Compute (Bridgend) Ltd

FILEKIT

z/OS Productivity Toolkit

New Features

Release 3.60

IBM Mainframe z/OS, VSE & VM/CMS Systems

8 Merthyr Mawr Road, Bridgend, Wales UK CF31 3NH

Tel: +44 (1656) 65 2222
Eml: support@cbl.com

CBL Web Site - www.cbl.com

This document may be downloaded from www.cbl.com/documentation.php

Contents

FileKit 3.60 New Features

Documentation Notes

Information in this New Features document details changes introduced in the FileKit (formerly SELCOPYi) component of the CBL Product Suite, since GA release 3.50.

CBL Product Suite for z/OS, z/VM (CMS) and z/VSE operating systems, which includes SELCOPY, SLC, FileKit and CBLVCAT, is available for download and install from **www.cbl.com/selcdl.php**.

The following publications for CBL Product Suite and its component products are available in Adobe Acrobat PDF format at CBL web page **www.cbl.com/documentation.php**:

- CBL Product Suite Customisation Guide
- SELCOPY User Manual
- SELCOPY SLCIMS Call Module Interface for IMS/DL1 Checkpoint/Restart
- SELCOPY C++ (SLC) Language Reference
- CBLVCAT User Manual
- FileKit Reference and User Guide
- FileKit Text Editor
- FileKit Data Editor (SDE)
- FileKit Quick Reference
- FileKit REPORT Utility
- FileKit SMF Utilities
- FileKit Training Manual

The following generic terms are used throughout this document to indicate all available versions and releases of IBM mainframe operating systems:

z/OS	-	z/OS, OS/390, MVS/ESA, MVS/XA, MVS/SP, OS.
z/VSE	-	z/VSE, VSE/ESA, VSE/SP, DOS.
z/VM CMS	-	z/VM, VM/ESA, VM/XA, VM/SP.
All	-	All z/OS, z/VSE and z/VM CMS operating systems.

FileKit - The new name for SELCOPYi

At release 3.60, **SELCOPYi** has been rebranded as **FileKit**.

This rebrand was considered worthwhile since the name, FileKit, more accurately ecapsulates the purpose of the software. FileKit is an online and batch application that includes a suite of tools and utilities to manage, interrogate and report on data in files and database tables.

FileKit is a file toolkit!

FileKit will continue to be included as part of the SELCOPY batch utility licence at no additional charge.

FileKit Random/Test Data Generator

The ability to mask and generate test data is a major new feature introduced in FileKit release 3.60.

The Test Data Utility provides the means to create accurate test data without loss of integrity. Test data output may comprise a mixture of copied and randomly generated values for the purpose of application testing and data analysis. Therefore, data fields that contain potentially sensitive values may be protected.

The utility uses FileKit's structured data processing to format input and output data in data set records and DB2 table rows.

Individual fields within data set records mapped by FileKit structures may be assigned a permanent or temporary randomizer object which is triggered when a value is written to that field. For example, when new records are inserted via the FILEIO or Data-Edit utilities, or when existing records are copied to another data set or updated using the File Search/Copy/Update (FSU) utility.

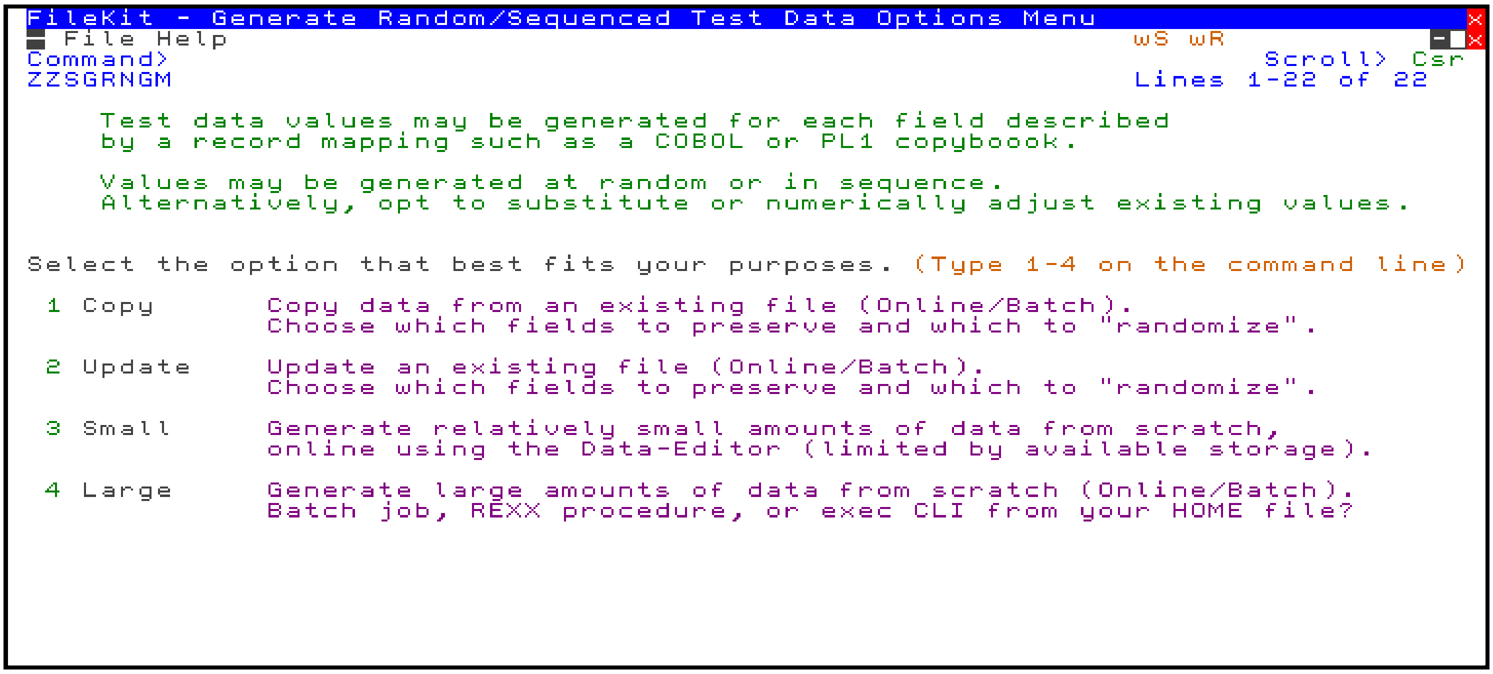
Test Data Panels

The Random/Test Data utility includes a number of user panel interfaces available via the online FileKit windowed environment.

Test Data Options Menu

The Test Data Menu panel (ZZSGRNGM) is an interactive panel window opened on selection of option 14. from the FileKit Primary option menu.

The individual items (Copy, Update, Small and Large) will open panels to specify output and/or input data sets, and the corresponding structure to be used to map output record fields or the input record fields to be updated.



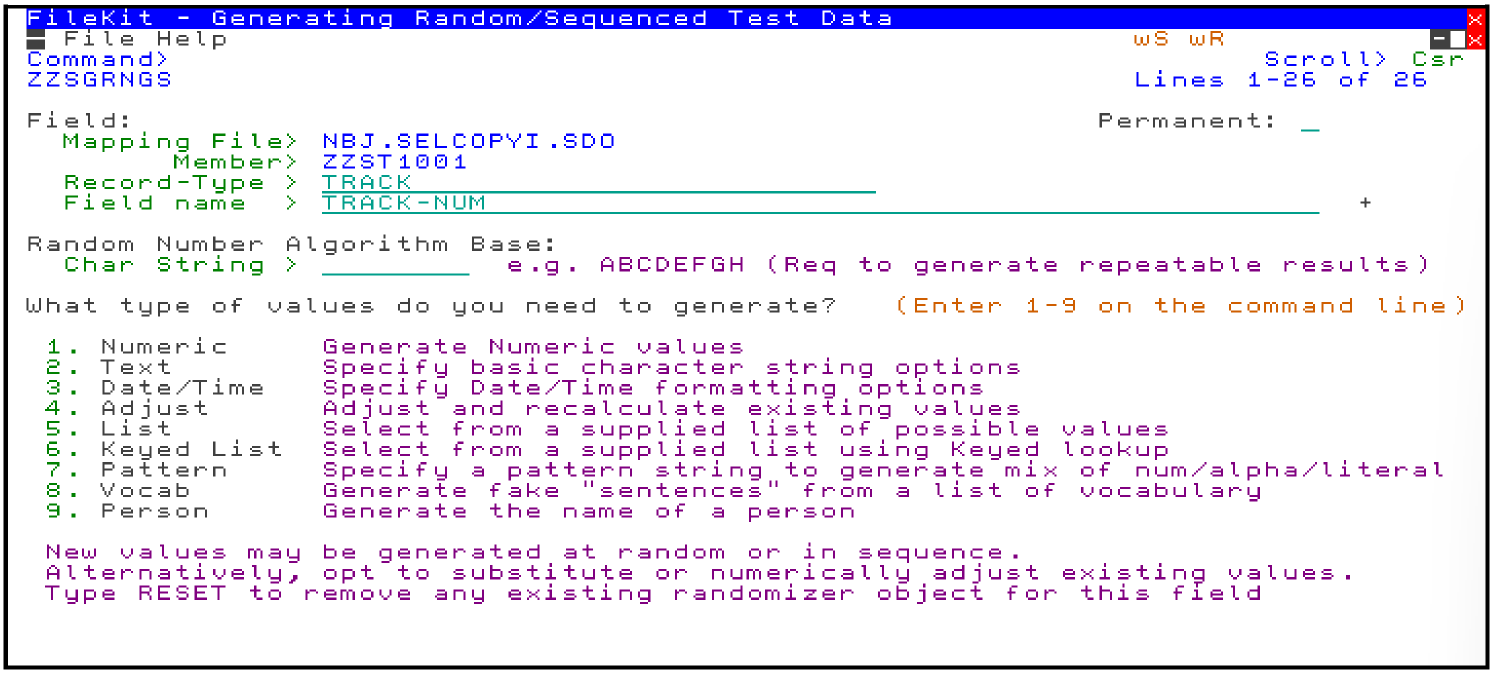
```
FileKit - Generate Random/Sequenced Test Data Options Menu
  File Help                                                    wS wR
Command>                                                            Scroll> Csr
ZZSGRNGM                                                          Lines 1-22 of 22

    Test data values may be generated for each field described
    by a record mapping such as a COBOL or PL1 copyboook.

    Values may be generated at random or in sequence.
    Alternatively, opt to substitute or numerically adjust existing values.

Select the option that best fits your purposes. (Type 1-4 on the command line)

 1 Copy        Copy data from an existing file (Online/Batch).
               Choose which fields to preserve and which to "randomize".

 2 Update      Update an existing file (Online/Batch).
               Choose which fields to preserve and which to "randomize".

 3 Small       Generate relatively small amounts of data from scratch,
               online using the Data-Editor (limited by available storage).

 4 Large       Generate large amounts of data from scratch (Online/Batch).
               Batch job, REXX procedure, or exec CLI from your HOME file?
```

Figure 1. Test Data Options Menu.

Generating Random/Sequenced Test Data

The Generate Random/Sequenced Test Data panel (ZZSGRNGS) is opened on execution of primary command "**GEN**" from one of the Copy, Update, Edit (Small) or Generate from Scratch (Large) with Randomizer panels. Note that a structure must have been specified before the "GEN" command is issued.

Having selected the record-type and field name, select one of the options (1-9) to generate a randomizer object for that field.

