

```

READ ( UNIT=u, FMT=f, IOSTAT=ivar, ERR=lab, END=lab, ADVANCE=str, SIZE=ivar, EOR=lab ) &
&      <E/A-Liste>                ! Listenelemente muessen VARIABLEN sein (l-values) !
WRITE ( UNIT=u, FMT=f, IOSTAT=ivar, ERR=lab, ADVANCE=str ) <E/A-Liste>    ! (r-values) !

```

Der Datentransfer erfolgt mittels **READ**-Anweisung (zum Einlesen/Eingeben/Lesen von Daten aus einer Datei in den Hauptspeicher) bzw. mittels **WRITE**-Anweisung (zum Auslesen/Ausgeben/Schreiben von Daten vom Hauptspeicher in eine Datei) in folgenden Schritten:

1. **Transferrichtung** (Ein- oder Ausgabe) wird durch Anweisungstyp (**READ** oder **WRITE**) bestimmt: von Datei in den Speicher oder vom Speicher in eine Datei.
2. **Ein-/Ausabeeinheit (unit)** bestimmen: Die unit muss mit einer externen Datei verbunden (connected) oder eine interne Datei (d.h. Zeichenkettenvariable) sein. **UNIT=\*** bedeutet beim Lesen die Tastatur und beim Schreiben den Bildschirm/Monitor; diese existieren immer und sind von Anfang an (ohne **OPEN**) verfügbar (preconnected files).
3. **Transfermodus** bestimmen:
  - (a) unformatiert (binär als direktes Speicherabbild) oder formatiert (zeichenweise als lesbarer Text)?
  - (b) falls formatiert: listengesteuert (vordefinierte Formattierung, **FMT=\***) oder formatgesteuert (explizite Formattierung, **FMT=<Zeichenkettenausdruck>** oder **<Marke>**)?
  - (c) bei expliziter Formattierung: Bestimmung des Format-Strings (Auswertung des Zeichenkettenausdrucks oder Verbindung herstellen mit der **FORMAT**-Anweisung mit der entsprechenden Marke).
4. **Positionierung** der Datei (vor dem Datentransfer): die aktuelle Position verbleibt direkt an der zuletzt gelesenen/geschriebenen Stelle, wenn diese sich nicht am Ende einer Zeile befindet, ansonsten wird beim Schreiben eine neue Zeile (ein neuer Datensatz) generiert und sowohl beim Schreiben als auch beim Lesen die aktuelle Position an den Anfang der nächsten Zeile gesetzt.
5. **eigentlicher Datentransfer** zwischen den Größen in der **<E/A-Liste>** und den Größen in der mit der E/A-Einheit (unit) verbundenen externen Datei oder der verwendeten internen Datei, ggf. unter Berücksichtigung der Formattierung.
6. **Fehlererkennung**: Der Fehlerzustand **ERR** (error) tritt z.B. bei Datei- oder Datenproblemen ein.  
 Nur beim Lesen: der Dateiende-Status **EOF** (end of file) tritt ein, wenn man versucht, über das Ende der Datei hinaus (eine nicht vorhandene Zeile) zu lesen.  
 Nur beim Lesen mit **ADVANCE='NO'**: der Zeilenende-Status **EOR** (end of record) tritt ein, wenn man versucht, über das Ende der aktuellen Zeile hinaus zu lesen.  
**Wichtig:** Die Zustände **EOF** bzw. **EOR** treten nicht schon nach dem Lesen des letzten Datenobjekts, sondern erst nach einem fehlgeschlagenen Versuch, weitere Daten einzulesen, ein; die einzulesenden Variablen haben dann einen **undefinierten Zustand**.
7. **Positionierung** der Datei (nach dem Datentransfer): bei **EOF** hinter dem **ENDFILE**-Datensatz, bei **EOR** am Ende der aktuellen Zeile, bei Fehlern ist die aktuelle Dateiposition ab diesem Zeitpunkt undefiniert. Ist kein Fehler aufgetreten, verbleibt bei **ADVANCE='NO'** die aktuelle Position hinter dem zuletzt gelesenen/geschriebenen Zeichen/Byte, bei **ADVANCE='YES'** oder ohne **ADVANCE** wird das Lesen/Schreiben der aktuellen Zeile beendet (beim Schreiben wird ein Zeilenende generiert); aktuelle Position am Zeilenende.
8. Wertzuweisung des (systemabhängigen) **Fehlercodes** an die (ganzzahlige) **IOSTAT**-Variable, falls eine solche vorhanden ist: *negative* Werte bedeuten **EOF** oder **EOR**, *positive* signalisieren einen Fehler, der Wert 0 bedeutet, dass alles normal gelaufen ist.  
 Nur beim Lesen mit **ADVANCE='NO'**: Wertzuweisung der Anzahl der mit dieser **READ**-Anweisung auf der aktuellen Zeile gelesenen Zeichen an die **SIZE**-Variable, falls vorhanden.
9. **Fehlerbehandlung** und ggf. Sprung zu einer anderen Anweisung oder Programmabbruch:
  - falls **EOF**-Status eingetreten ist und **END=<Marke>** angegeben wurde: Sprung zur Anweisung mit dieser Marke;
  - falls **EOR**-Status eingetreten ist und **EOR=<Marke>** angegeben wurde: Sprung zur Anweisung mit dieser Marke;
  - falls ein Fehler aufgetreten ist und **ERR=<Marke>** angegeben wurde: Sprung zur Anweisung mit dieser Marke.
 Falls ein Fehlerzustand eingetreten ist und dieser nicht durch eine entsprechende Spezifikation oder durch eine **IOSTAT**-Spezifikation (diese verhindert grundsätzlich einen Programmabsturz) in der Kontrollliste behandelt wird, bricht das Programm ab.

