Derived Type I/O issues resolved

- **The EOF, EOR, and ERR dummy arguments to the derived type I/O routines will be default LOGICAL type**
  Avoids name space pollution due to implicitly defined derived type alternative
- Unformatted I/O: the READs/WRITEs in the DT I/O routine will be implicitly non-advancing (i.e. won't start a new record), but formatted READs/WRITEs are implicitly advancing
- All internal I/O (reads and writes) will be permitted within a DT I/O routine (or routine called therefrom)
- Will add INTENTs for various arguments.

- **UNIT #:** we'll only pass in ONE unit #, a default integer. If the original READ/WRITE was to a external unit (other than *), the dummy arg will have that value, otherwise a negative value. Removed restriction about only using the actual dummy arg in recursive i/o statements
- Will add an ERRMSG intent out dummy arg, for returning an error string. Library can print this if original user I/O stmt did not have an ERR= nor an IOSTAT=. (might add ERRMSG= specifer to all I/O statements)
- INQUIRE by I/O length will not interact with user supplied D/T I/O routines. Will assume current rules for DT I/O.

- **State Information:** the BN/BZ/S/SP/SS/P state will be pushed (saved) and then "reset" before calling the users DT I/O routine, and set to the pushed state after the DT I/O routine returns. The DT I/O routine cannot inquire about these, but can freely change them as it wishes.
- Will have separate routines for formatted vs. unformatted. Will NOT have separate routines internal vs. external (no longer needed) (new internal I/O support makes this restriction less onerous).
- Deleted "rec" dummy arg, not needed (record#)

ASYNC I/O Issues Resolved

- **any list item (from an I/O list in an ASYNC I/O stmt) must have the ASYNC attribute in any routine it is used within, if ANY code in that routine is executed when I/O is pending. This ASYNC attribute will force pass by descriptor. Copyin/copyout prohibited.**
- **List items will be "processed" when the original I/O stmt is executed (will compute addresses of ech list item). Will force READ(...)I,A(I) to be synchronous.**
- **Can't do async I/O on the "*" unit.**