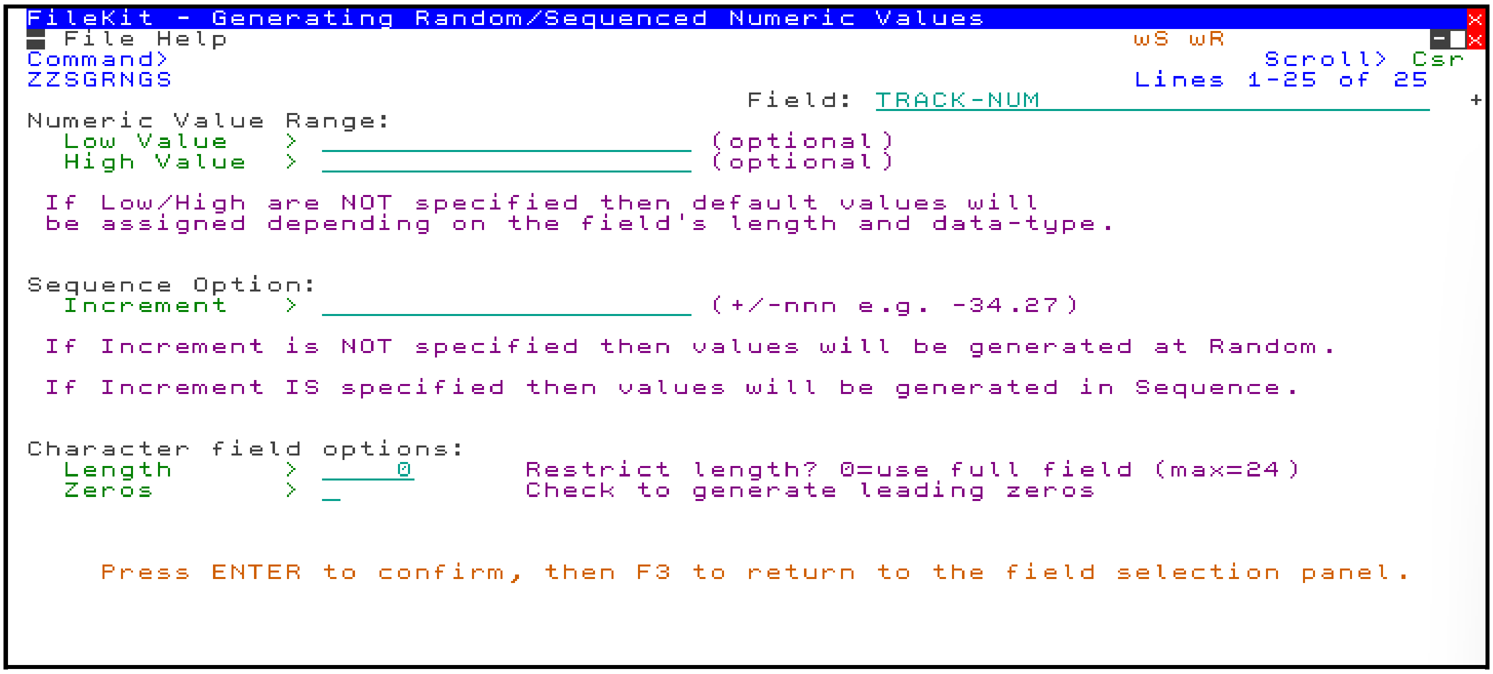


```
FileKit - Generating Random/Sequenced Test Data
 File Help                                                      wS wR
Command>                                                          Scroll> Csr
ZZSGRNGS                                                        Lines 1-26 of 26

Field:                                                        Permanent: _
   Mapping File>  NBJ.SELCOPYI.SDO
         Member>  ZZST1001
   Record-Type >  TRACK
   Field name  >  TRACK-NUM                                                    +

Random Number Algorithm Base:
   Char String >  ________   e.g. ABCDEFGH (Req to generate repeatable results)

What type of values do you need to generate?   (Enter 1-9 on the command line)

 1. Numeric      Generate Numeric values
 2. Text         Specify basic character string options
 3. Date/Time    Specify Date/Time formatting options
 4. Adjust       Adjust and recalculate existing values
 5. List         Select from a supplied list of possible values
 6. Keyed List   Select from a supplied list using Keyed lookup
 7. Pattern      Specify a pattern string to generate mix of num/alpha/literal
 8. Vocab        Generate fake "sentences" from a list of vocabulary
 9. Person       Generate the name of a person

 New values may be generated at random or in sequence.
 Alternatively, opt to substitute or numerically adjust existing values.
 Type RESET to remove any existing randomizer object for this field
```

Figure 2. Generating Random/Sequenced Test Data.

Generating Random/Sequenced Numeric Values

Option 1. will open the "Numeric Values" panel view which provides randomizer options for generating numeric values. The values may be generated at random within a range of supplied values, or in an ascending or descending sequence starting at the low value.

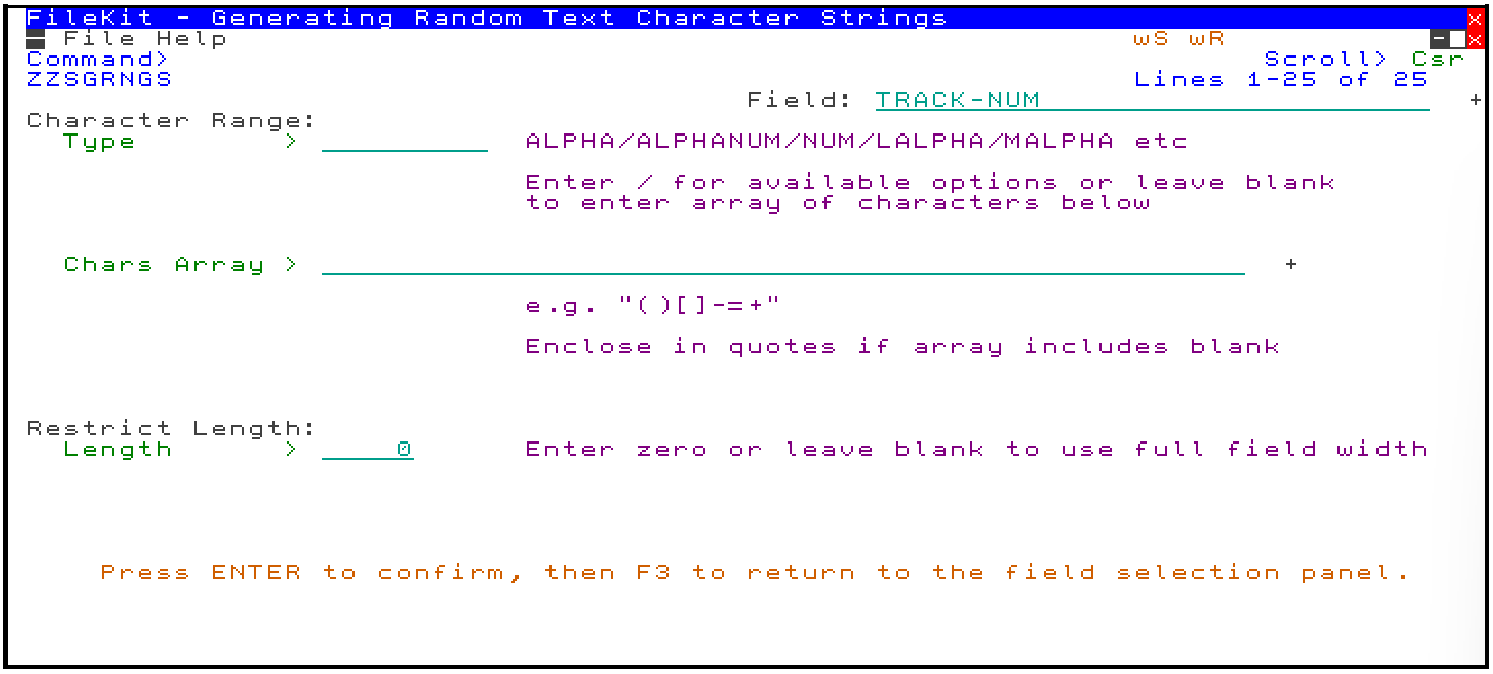


```
FileKit - Generating Random/Sequenced Numeric Values
 File Help                                                      wS wR
Command>                                                          Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                           Field: TRACK-NUM                     +
Numeric Value Range:
   Low Value   >  ____________________  (optional)
   High Value  >  ____________________  (optional)

 If Low/High are NOT specified then default values will
 be assigned depending on the field's length and data-type.


Sequence Option:
   Increment   >  ____________________  (+/-nnn e.g. -34.27)

 If Increment is NOT specified then values will be generated at Random.

 If Increment IS specified then values will be generated in Sequence.


Character field options:
   Length      >  _____0     Restrict length? 0=use full field (max=24)
   Zeros       >  _          Check to generate leading zeros



    Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 3. Generating Random/Sequenced Numeric Values.

Generating Random Text Character Strings

Option 2. will open the "Character Strings" panel view which provides randomizer options for generating text string values. The values are generated at random from a list of specified characters.

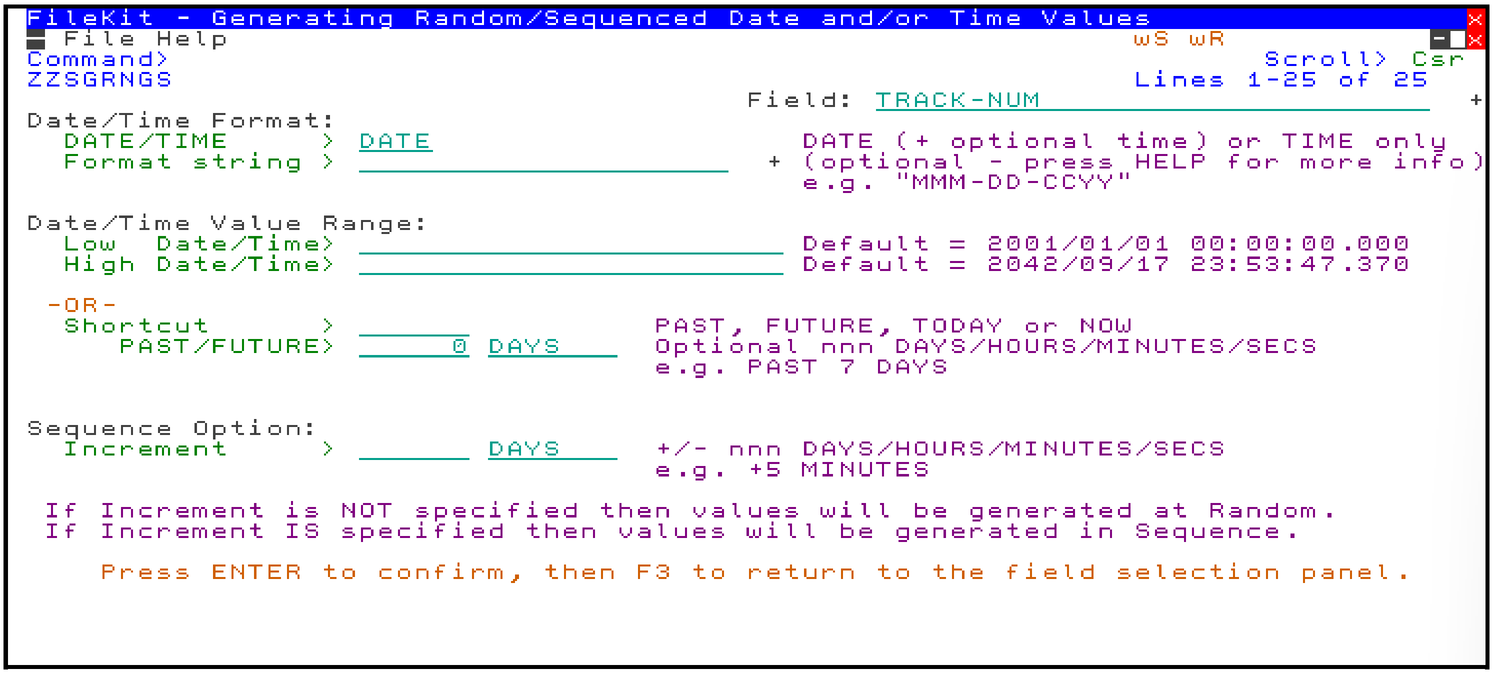


```
FileKit - Generating Random Text Character Strings
 File Help                                                     wS wR
Command>                                                              Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                           Field: TRACK-NUM                        +
Character Range:
   Type         > __________   ALPHA/ALPHANUM/NUM/LALPHA/MALPHA etc

                               Enter / for available options or leave blank
                               to enter array of characters below

   Chars Array > _______________________________________________________  +

                               e.g. "()[]-=+"

                               Enclose in quotes if array includes blank

Restrict Length:
   Length       > _____0       Enter zero or leave blank to use full field width

     Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 4. Generating Random Text Character Strings.