- $n$  Gesamtfeldbreite für auszugebenden Wert (Anzahl Zeichen/Spalten),  $> 0$   
 $m$  Minimalfeldbreite für ganzzahligen Wert,  $\geq 0$   
 $d$  Anzahl Dezimalziffern nach dem Dezimalpunkt,  $\geq 0$   
 $e$  Anzahl Dezimalziffern zur Exponentendarstellung,  $> 0$ , in der Regel = 2 oder 3  
 $s$  „Skalierfaktor“, beliebige ganze Zahl

$In$	Ausgabe eines INTEGER-Werts rechtsbündig in $n$ Spalten.
$In.m$	Ausgabe eines INTEGER-Werts rechtsbündig in möglichst wenig, jedoch mindestens $m$ (evtl. mit führenden Nullen) und höchstens $n$ Spalten.
$Fn.d$	Rechtsbündige Ausgabe eines REAL-Werts möglichst in <i>Festkommadarstellung</i> , d.h. ohne Exponent, in $n$ Spalten mit $d$ Stellen nach dem Dezimalpunkt.
$En.d[Ee]$	Rechtsbündige Ausgabe eines REAL-Werts in <i>Gleitkommadarstellung</i> in $n$ Spalten mit $d$ Stellen nach dem Dezimalpunkt. Der Exponent wird auf jeden Fall ausgegeben. Die Darstellung ist normalisiert, d.h. die Mantisse beginnt mit „0.“ gefolgt von der ersten signifikanten Mantissenziffer. Falls $Ee$ angegeben ist, wird der Exponent mit $e$ Ziffern ausgegeben, ansonsten mit 2 oder 3.
$ENn.d[Ee]$	Rechtsbündige Ausgabe eines REAL-Werts in <i>Engineering-Gleitkommadarstellung</i> in $n$ Spalten mit $d$ Stellen nach dem Dezimalpunkt. Dabei ist der Exponent ein Vielfaches von 3 und der Betrag der Mantisse liegt im Intervall $[1, 1000)$ , falls die Zahl nicht Null ist. Es werden immer 3 Positionen für die Vorkommastellen verwendet, auch wenn eine oder zwei leer sind. Exponentenbreite $Ee$ wie beim E-Format.
$ESn.d[Ee]$	Rechtsbündige Ausgabe eines REAL-Werts in <i>Scientific-Gleitkommadarstellung</i> in $n$ Spalten mit $d$ Stellen nach dem Dezimalpunkt. Dabei liegt der Betrag der Mantisse im Intervall $[1, 10)$ , falls die Zahl nicht Null ist, es wird also immer eine signifikante Vorkommaziffer verwendet. Exponentenbreite $Ee$ wie beim E-Format.
$Dn.d$	wie E-Format, aber mit Exponentenzeichen „D“ statt „E“ (früher für DOUBLE PRECISION).
$Gn.d[Ee]$	Allgemeines Format („general“), für alle intrinsischen Datentypen geeignet ( $d$ und $e$ werden bei INTEGER, LOGICAL, CHARACTER ignoriert). Bei REAL und COMPLEX: Ist der Wert in Festpunkt darstellbar, so erfolgt die Ausgabe wie mit dem F-Format, ansonsten wie mit dem E-Format. In Festpunktdarstellung bleibt der Platz für den Exponenten leer.
$Ln$	Ausgabe eines LOGICAL-Werts rechtsbündig in $n$ Spalten. Für .TRUE. wird ein T ausgegeben, für .FALSE. ein F.
$A$	Ausgabe aller Zeichen einer Zeichenkette.
$An$	Rechtsbündige Ausgabe einer Zeichenkette in $n$ Spalten. Ist die Zeichenkette länger, so werden <i>nur</i> die ersten $n$ Zeichen ausgegeben.
$nX$	Ausgabe von $n$ Leerzeichen.
$sP$	Alle nachfolgenden D-, E-, F- und G-Formate werden skaliert. Dies bedeutet für F-Format: Multiplikation des Ausgabewerts mit $10^s$ E-, D-Format: Multiplikation der Ausgabemantisse mit $10^s$ und Subtraktion von $s$ vom Ausgabexponenten G-Format: wirkt nur bei Ausgabe im E-Format (mit derselben Wirkung)
$Tn$	Fortsetzung der Ausgabe an (Spalten-)Position $n$ der laufenden Ausgabezeile.
$TLn$	Ausgabe wird $n$ Spalten <i>links</i> der aktuellen Zeilenposition fortgesetzt.
$TRn$	Ausgabe wird $n$ Spalten <i>rechts</i> der aktuellen Zeilenposition fortgesetzt.

- Durch Voranstellen einer positiven ganzen Zahl  $w$  gelten alle genannten Formatelemente (außer X, P und T) als  $w$ -mal hintereinander notiert (Wiederholungsfaktor). Mehrere Listenelemente können geklammert und mit einem Wiederholungsfaktor vor der Klammer versehen werden.
- Wird  $sP$  einem D-, E-, F- oder G-Format *unmittelbar*, d.h. ohne trennendes Komma, vorangestellt, so wirkt die Skalierung nur für dieses Formatlistenelement.
- Ein Schrägstrich / in der Formatliste bewirkt einen Zeilenvorschub (Beginn einer neuen Zeile). Er kann gleichzeitig zur Trennung zweier Formatlistenelemente (statt Komma) verwendet werden.
- Ausgabertext kann, in Apostrophe / Hochkommata '... ' oder in Anführungszeichen / Doppelhochkommata "... " eingeschlossen, unmittelbar in die Formatliste aufgenommen werden. Wenn im Ausgabertext das Begrenzungszeichen (Hochkomma oder Doppelhochkomma) vorkommt, muß es verdoppelt werden.