures in PDB, Oct 2007)
  - Only 62 “official” chainid characters
    - Automatic model building needs more (George Sheldrick)
  - More than 10k residues in one chain
- Maximally backward compatible extensions
  - Most users will not see a change
  - Visible changes only for
    - very large structures
    - or if researchers intentionally use the extensions
      - e.g. for more meaningful chainids

chainid AB

- Use ATOM columns 21-22 for two-character chainid
  - Suggested by George Sheldrick (many chain fragments)
  - Standard is just column 22
  - Thorough examination of PDB: two-character chainid compatible with all PDB records in which they appear

```
DBREF SEQADV SEQRES MODRES HET HELIX SHEET TURN SSBOND
LINK CISPEP SITE ATOM SIGATM ANISOU SIGUIJ TER HETATM
```

- Backward compatibility
  - Writing: chainid right-adjusted, e.g. chainid A written as " A"
  - Reading: only strip leading spaces!

```
ATOM    369 PEAK PEAK    1      61.114  12.134   8.619  1.00 20.00      PEAK
ATOM    504 SITE SITE    2      67.707   2.505  14.951  1.00 20.00      SITE
                    **
```

  - Preserving trailing spaces maximizes backward compatibility
    - Reading-writing cycle preserves intended meaning

serial numbers = strings

- Atom serial numbers: 5 columns
  - referenced in CONECT records
- Residue sequence numbers: 4 columns
  - referenced in several other records (e.g. LINK)
- Basic idea for maximum flexibility
  - Simply preserve strings
  - Convert from/to integers “just in time”
  - Conversion only needed for arithmetic
    - resseq 10:20 implies integer ordinal for each string
    - resseq + 1

“hybrid” serial numbers

- Assignment of integer ordinals to strings
- “Hybrid-36” maximizes compatibility
  - Strings that look like integers: no change (base-10)
    - `-999` to `9999`
  - Only if we run out of colums: switch to upper-case base-36
    `0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ`
  - First character is an upper-case letter
  - Examples: `A000`, `A001`, `A002`, …, `ZZZZ`
  - Only if upper-case exhausted: switch to lower-case base-36
  - First character is a lower-case letter
  - Examples: `a000`, `a001`, `a002`, …, `zzzz`
  - Mixed-case symbols intentionally avoided to minimize potential for confusion
  - Atom serial numbers: 87,440,031
  - Residue sequence numbers: 2,436,111

Practical advice

Golden Rule:

Preserve as much as possible, as long as possible.

- Preserve chainid trailing spaces
- Preserve serial number strings
- Convert to/from integers only if necessary
  - **Hide base-10 integers from users; convert back to strings for ouput!** The PDB format dictates that we show strings. To avoid confusion, show the same symbol in all contexts. Don't confuse users with a mix of base-10 symbols in one context and hybrid-36 symbols in another.

Use our open-source hybrid-36 implementations

- Python
- Java
- C/C++
- Fortran
- Drop-in replacement of built-in string<->integer conversions
- **NO** dependencies other than compiler or interpreter
- **NO** strings attached

Resources

Open-source hybrid-36 implementations:

`http://cci.lbl.gov/hybrid_36/`

Fast C++ PDB parser with Python interface (`iotbx.pdb.input`)

`http://cci.lbl.gov/publications/download/iucrcompcomm_nov2006.pdf`

Contact:

Ralf W. Grosse-Kunstleve

`rwgk@cci.lbl.gov`

September/October 2007