Generating Random/Sequenced Date and/or Time Values.

Option 3. will open the "Date and/or Time Values" panel view which provides randomizer options for generating date/time (timestamp) or time only values. The values may be generated at random within a range of supplied values, in an ascending or descending sequence starting at the low value, or as an offset from the current date/time.



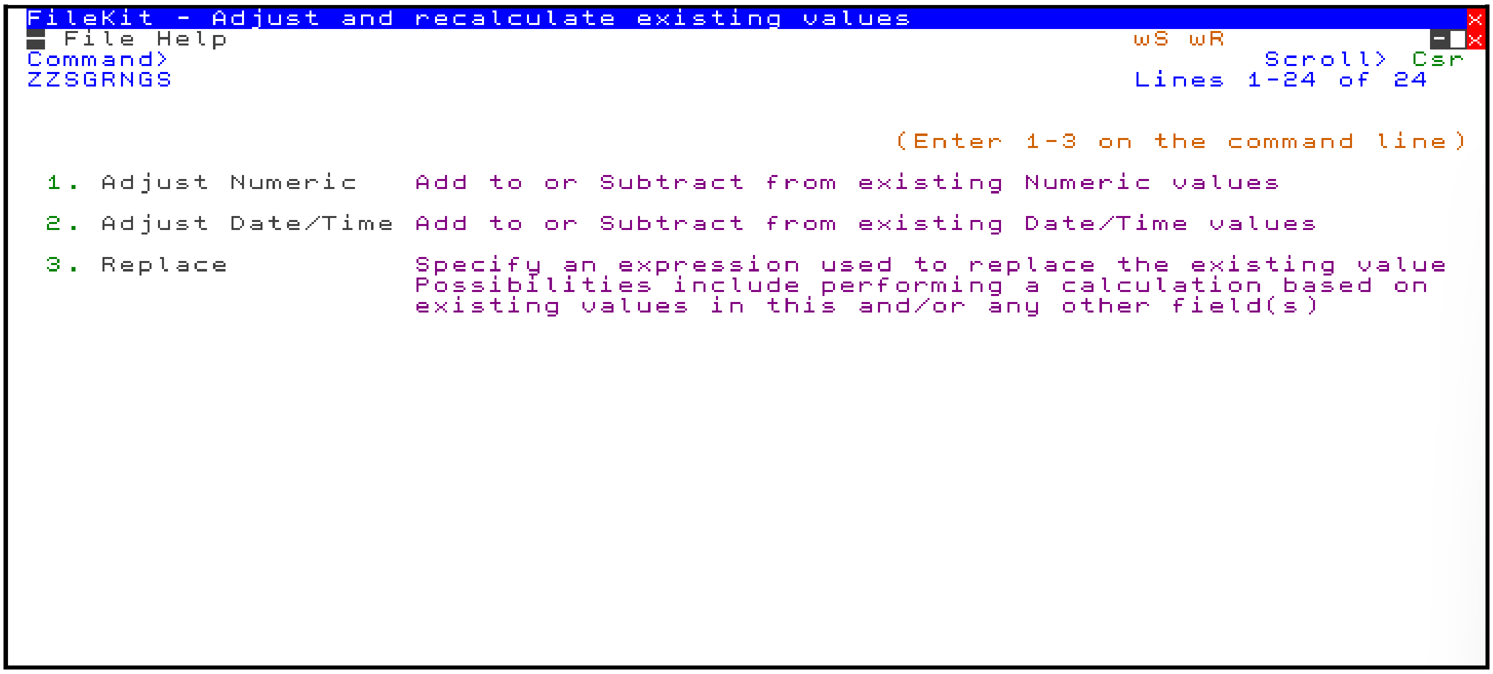
```
FileKit - Generating Random/Sequenced Date and/or Time Values
 File Help                                                     wS wR
Command>                                                              Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                           Field: TRACK-NUM                        +
Date/Time Format:
   DATE/TIME      > DATE                           DATE (+ optional time) or TIME only
   Format string  > ____________________       +  (optional - press HELP for more info)
                                                   e.g. "MMM-DD-CCYY"

Date/Time Value Range:
   Low  Date/Time> _______________________         Default = 2001/01/01 00:00:00.000
   High Date/Time> _______________________         Default = 2042/09/17 23:53:47.370

  -OR-
   Shortcut       > ______             PAST, FUTURE, TODAY or NOW
      PAST/FUTURE> _____0 DAYS___      Optional nnn DAYS/HOURS/MINUTES/SECS
                                       e.g. PAST 7 DAYS

Sequence Option:
   Increment      > ______ DAYS___     +/- nnn DAYS/HOURS/MINUTES/SECS
                                       e.g. +5 MINUTES

  If Increment is NOT specified then values will be generated at Random.
  If Increment IS specified then values will be generated in Sequence.

     Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 5. Generating Random/Sequenced Date and/or Time Values.

Adjust and recalculate existing values

Option 4. will open the "Adjust and recalculate existing values" menu panel view to replace the field contents with a value derived from its current value. Select one of the 3 options to open another panel view to specify parameters for adjusting the current field value. Use option 1. if the field contains numeric values, option 2. if the field contains date/time values, or option 3. if the field value is to be replaced with a value derived from a Data-Edit expression.

Note that a Data-Edit expressions may include arithmetic operators and variables (field names), and support a number of named, built-in functions.

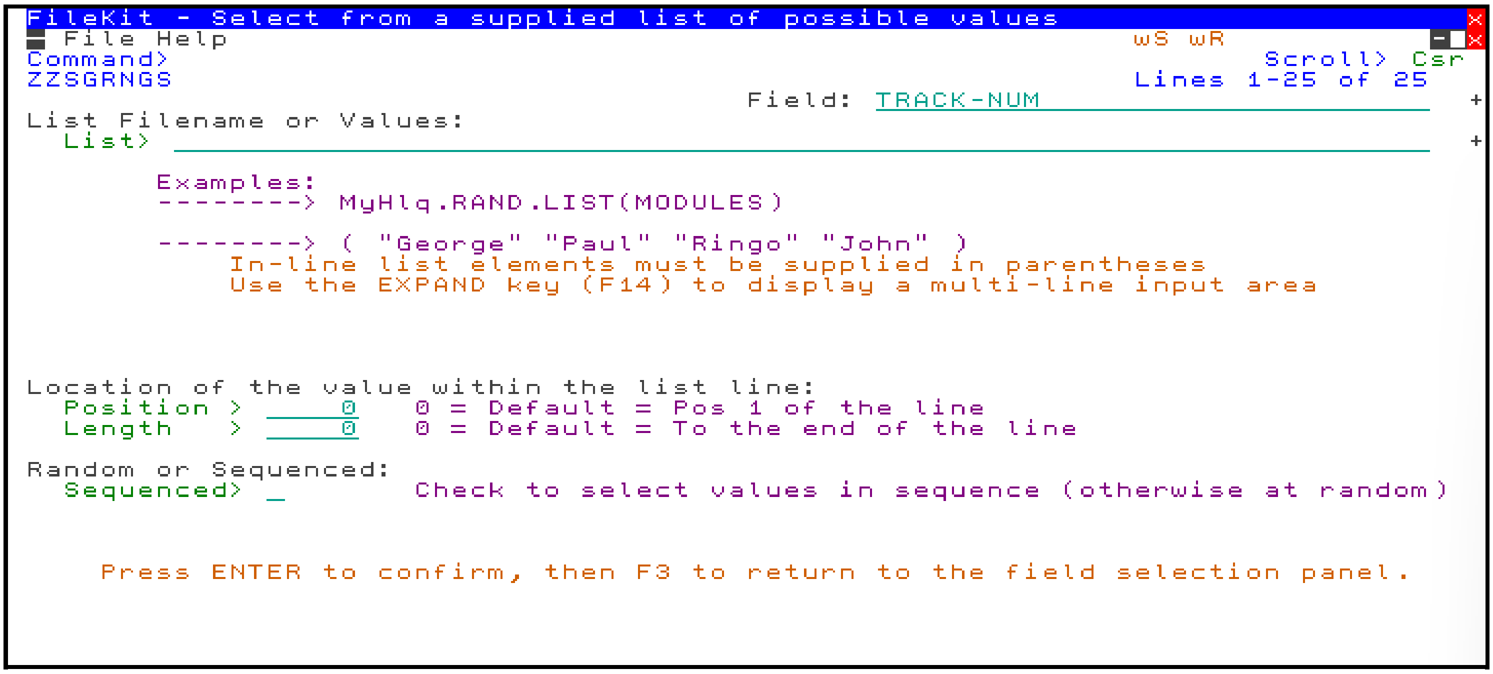


```
FileKit - Adjust and recalculate existing values
  File  Help                                               wS wR
Command>                                                       Scroll> Csr
ZZSGRNGS                                                    Lines 1-24 of 24

                                                   (Enter 1-3 on the command line)
 1. Adjust Numeric    Add to or Subtract from existing Numeric values

 2. Adjust Date/Time  Add to or Subtract from existing Date/Time values

 3. Replace           Specify an expression used to replace the existing value
                      Possibilities include performing a calculation based on
                      existing values in this and/or any other field(s)
```

Figure 6. Adjust and recalculate existing values.

Select from a supplied list of possible values

Option 5. will open the "Select Values from a Supplied List" panel view. The values will be generated at random or in sequence from the list of supplied values.

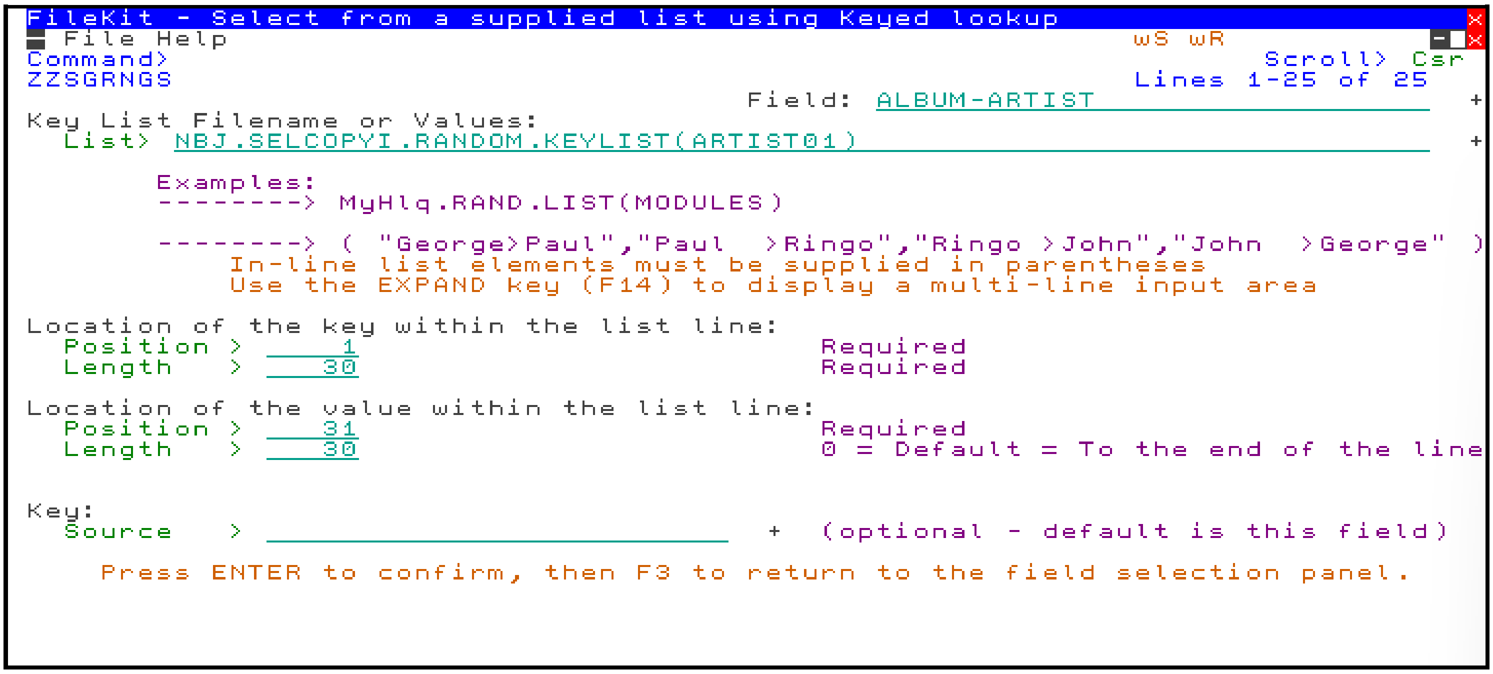


```
FileKit - Select from a supplied list of possible values
  File  Help                                               wS wR
Command>                                                       Scroll> Csr
ZZSGRNGS                                                    Lines 1-25 of 25
                                           Field: TRACK-NUM                      +
List Filename or Values:
   List> ___________________________________________________________________     +

         Examples:
         ---------> MyHlq.RAND.LIST(MODULES)

         ---------> ( "George" "Paul" "Ringo" "John" )
             In-line list elements must be supplied in parentheses
             Use the EXPAND key (F14) to display a multi-line input area


Location of the value within the list line:
   Position > _____0    0 = Default = Pos 1 of the line
   Length   > _____0    0 = Default = To the end of the line

Random or Sequenced:
   Sequenced> _         Check to select values in sequence (otherwise at random)


        Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 7. Select from a supplied list of possible values.

Select from a supplied list using Keyed lookup

Option 6. will open the "Select Values from a Supplied List using Keyed Lookup" panel view. The values will be selected from the list of supplied values that each have a unique key. By default, the current field value is used as the key lookup; however, the key value may be derived from a number of fields using a Data-Edit expression.

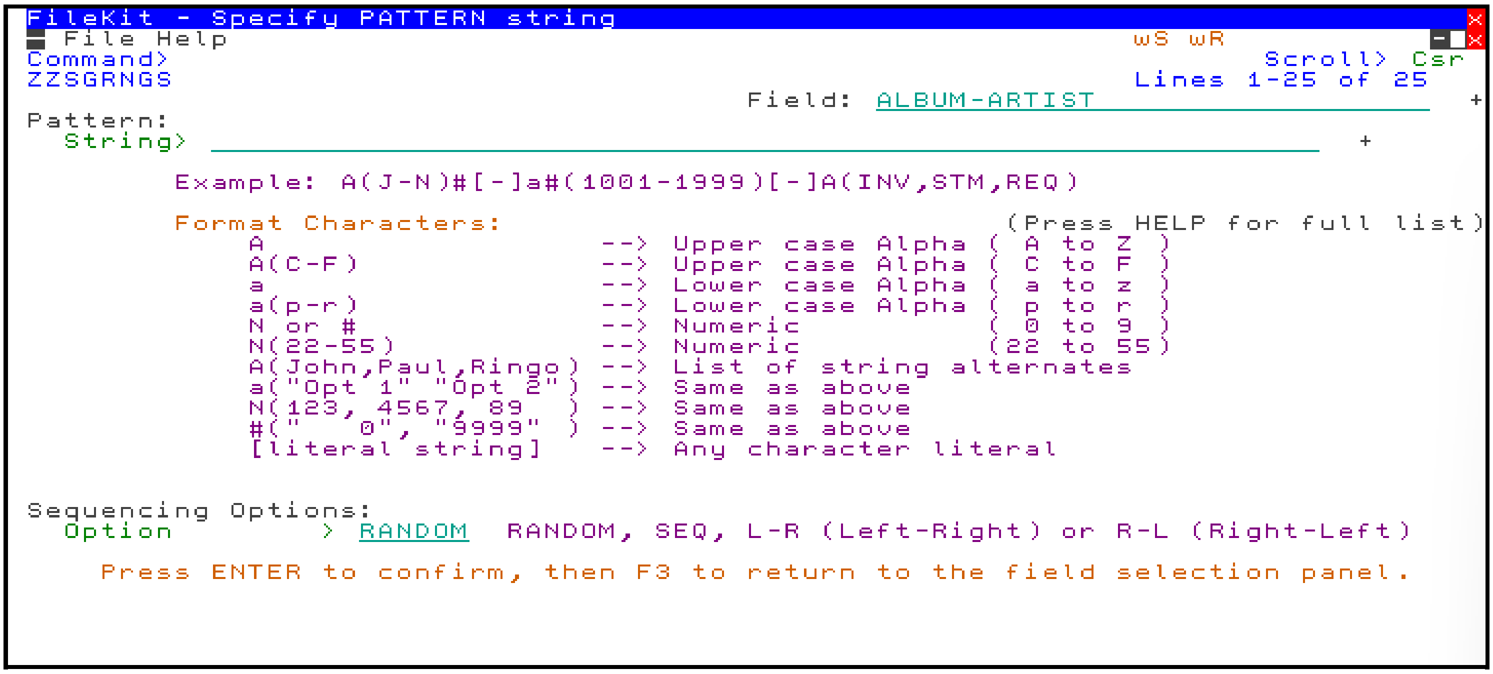


```
FileKit - Select from a supplied list using Keyed lookup
 File Help                                                    wS wR
Command>                                                             Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                          Field: ALBUM-ARTIST                  +
Key List Filename or Values:
   List> NBJ.SELCOPYI.RANDOM.KEYLIST(ARTIST01)                                 +

          Examples:
          ---------> MyHlq.RAND.LIST(MODULES)

          ---------> ( "George>Paul","Paul  >Ringo","Ringo >John","John  >George" )
              In-line list elements must be supplied in parentheses
              Use the EXPAND key (F14) to display a multi-line input area

Location of the key within the list line:
   Position >      1                            Required
   Length   >     30                            Required

Location of the value within the list line:
   Position >     31                            Required
   Length   >     30                            0 = Default = To the end of the line


Key:
   Source   >                         +  (optional - default is this field)

     Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 8. Select from a supplied list using Keyed lookup.

Specify PATTERN string

Option 7. will open the "Pattern String" panel view whereby values are generated in a format that matches a pattern containing varying and fixed elements. Varying elements specify a range of values that may be random or sequential.

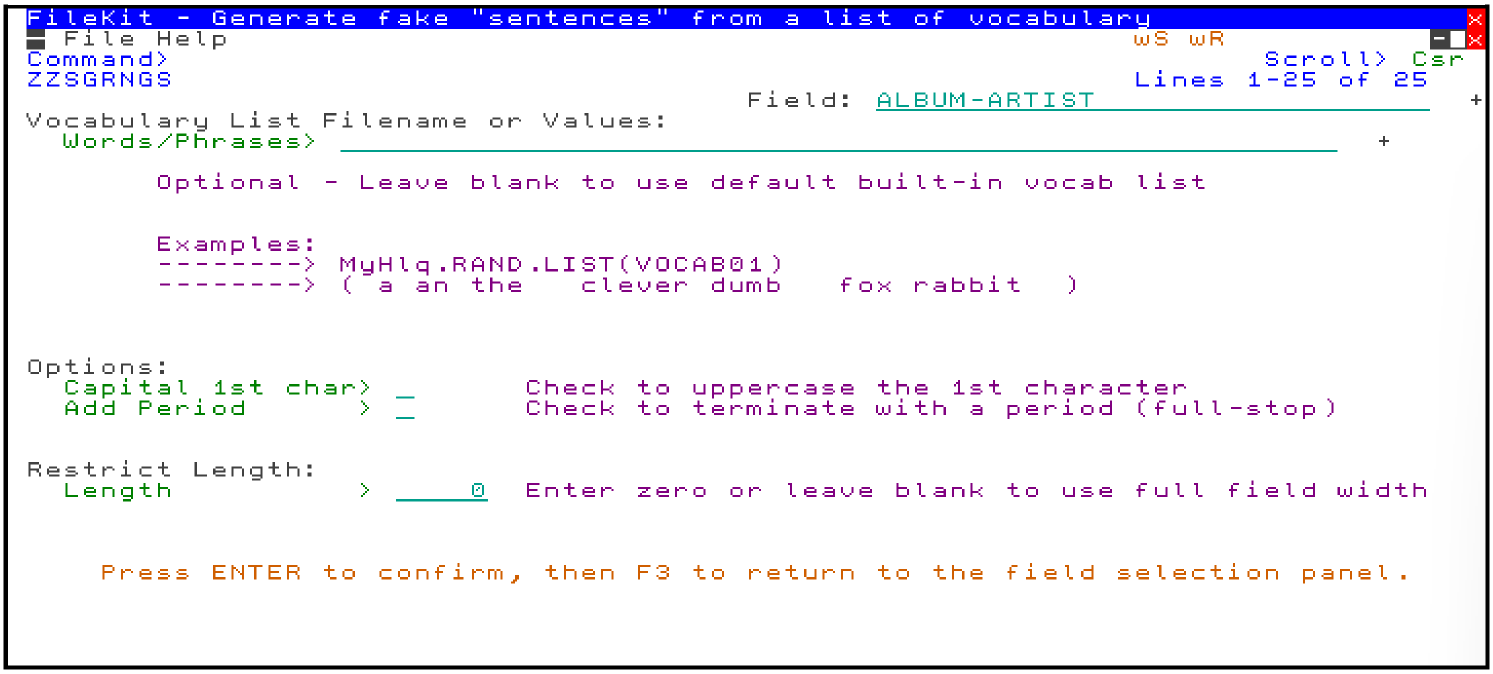


```
FileKit - Specify PATTERN string
 File Help                                                    wS wR
Command>                                                             Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                          Field: ALBUM-ARTIST                  +
Pattern:
   String>                                                                +

           Example: A(J-N)#[-]a#(1001-1999)[-]A(INV,STM,REQ)

           Format Characters:                               (Press HELP for full list)
               A                  --> Upper case Alpha ( A to Z  )
               A(C-F)             --> Upper case Alpha ( C to F  )
               a                  --> Lower case Alpha ( a to z  )
               a(p-r)             --> Lower case Alpha ( p to r  )
               N or #             --> Numeric          ( 0 to 9  )
               N(22-55)           --> Numeric          (22 to 55 )
               A(John,Paul,Ringo) --> List of string alternates
               a("Opt 1" "Opt 2") --> Same as above
               N(123, 4567, 89  ) --> Same as above
               #("    0", "9999"  ) --> Same as above
               [literal string]   --> Any character literal


Sequencing Options:
   Option         > RANDOM  RANDOM, SEQ, L-R (Left-Right) or R-L (Right-Left)

     Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 9. Specify PATTERN string.

Generate fake "sentences" from a list of Vocabulary

Option 8. will open the "Fake Sentences" panel view which is used to generate values comprising a number of blank delimited words obtained from a list.

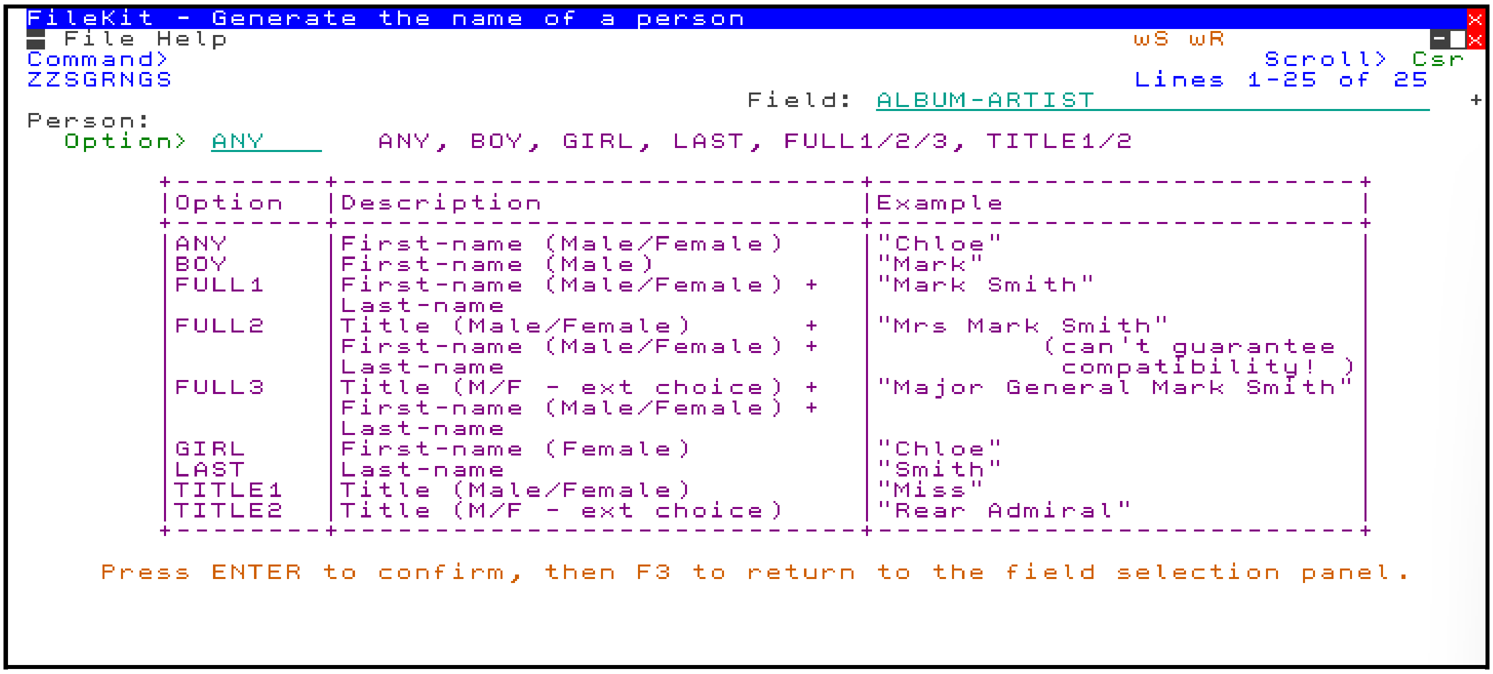


```
FileKit - Generate fake "sentences" from a list of vocabulary
 File Help                                                  wS wR
Command>                                                              Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                   Field: ALBUM-ARTIST                            +
Vocabulary List Filename or Values:
  Words/Phrases> ____________________________________________________ +

      Optional - Leave blank to use default built-in vocab list


      Examples:
      --------> MyHlq.RAND.LIST(VOCAB01)
      --------> ( a an the   clever dumb   fox rabbit   )


Options:
  Capital 1st char> _       Check to uppercase the 1st character
  Add Period      > _       Check to terminate with a period (full-stop)


Restrict Length:
  Length          > ____0   Enter zero or leave blank to use full field width


   Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 10. Generate fake "sentences" from a list of vocabulary.

Generate the name of a Person

Option 9. will open the "Personal Name" panel view which is used to generate a person's name (with or without a surname and/or title) from an internal list of names.



```
FileKit - Generate the name of a person
 File Help                                                  wS wR
Command>                                                              Scroll> Csr
ZZSGRNGS                                                        Lines 1-25 of 25
                                   Field: ALBUM-ARTIST                            +
Person:
  Option> ANY      ANY, BOY, GIRL, LAST, FULL1/2/3, TITLE1/2

      +--------+-----------------------------+--------------------------+
      |Option  |Description                  |Example                   |
      +--------+-----------------------------+--------------------------+
      |ANY     |First-name (Male/Female)     |"Chloe"                   |
      |BOY     |First-name (Male)            |"Mark"                    |
      |FULL1   |First-name (Male/Female) +   |"Mark Smith"              |
      |        |Last-name                    |                          |
      |FULL2   |Title (Male/Female)      +   |"Mrs Mark Smith"          |
      |        |First-name (Male/Female) +   |         (can't guarantee |
      |        |Last-name                    |          compatibility! )|
      |FULL3   |Title (M/F - ext choice) +   |"Major General Mark Smith"|
      |        |First-name (Male/Female) +   |                          |
      |        |Last-name                    |                          |
      |GIRL    |First-name (Female)          |"Chloe"                   |
      |LAST    |Last-name                    |"Smith"                   |
      |TITLE1  |Title (Male/Female)          |"Miss"                    |
      |TITLE2  |Title (M/F - ext choice)     |"Rear Admiral"            |
      +--------+-----------------------------+--------------------------+

   Press ENTER to confirm, then F3 to return to the field selection panel.
```

Figure 11. Generate the name of a person.

Test Data - SET RANDOMIZER

Syntax:

```
         +- TEMPorary -+  +- SET -+
         |             |  |       |
     >>--+-------------+--+-------+--- RANDomizer ---+-- field_name -+------------>
         |             |                             |               |
         +- PERManent -+                             +-- field_ref --+

       (1) +--- | Character Values | ---+
           +---- | Numeric Values | ----+
           +--- | Date/Time Values | ---+
           +----- | Time Values | ------+
           |                            |
      >--+-+----------------------------+--+------------------------+-------+------>
         | |                            |  |                        |       |
         | +----- | List Values | ------+  |         (2)            |       |
         | |                            |  |           +- SEQRL -+ |       |
         | +--- | Key List Values | ----+  |           |         | |       |
         | |                            |  +- CHAIN ---+---------+-+       |
         | +--- | Sentence Values | ----+              |         |         |
         | |                            |              +- SEQLR -+         |
         | +- | Personal Name Values | -+                                  |
         | |                            |                                  |
         | +-- LITeral -- "string" -----+                                  |
         |                                                                 |
         +------- | Pattern Value | ---------------------------------------+
         |                                                                 |
         +-- REPlacement --+-------------------------+---------------------+
                           |                         |
                           +- ( replace_expression ) -+

      >--+--------------+--+------------------+---------------------------------->
         |              |  |                  |
         +- LENGTH len -+  +- STRIPboth -----+
                           +- STRIPLeading --+
                           +- STRIPTrailing -+

      >--+---------------------------------------------------------------+--------->>
         |                                                               |
         +- FOR +--------+- rec_type -+----------------------------------+
                |        |            |                                  |
                + RECord +            + IN -+-----------+- struct_name -+
                                            |           |
                                            + STRUCTure +
```

(1) Default value types matches that of the source field data type.

Numeric Values	Default for any numeric data type field.
Character Values	Default for any character or hex data type field.
Date/Time Values	Default for any DATE or TIMESTAMP data type field.
Time Values	Default for any TIME data type field.

(2) The last CHAIN element to have SEQRL or SEQLR specified will apply.

Character Values:

```
                           (2)
         +---- CHARS ------ "default_chars_list" --+
         | (1)                                     |
         |  +- CHars --+                           |
         |  |          |                           |
      |--+--+----------+-+-------------------------+---- | Index Values | --------|
                         |                         |
                         +- "chars_list" ----------+
                         +- ALPHA -----------------+
                         +- ALPHANumeric ----------+
                         +- HEX -------------------+
                         +- HEXEVEN ---------------+
                         +- NUMeric ---------------+
                         +- LALPHA ----------------+
                         +- LALPHANumeric ---------+
                         +- LHEX ------------------+
                         +- LHEXEVEN --------------+
                         +- MALPHA ----------------+
                         +- MALPHANumeric ---------+
```

(1) CHARS keyword is mandatory for *"chars_list"*

(2) "*default_chars_list*" characters are: "A-Z", "0-9", "+-=,./*()_!:@#$" and the blank character.

Numeric Values:

```
                    (1)      (2)                               (4)
   +- RANGE -------- * -------- * ----------------------------+
   |                                                          |
|--+----------------------------------------------------------+-+---------+-|
   |                                                          | |         |
   +- RANGE ------ lo_num -+----------+-----+---------------+-+ +- ZEROS -+
   |                       |          |     |               | |
   |                       +- hi_num -+     +- BASE string -+ |
   |                     (2)                                  |
   |                                                          |
   |                         (2)                              |
   |            +- 1 ---------- * ------ +1 ----+             |
   |            |                               |             |
   +- SEQuence -+-------------------------------+-------------+
   |            |                               |             |
   |            +- lo_num -+------------------+-+             |
   |                       |                  |               |
   |                   (2) +- hi_num -+-------+               |
   |                                  |       |               |
   |                              (3) +- inc -+               |
   |          (3)                                             |
   +- ADJust -- inc --+-----------+--+------------------+-----+
                      |           |  |                  |
                      +- PERCENT -+  +- (source_field) -+
```

(1) Default low number (*lo_num*) for a RANGE is "*", the minimum value supported for the numeric field.

(2) Default high number (*hi_num*) is "*", the maximum supported for the field. (Character => "9" for length #digits).

(3) Negative integer increment (*inc*) values are supported. Default increment for SEQUENCE is: +1
The low number (*lo_num*) may be **higher** than the high number (*hi_num*), in which case the default increment is -1.

(4) Numbers may be expressed with or without a decimal point and may include an exponent. The scale of the generated value will equal the largest scale supplied on *lo_num*, *hi_num* and *inc* values.

Date/Time Values:

```
(1)                                                           (3)
  +- DATE -+              +- RANGE -- 2001/01/01 00:00:00.00 -- hi_ts -----+
  |        |              |                                                |
|-+--------+-+----------+-+------------------------------------------------+-|
             |          | |                                                |
             + "str_ts" + +--- NOW ----------------------------------------+
           (2)            |                                                |
                          +--- TODay --------------------------------------+
                          |                                                |
                          +-+- PAST ---+-+---------------------------------+
                          | |          | |                                 |
                          | +- FUTure -+ |           +- DAYs --------+     |
                          |              |           |               |     |
                          |              +- int -----+---------------+-----+
                          |                          |               |     |
                          |                          +- HOURs -------+     |
                          |                          +- MINutes -----+     |
                          |                          +- SECs --------+     |
                          |                                                |
                          +- RANGE -- lo_ts -+--------+-+---------------+--+
                          |                  |        | |               |  |
                          |                  + hi_ts -+ +- BASE string -+  |
                          |                (3)                             |
                          |                                   + DAYs ---+  |
                          |                                   |         |  |
                          +- SEQuence lo_ts -+--------------+-+---------+--+
                          |                  |              | |         |  |
                          |              (3) + hi_ts -+-----+ + HOURs --+  |
                          |                           |     | + MINutes +  |
                          |                       (4) + inc + + SECs ---+  |
                          |                                                |
                          |                + DAYs ---+                     |
                          |                |         |                     |
                          +- ADJust - inc -+---------+-+----------------+--+
                                     (4)   |         | |                |
                                           + HOURs --+ + (source_field) +
                                           + MINutes +
                                           + SECs ---+
```

(1) DATE keyword required if source field is **not** a FileKit DATE or TIMESTAMP data type. (e.g. a character field)

(2) Omit "*str_ts*" for a DATE or TIMESTAMP source field.
"*str_ts*" is a quoted date/time string containing any character, but the following are substituted with date/time elements:
"CC", "CI", "YYYY", "YY", "MM", "MMM", "Mmm", "DD", "DDD", "Ddd", "JJJ", "WW", "WWS", "hh", "mm", "ss", "ttt"
Default combination is: **"CCYY/MM/DD hh:mm:ss.ttt"**, or **"CCYYMMDDhhmmssttt"** for numeric source fields.

(3) Default high date/time (*hi_ts*) is: 2042/09/17 23:53:47.37

(4) Negative integer increment (*inc*) values are supported. Default increment for SEQUENCE is: +1 DAY
The low date/time (*lo_ts*) may be **higher** than the high date/time (*hi_ts*), in which case the default increment is -1 DAY.

Time Values:

```
(1)                                                          (3)
  +- TIME -+              +- RANGE ------------- 00:00:00.00 -- hi_ti ----+
  |        |              |                                               |
|-+--------+-+----------+-+-----------------------------------------------+-|
             |          | |                                               |
             + "str_ti" + +--- NOW ---------------------------------------+
           (2)            |                                               |
                          +-+- PAST ---+-+--------------------------------+
                          | |          | |                                |
                          | +- FUTure -+ |           +- SECs --------+    |
                          |              |           |               |    |
                          |              +- int -----+---------------+----+
                          |                          |               |    |
                          |                          |               |    |
                          |                          +- HOURs -------+    |
                          |                          +- MINutes -----+    |
                          |                                               |
                          +- RANGE -- lo_ti -+--------+-+---------------+-+
                          |                  |        | |               | |
                          |                  + hi_ti -+ +- BASE string -+ |
                          |                (3)                            |
                          |                                   + SECs ---+ |
                          |                                   |         | |
                          +- SEQuence lo_ti -+--------------+-+---------+-+
                          |                  |              | |         | |
                          |                  + hi_ti -+-----+ + HOURs --+ |
                          |                (3)        |     | + MINutes + |
                          |                           + inc +             |
                          |                         (4)                   |
                          |                                               |
                          |                + SECs ---+                    |
                          |                |         |                    |
                          +- ADJust - inc -+---------+-+----------------+-+
                                     (4)   |         | |                |
                                           + HOURs --+ + (source_field) +
                                           + MINutes +
```

(1) TIME keyword required if source field is **not** a FileKit TIME data type. (e.g. a character field)

(2) Omit "*str_ti*" for a TIME source field.
"*str_ti*" is a quoted time string containing any character, but the following are substituted with time elements:
"hh", "mm", "ss", "ttt"
Default combination is: **"hh:mm:ss.ttt"**, or **"hhmmssttt"** for numeric source fields.

(3) Default high time (*hi_ti*) is: 23:59:59.99

(4) Negative integer increment (*inc*) values are supported. Default increment for SEQUENCE is: +1 SEC
The low time (*lo_ti*) may be **higher** than the high time (*hi_ti*), in which case the default increment is -1 SEC.

List Values:

```
                  +--------+        +- VALLOCATION (1) ------------------+
                  v        |        |                                    |
|-- LIST --+-- ( -+- item -+- ) --+-+------------------------------------+-->
           |                      | |                                    |
           +------ list_file -----+ +- VALLOCATION ( pos -+--------+- ) -+
                                                          |        |
                                                          +- ,len -+

>---- | Index Values | -----------------------------------------------------|
```

Key List Values:

```
                   +--------+
                   v        |
|- KEYLIST -+-- ( -+- item -+- ) --+--- VALLOCATION ( pos -+--------+- ) --->
            |                      |                       |        |
            +------ list_file -----+                       +- ,len -+

>-- KEYLOCATION ( pos -+--------+- ) --+-------------------------+----------|
                       |        |      |                         |
                       +- ,len -+      +- KEY( key_expression ) -+
```

Sentence Values:

```
  +- LIST -+       +--------+       +- VALLOCATION (1) ------------------+
  |        |       v        |       |                                    |
|-+--------+-+- ( -+- item -+- ) -+-+------------------------------------+-->
             |                    | |                                    |
             +---- list_file -----+ +- VALLOCATION ( pos -+--------+- ) -+
                                                          |        |
                                                          +- ,len -+

>---- | Index Values | ------ VOCab ----+--------+-+----------+-------------|
                                        |        | |          |
                                        +- CAP1 -+ +- PERIOD -+
```

Personal Name Values:

```
                   +----- ANY ---------+
                   |                   |
|-- PERSon --- ( --+-------------------+-- ) ---- | Index Values | ---------|
                   |                   |
                   +----- BOY ---------+
                   +----- FULL --------+
                   +----- FULL1 -------+
                   +----- FULL2 -------+
                   +----- FULL3 -------+
                   +----- GIRL --------+
                   +----- LAST --------+
                   +----- TITLE -------+
                   +----- TITLE2 ------+
```

Pattern Values:

```
|-- PATTERN -- "pattern_string" ---+------------------------+-+---------+---|
                                   |                        | |         |
                                   |            +- SEQRL -+ | +- ECHO --+
                                   |            |         | |
                                   +- SEQuence -+---------+-+
                                   |            |         | |
                                   |            +- SEQLR -+ |
                                   |                        |
                                   +- BASE string ----------+
```

Index Values:

```
                               (2)                               (4)
     +- RANGE -------- 1 -------- * ----------------------------+
(1)  |                                                          |
  |--+----------------------------------------------------------+-------------|
     |                                                          |
     +- RANGE ------ lo_num -+----------+-----+---------------+-+
     |                       |          |     |               | |
     |                       +- hi_num -+     +- BASE string -+ |
     |                     (2)                                  |
     |                                                          |
     |                         (2)                              |
     |            +- 1 ---------- * ------ +1 ----+             |
     |            |                               |             |
     +- SEQuence -+-------------------------------+-------------+
                  |                               |
                  +- lo_num -+------------------+-+
                             |                  |
                         (2) +- hi_num -+-------+
                                        |       |
                                    (3) +- inc -+
```

(1) An **Index Value** is a numeric index into a string of characters or list items.

(2) Default high number (*hi_num*) is "*", the number of random characters, or the number of items in the random value list.

(3) Negative integer increment (*inc*) values are supported. Default increment is: +1
The low number (*lo_num*) may be **higher** than the high number (*hi_num*), in which case the default increment is -1.

(4) Numbers *lo_num*, *hi_num* and *inc* are integer values.

Description:

Use the SET RANDOMIZER command to define test data generation options for any individual field in your record layout structure. The randomizer definition may be temporary, or it may be saved to the FileKit SDO structure permanently.

Once defined to a field, the randomizer will generate values for that field when any of the following occur.

- Command **INSERT** or **REPLACELINE** is executed when using the FileKit Data-Editor to edit data fields for which a randomizer is defined .
- The FSU utility is used to **copy** or **update** record field values for which a randomizer is defined.
- The FILEIO utility (typically executed under control of a REXX procedure) is used **write** or **update** record field values for which a randomizer is defined.

Examples:

Generate **random** numbers in a range.

```
set randomizer TRACK-NUM  range 1 30
```

Generate a **sequence** of numbers based on a supplied increment/decrement value.

```
rand SALARY seq 500   600.30   +0.05

      /* Will produce "500.00",    on 1st record
      |                "500.05",    on 2nd record
      |                "500.10" etc
      |         up to "600.30"     ... after which the sequence will repeat.
      */
```

Generate **date/time** fields either randomly or in sequence.

```
rand START-DATE date "CCYY-MM-DD (DDD)"       range 1980/01/01 2030/12/31
      /* e.g. "2002-10-27 (SUN)" */

rand LASTUPDATE date "Mmm DD CCYY hh:mm:ss"
                                 seq  "2003/04/10 08:30" 2003/04/20 +5 SECS

      /* i.e. "Apr 10 2003 08:30:00"
      |       "Apr 10 2003 08:30:05"
      |       "Apr 10 2003 08:30:10"
      |       "Apr 10 2003 08:30:15"
      |        etc
      */
```

Adjust existing numeric or date/time values using a supplied increment/decrement value.

```
rand NAME-POS    adjust +8
rand ORDER-DATE  date "CCYY-MM-DD (DDD)"     adjust -30 days
rand SALARY      adjust +3.75 percent
```

Perform a **calculation** based on current value(s) in this and/or other field(s).

```
rand BONUS       replacement(BONUS*2)      /* Double it! */
rand GAS-RATE    rep(  (GAS-COST-0.2848) / GAS-KWHs  )
```

Pick from a **list** of values, either randomly or in sequence. Any list may be supplied as a **separate file** or from **in-line** values.

```
rand CALLER          list=MY.RAND.LIST(MODULES)   /* Get list from a file */
rand FIRST-NAME  seq list( "George", "Paul", "Ringo", "John" )
```

Substitute values by performing a **keyed lookup** into a supplied list.

```
rand FIRST-NAME  KeyLocation(1,10) ValLocation(11,10)
              /*  keyList MY.RAND.LIST(FIRSTNAME)  */
                 keyList(
                          "Annabel   Alison    "
                          "Edward    David     "
                          "Heidi     Etta      "
                          "Jack      James     "
                          "Laurence  John      "
                          "Paul      Nicholas  "
                          "Peter     Paul      "
                          "Pasqual   Peter     "
                          "Simon     Ricky     "
                        )
```

Fill character fields using a supplied list of **"vocabulary"**. The selected words will be blank separated. Optionally the first character of the first word may be uppercased, and a period (full-stop) added to the end.

```
rand DESC-TXT  vocab    cap1  period
             /*     List MY.RAND.LIST(WORDLIST)  */
                    List(
                          "a"     "an"    "the"   "and"
                          "have"  "that"  "for"   "you"
                          "with"  "say"   "this"  "they"
                          "but"   "his"   "from"  "not"
                          "ask"   "need"  "too"   "feel"
                          "three" "state" "never" "become"
                          "night" "high"  "real"  "each"
                          "most"  "other" "much"  "family"

                          "a"     "an"    "the"   "and"
                          "a"     "an"    "the"   "and"

                          "A complete sentance?"
                         )
```

Generate data according to a supplied **"pattern"** string.

```
rand PART-ID   pattern "A(J-N)#[-]#(1001-1999)[-]A(JGE,DJG,NBJ)"

      /* Example output "K5-1758-NBJ"
      |                 "J8-1044-JGE"
      |                 "M1-1346-DJG"
      */
```

To go beyond any limitations of the built-in "pattern" string syntax, multiple components may be manually **"chained"** together.

```
rand PART-ID            chars "JND"     len=1
rand PART-ID  chain     range 1,3       len=1
rand PART-ID  chain     lit   "-"
rand PART-ID  chain     range 901,999   len=3 zeros sequence
rand PART-ID  chain     lit   "-"
rand PART-ID  chain     range 1001,1999 len=4
rand PART-ID  chain     lit   "-"
rand PART-ID  chain     list ( "JGE", "DJG", "NBJ", "NGH", "CLS" )

      /* Example output "D3-901-1758-CLS"
      |                 "J1-902-1044-NBJ"
      |                 "N2-903-1346-DJG"
      */
```

Test Data - FSU (File Search & Update Utility)

Syntax:

```
>>-- FSU --+---------------------------------------------------------+------><
           |                                                         |
           +- | UnFormatted Data Options | -+-- | Common Options | --+
           |                                |
           +-- | Formatted Data Options | --+
```

Formatted Data Options:

```
|-- ... Other options ----+-------------------------------------+------------|
                          |                                     |
                          +- | Test Data Generation Options | -+
```

Test Data Generation Options:

```
  +-----------------------------------------------------------------------+
  v                                                                       |
|-+- RANDomize ( -+ field_name +-+-------------+- | Randomize Opts | - ) -+-|
                  |            | |             |
                  + field_ref -+ + FOR rectype +
```

Randomize Options:

```
   (1) +--- | Character Values | ---+
       |                            |
       +---- | Numeric Values | ----+
       |                            |
       +--- | Date/Time Values | ---+
       |                            |
       +----- | Time Values | ------+
       |                            |
|--+-+----------------------------+--+-----------------------+-------+------>
   | |                            |  |                       |       |
   | +----- | List Values | ------+  |         (2)           |       |
   | |                            |  |           +- SEQRL -+ |       |
   | +--- | Key List Values | ----+  |           |         | |       |
   | |                            |  +- CHAIN ---+---------+-+       |
   | +--- | Sentence Values | ----+              |         |         |
   | |                            |              +- SEQLR -+         |
   | +- | Personal Name Values | -+                                  |
   | |                            |                                  |
   | +-- LITeral -- "string" -----+                                  |
   |                                                                 |
   +------- | Pattern Value | ---------------------------------------+
   |                                                                 |
   +-- REPlacement --+--------------------------+--------------------+
                     |                          |
                     +- ( replace_expression ) -+

>--+----------+--+-----------------+---------------------------------------|
   |          |  |                 |
   +- LENGTH -+  +- STRIPboth -----+
                 +- STRIPLeading --+
                 +- STRIPTrailing -+
```

(1) Default value types matches that of the source field data type.
(2) The last CHAIN element to have SEQRL or SEQLR specified will apply.

Description:

The FSU command, used to execute the File Search & Update utility, has been updated to support definition of randomizers for formatted output fields. i.e. for formatted input record update, and for formatted record copy to another file (with or without target record field remapping).

The randomizer definition syntax matches that described for SET RANDOMIZER.

Test Data - FILEIO (File I/O Utility)

Syntax:

```
>>--+- FILEIO -+-- filename ---+--- | Open/Close Operations | -----+--------><
    |          |               |                                  |
    +- FIO ----+               +--- | Input Operations | ----------+
                               |                                  |
                               +--- | Output Operations | ---------+
                               |                                  |
                               +--- | Selection Operations | ------+
                               |                                  |
                               +--- | Test Data Options | ---------+
                               |                                  |
                               +--- | Miscellaneous Operations | --+
                               |                                  |
                               +--- | Options | -------------------+
```

Test Data Options:

```
  +-------------------------------------------------------------------------+
  v                                                                         |
|-+- RANDomize ( -+ field_name +-+-------------+- | Randomize Opts | - ) -+-|
                  |            | |             |
                  + field_ref -+ + FOR rectype +
```

Description:

The FILEIO command, used to execute the File I/O utility, has been updated to support definition of randomizers for formatted output fields.

The Randomize options syntax matches that described for FSU and SET RANDOMIZER.

FileKit REPORT Utility

The FileKit REPORT utility may be used to produce attractive printed reports and/or CSV, XML and JSON format output from structured data.

The input data may be read from MVS data sets, HFS/ZFS file systems or from a DB2 result table. In particular, the REPORT utility can accurately processes and generate its output from SMF records. The following is an example of REPORT utlity report definition control statements and the generated SMF printed report output.

```
COLUMNS:
  SMF119#02_TCP_Connection_Termination.zRName
  SMF119#02_TCP_Connection_Termination.zConnectStart
  SMF119#02_TCP_Connection_Termination.zConnectEnd
  SMF119#02_TCP_Connection_Termination.zInBytes
  SMF119#02_TCP_Connection_Termination.zOutBytes
  SMF119#02_TCP_Connection_Termination.zTermCode
```

```
12024/03/21 17:22                                                                                      PAGE    1

 TCP
 socket                                                                                           Connection
 resource Connection start date  Connection end date &                                            Termination
 name     & time                 time                   Inbound byte count   Outbound byte count  reason
 :zRNAME  :zCONNECTSTART         :zCONNECTEND           :zINBYTES            :zOUTBYTES           :zTERMCODE
 -------- ---------------------- ---------------------- -------------------- -------------------- --------------
 TN3270   2024/03/07 08:40:37.60 2024/03/07 08:40:39.99                   31                 1439 RESET_Received
 JGE      2024/03/07 09:02:35.91 2024/03/07 09:02:36.14                21565                    0 App_Close
 FTPD1    2024/03/07 09:02:35.11 2024/03/07 09:02:36.37                  125                  485 App_Close
 RXSERVE  2024/03/07 09:02:36.56 2024/03/07 09:02:41.86                  145                   42 App_Close
 RXSERVE  2024/03/07 09:02:37.26 2024/03/07 09:02:41.97                    0                    0 App_Close
 JGE      2024/03/07 09:04:07.80 2024/03/07 09:04:07.98                21594                    0 App_Close
 FTPD1    2024/03/07 09:04:07.26 2024/03/07 09:04:08.18                  123                  483 App_Close
 RXSERVE  2024/03/07 09:04:08.40 2024/03/07 09:04:11.16                  143                   40 App_Close
 RXSERVE  2024/03/07 09:04:08.53 2024/03/07 09:04:11.28                    0                    0 App_Close
 TN3270   2024/03/07 10:19:04.72 2024/03/07 10:32:39.26                   57                 1439 RESET_Received
 TN3270   2024/03/07 08:40:39.99 2024/03/07 13:28:14.42                 7444               952807 No_FIN
 TN3270   2024/03/07 10:32:39.27 2024/03/07 13:54:56.90                25214              4368142 RESET_Received
 JGE      2024/03/07 15:23:29.06 2024/03/07 15:23:29.34                38614                    0 App_Close
 FTPD1    2024/03/07 15:23:28.16 2024/03/07 15:23:29.54                  122                  483 App_Close
 RXSERVE  2024/03/07 15:23:29.77 2024/03/07 15:23:56.37                  143                   40 App_Close
 RXSERVE  2024/03/07 15:23:30.36 2024/03/07 15:23:56.49                    0                    0 App_Close
 JGE      2024/03/07 16:22:16.25 2024/03/07 16:22:16.47                    0                 2598 App_Close
 FTPD1    2024/03/07 16:22:15.49 2024/03/07 16:22:16.49                  120                  465 App_Close
 JGE      2024/03/07 16:22:17.98 2024/03/07 16:22:18.17                    0                13368 App_Close
 FTPD1    2024/03/07 16:22:17.55 2024/03/07 16:22:18.18                  117                  474 App_Close
                                                        ==================== ====================
 == Grand Totals (20 Items)                                           115557              5342305
                                                        ==================== ====================
```

The following is a summary of the significant new REPORT Utility features provided in FileKit release 3.60. A full description of the REPORT utility including working samples, panels and report definition control statement syntax may be found in the CBL publication, "*FileKit REPORT Utility*".

REPORT Execution Speed

Significant improvement has been made to the performance of REPORT utility execution.

Conversion of the REPORT utility source code to High Level Assembler has meant a large reduction in REPORT execution times and CPU usage with many simple jobs running up to 10 times faster than previously achieved.

The original (REXX based) version of the REPORT utility command has been renamed as "REPORX" and may still be used if necessary. Note, however, that this version will no longer receive maintenance or include new product enhancements.

REPORT Definition INPUT/OUTPUT Sections

Sections INPUT: and OUTPUT: introduced to define default input source and output data set destination respectively. Entries specified in these sections may be overridden using REPORT command parameters or values entered in REPORT utility panel input fields.

Examples of INPUT: and OUTPUT: sections specifications follow.

```
INPUT:          'CBL.INST.CBL21042.SZZSSAM2(ZZSDF1DR)' \
    USING COBOL 'CBL.INST.CBL21042.SZZSSAM1(ZZSCF1DR)'

OUTPUT:         'CBL.F1DRIVER.REPORT.DS2024253'


INPUT:      DD=SMFCAT
OUTPUT:     DD=SMFXML


INPUT:
    DB2(CBLA) CBL.ZZSFUNC                \
              FROM ROW 31   FOR 50 ROWS  \
         WHERE( FUNCNAME LIKE 'B%')      \
      ORDER BY( FUNCNAME DESC )

OUTPUT:         "/XS01/home/john/Functions.csv"
```

REPORT Definition MAP Section

The MAP: section has been updated to support field definitions specified using FileKit CREATE STRUCTURE command syntax. This provides a more comprehensive alternative to the SYMNAMES format currently supported.

Example MAP: section using CREATE STRUCTURE syntax follows.

```
MAP:
   RefID        int(2)  unsigned  remark "Unique Reference Number"
  ,            char(17)           remark "Filler"
  ,Host        char(12)           remark "Host Name"
  ,            char(6)            remark "Filler"
  ,UserName  struct(
                    LastName   char(30)
                   ,FirstName  char(30)
                  )
  ,            char(32)           remark "Filler"
  ,IPv4          ip(4)            remark "Host IP Address"
```

REPORT Definition INIT-EXIT Section

The INIT-EXIT: section has been introduced to allow initialisation of *compute-field* values when input is via FILEIO (default).

A *compute-field* is one that corresponds to a REXX variable of the same name, and which may be updated in the REXX statements of the COMPUTE: section.

The sections INIT-EXIT: and BROWSE-EXIT: are similar in that they are both executed prior to processing the first input record. However, presence of a BROWSE-EXIT section will trigger use of REPORT's Data-Edit BROWSE input processing.

Example use of an INIT-EXIT: section with a COMPUTE: section follows.

```
INIT-EXIT:
  JobArray = ''                 /* Initialise Compute-field variables. */

COMPUTE:
  if wordpos(zJobName,JobArray) = 0   /* First occurrence of this Job Name ? */
    then  JobArray = JobArray zJobName      /* Add it to array of Job Names. */
```

REPORT Input Record Field Value Reset

By default, the RESET: section will reset all input field values to null after processing a record or record segment of the specified record type mapping. The reset of an individual input field's value may now be suppressed in different circumstances by including one of the following on the column/field specification in the COLUMNS: or REQUIRED: section.

NORESET
Suppresses reset of the field value that would occur as a result of a RESET: section specification.

NORESETBREAK
Suppresses reset of the field value that would occur as a result of a RESET: section specification, only if the field value is to be displayed in the first line of a control group. Only applicable to COLUMNS: section.

NORESETPAGE
Suppresses reset of the field value that would occur as a result of a RESET: section specification, only if the field value is to be displayed in the first line of a new page. Only applicable to COLUMNS: section.

REPORT OPTIONS for SMF Filtering

Options introduced specifically for SMF record processing, These options correspond with SMF record filtering options that may be specified as parameters to the REPORT utility execution (either via a panel input field or the on the REPORT command).

Note that, these report definition options are overridden by their corresponding parameters if specified on the REPORT utility execution.

SMFDATEHI(*timestamp* | *-days*)
Specifies the latest SMF record timestamp to be processed.

SMFDATELO(*timestamp* | *-days*)
Specifies the earliest SMF record timestamp to be processed.

SMFJOBNAME(*jobname* [, ...])
Specifies jobname masks for SMF record content match criteria processing.

SMFLOGIC(OR|AND)
Specifies the logical operation performed between SMF record content match criteria.

SMFONLINE(YES|NO)
Specifies whether SMF records are processed directly from an online SMF log data set or from SMF archive data sets.

SMFSID(*sid* [, ...])
Specifies system identification masks for SMF record content match criteria processing.

SMFTYPES(*rectype* | *rectype*:*rectype* | {*rectype-subtype* | *rectype#subtype*} [, ...])
Specifies record type/subtype masks for SMF record content match criteria processing.

SMFUSER(*username* [, ...])
Specifies user name masks for SMF record content match criteria processing.

REPORT OPTIONS for CSV Output

The following options have been introduced to manage the appearance of values in CSV output.

CSVLITERALS(YES|NO)
Determines whether or not *literal* values specified in the COLUMNS: section are included as values in the CSV output.

CSVQUOTED(YES|NO)
Determines whether values are **always** enclosed in quotation marks ("), or are enclosed in quotation marks only when necessary. e.g. values containing commas (",").

CSVSTRIPALL(YES|NO)
Determines whether or not leading and trailing blanks are stripped from the values so that the comma separator immediately follows the last non-blank character on all but the last value in the output line. If blanks are not stripped, then the value will be of a fixed length equal to the specified (or default) field width.

```
Album,Track#,Track Name,Artist,Time (1/1000 sec)
Feel,11,Flow,Roachford,281054
Feel,10,Time,Roachford,452417
Feel,9,Testify,Roachford,217292
Feel,8,Down,Roachford,244645
Feel,7,Move On,Roachford,267818
Feel,6,Nothing Free,Roachford,269026
Feel,5,Naked Without You,Roachford,209536
Feel,4,Someday,Roachford,210326
Feel,3,Don't Make Me Love You,Roachford,232710
Feel,2,How Could I? (Insecurity),Roachford,224769
Feel,1,Way I Feel,Roachford,226023
Feels Like Today,12,Skin (Sarabeth),Rascal Flatts,261280
Feels Like Today,11,Oklahoma-Texas Line,Rascal Flatts,175013
Feels Like Today,10,Holes,Rascal Flatts,258933
Feels Like Today,9,Break Away,Rascal Flatts,191680
Feels Like Today,8,The Day Before You,Rascal Flatts,246440
Feels Like Today,7,Here's to You,Rascal Flatts,217840
Feels Like Today,6,When the Sand Runs Out,Rascal Flatts,226506
Feels Like Today,5,Fast Cars and Freedom,Rascal Flatts,263053
Feels Like Today,4,Feels Like Today,Rascal Flatts,201640
Feels Like Today,3,Then I Did,Rascal Flatts,192586
Feels Like Today,2,Bless the Broken Road,Rascal Flatts,227186
Feels Like Today,1,Where You Are,Rascal Flatts,232933
```

REPORT OPTIONS for JSON Output

The following options have been introduced to manage the appearance of values in JSON output.

JSONARRAY(YES|NO)
Determines whether field values for each report line are part of a single JSON object, or one object within an array of objects.

JSONINDENT(YES|NO)
Determines whether the key/value pairs for each report field are to occur on the same report output line, or are to be written to a new line of the report output and indented beneath the opening and closing JSON object string braces ("{}").

JSONLITERALS(YES|NO)
Determines whether or not *literal* values specified in the COLUMNS: section are included as the "string" value in a key/value pair of the JSON output.

JSONQUOTED(YES|NO)
Determines whether or not values are **always** treated as JSON strings and enclosed in quotation marks ("), or are treated as strings for non-numeric field values only.

JSONSTRIPALL(YES|NO)
Determines whether or not leading and trailing blanks are stripped from the values. This is particularly relevant to quoted JSON string values where leading trailing blanks would be treated as part of the string value. If blanks are not stripped, then the value will be of a fixed length equal to the specified (or default) field width.

```
{"FileKit_Report" :
 [
   ,{"Album" : "Feel",
     "Track#" : 3,
     "Track Name" : "Don't Make Me Love You"
    }
   ,{"Album" : "Feel",
     "Track#" : 2,
     "Track Name" : "How Could I? (Insecurity)"
    }
   ,{"Album" : "Feel",
     "Track#" : 1,
     "Track Name" : "Way I Feel"
    }
   ,{"Album" : "Feels Like Today",
     "Track#" : 3,
     "Track Name" : "Then I Did"
    }
   ,{"Album" : "Feels Like Today",
     "Track#" : 2,
     "Track Name" : "Bless the Broken Road"
    }
   ,{"Album" : "Feels Like Today",
     "Track#" : 1,
     "Track Name" : "Where You Are"
    }
 ]
}
```

REPORT OPTIONS for XML Output

The following options have been introduced to manage the appearance of values in XML output.

XMLINDENT(YES|NO)
Determines whether the XML tagged report field values are to occur on the same report output line, or are to be each written to a new line of the report output and indented within the report line tags.

XMLLITERALS(YES|NO)
Determines whether or not *literal* values specified in the COLUMNS: section are included as values in the XML output.

XMLSTRIPALL(YES|NO)
Determines whether or not leading and trailing blanks are stripped from the XML values. If blanks are not stripped, then the value will be of a fixed length equal to the specified (or default) field width.

```
<?xml version="1.0"?>
<INPUT>
 <FileKit_Report>
   <Track_>3</Track_>
   <Track_Name>Don&apos;t Make Me Love You</Track_Name>
 </FileKit_Report>
 <FileKit_Report>
   <Track_>2</Track_>
   <Track_Name>How Could I? (Insecurity)</Track_Name>
 </FileKit_Report>
 <FileKit_Report>
   <Track_>1</Track_>
   <Track_Name>Way I Feel</Track_Name>
 </FileKit_Report>
 <FileKit_Report>
   <Track_>3</Track_>
   <Track_Name>Then I Did</Track_Name>
 </FileKit_Report>
 <FileKit_Report>
   <Track_>2</Track_>
   <Track_Name>Bless the Broken Road</Track_Name>
 </FileKit_Report>
 <FileKit_Report>
   <Track_>1</Track_>
   <Track_Name>Where You Are</Track_Name>
 </FileKit_Report>
</INPUT>
```

REPORT OPTIONS for General Uasage

Support has been included for the following OPTIONS: section options:

DB2NULL(YES|NO)
Determines whether or not the default Data-Edit NULL value output indicator character is displayed for a null value in a DB2 column defined with NULL. (See the NULLCHAR Data-Edit SET/QUERY/EXTRACT option). If DB2NULL(NO), the output value for a DB2 NULL value is blank.

DETAIL(*nlines*[,ALL|DISPLAY])
ALL or DISPLAY options indicate whether generated statistics values are derived from all detail lines in control group or only those displayed. The DETAIL option is now also obeyed for CSV, JSON and XML report output.

FIELDNAME([SHORT] | [LONG])
For input fields, forces REPORT to assign field values to the unqualified (SHORT) format of the field name variable, the qualified (LONG) format of the field name variable, or both.

FIND(*string* [, ...])
For non-DB2 type input, specifies search strings to be used for record selection. For SMF type input, FIND is one of the SMF record content match criteria options.

HEADWIDTH(*int*)
Specifies the width of header and footer lines within a printed report.

ILIM(*int*)
Specifies the input limit, maximum number of records or DB2 rows to be read from the data source.

LINESTRIP(YES|NO)
Determines whether or not trailing blank characters are to be stripped from the lines of text written to the REPORT output.

NUMBLANK(INCLUDE|EXCLUDE)
Determines whether numeric field values, displayed as blanks in the report PRINT output as a result of a BLANKIFEQUAL specification, are to be included in or excluded from column statistics calculations.

NUMDUP(INCLUDE|EXCLUDE)
Determines whether numeric field values, displayed as duplicates of the same column values on the previous detail line, are to be included in or excluded from column statistics calculations. Duplicate column values may occur when column values are not reset following output of a report detail line.

NUMTRUNC(YES|NO[,*char*][,INCLUDE|EXCLUDE])
Specifies whether truncation of displayed numeric values will occur when the value is too large for the designated display area width. Truncation will abbreviate a numeric value using a muliplier suffix ("K", "M", "G", "T" or "P") and an inequality symbol prefix ("<" or ">") if necessary.

If no truncation is to occur, then the display area for numbers that are too large will be filled with the truncated number filler character (*char*). INCLUDE/EXCLUDE determines whether these values are included in column statistics calculations.

OLIM(*int*)
Specifies the output limit, maximum number of detail report lines that may be written to the output report.

PAGEDEPTH(*int*)
Specifies the number of lines written to each page of the printed report output.

PAGEPAD(YES|NO|AUTO)
Determines whether or not blank lines are to be written to the last page of a PRINT output report to pad the page to the specified (or defaulted) PAGEDEPTH. The AUTO operand will pad the last page with blank lines only if it is not the first (and therefore only) page of the report. No page padding occurs if a page footing (report definition FOOT: section) exists.

REXXCOMPOUND(YES|NO)
Determines whether or not the REXX variable names, defined for input-fields identified using a qualified field name, inherit the dot/period (".") field name qualifier separator character and so define a REXX compound symbol variable name. REXXCOMPOUND(NO) will instead use an underscore ("_") in place of a dot/period character in the variable name.

REPORT(DB2|SDE|SMF)
Specifies the type of data in the input source.

SHORTSTATS(YES|NO)
Specifies whether statistics values that overflow the display area width will be shortened to a value with a multiplier suffix and possible "greater than" (">") prefix, or will be substituted with repeating truncation filler characters as defined by option NUMTRUNC.

REPORT COLUMN Equal Values

Applicable to PRINT output only, support introduced for parameter BLANKIFEQUAL (synonym BIEQual, BLANKWHENEQUAL, BWEQual) on individual column/field entries in the COLUMNS: section of the report definition. If specified, then if the column value in the previous report detail line matches the column value in the current report detail line, the the value in the current line will be replaced with blanks.

Option BLANKIFEQUAL(YES/NO) has also been introduced to imply BLANKIFEQUAL on **all** column entries.

REPORT Built-In Functions

The following built-in REXX functions have been introduced for use in the COMPUTE: section:

BYPASS()
Used to skip reporting on the record or record segment currently being processed.

DATEINC()
Used to increment (or decrement) a date value by a number of days, months or years.

EOF()
Used to force end of input to skip reporting on the current record or record segment, and all records that follow.

MONTHBEG()
Used to return the ISO format date for the first day in the month of the current or specified date.

MONTHEND()
Used to return the ISO format date for the last day in the month of the current or specified date.

TIMEINC()
Increment (or decrement) a time or timestamp (date & time) value by a number of hours, minutes or seconds.

REPORT Field Values

Fields identified as a SORT key, BREAK key or a variable value in a *print-expression* no longer inherit the formatted display value defined by a field entry of the same name in the COLUMNS: or REQUIRED: section. The format and display of key field values and *print-expression* field elemments may now be controlled independently.

One or more of the following formatting operands may be specified on an entry in the COLUMNS: or REQUIRED: section, on the key field of a BREAK: or SORT: section, and on a *print-expression* field element:

SUBSTR
Specifies the start character number and optionally the length of the field data to be used as the field value. This occurs before any STRIP operation takes place.

STRIP
Specifies that leading and trailing blanks are to be stripped from the field value. This occurs before field alignment takes place.

width:
The width of the field value text. The field value will be aligned within this field width and blank padded or truncated accordingly. If not specified, the defined maximum width of the built-in or input field is used. Otherwise, for compute fields a default width of 9 is used.

CENTER | **CENTRE** | **LEFT** | **RIGHT**
The alignment of the field value within the field width text. Alignment will occur after any STRIP keyword operation has taken place.

FileKit Structures

FileKit utilities perform functions on data which comprises multiple individual fields of potentially different data type formats. These data records (or DB2 table rows) are usually mapped by structures written for different programming or utility languages (e.g. COBOL, PL1, Assembler and DFSORT).

To process structured data, FileKit uses its own structure definition objects (SDO). An SDO may be created using the Structured Data **CREATE STRUCTURE** utility command. Alternatively, a temporary SDO may be created automatically by FileKit when a COBOL, PL1, HLASM or DFSORT programming language structure is passed to a FileKit utility as the data mapping source.

The CREATE STRUCTURE utility can create an SDO from one or more existing COBOL, PL1, HLASM or DFSORT programming language record mapping structures. It also supports creating an SDO without a pre-existing mapping source. Each SDO may be defined with multiple record structure layouts each containing any number of field definitions. Unless TEMPORARY is specified, CREATE STRUCTURE will save the SDO as the specified SDO name.

FileKit 3.60 includes a number of enhancements to FileKit SDO structures.

SDO Library Search Path

FileKit 3.60 introduces support for an Structure Definition Object (SDO) structure library search path.

SDO structures may be saved as library members for subsequent use by any FileKit utility (e.g. Data Edit, BROWSE, REPORT, FSU, etc.) to apply required record mapping. Consequently, a number of libraries may exist, each containing SDO members.

If an SDO search path has been defined, then an SDO specification in a utility primary command may simply be a member name reference. FileKit will select the first matching mamber name in the SDO library search path. For example, the following will search for member CONTMAP in the SDO search library path.

```
BROWSE  NBJ.MYCONT.DATA   USING CONTMAP
```

To define an SDO library search path, the FileKit INI option "SDE.SDOPATH" must be set. SDO search libraries that are common to all users should be assigned to SDOPATH in the System INI file. Thereafter, %SYSTEM.SDE.SDOPATH% may be prefixed by one or more user specific SDO libraries on an SDOPATH specification in an individual's User INI file.

The following example demonstrates how a common SDO library concatenation of 2 libraries can be extended to include 2 additional, user specific SDO libraries for an individual user "NBJ". Note that "SITESDO", "DISTSDO" and "USERSDO" are arbitrary (yet meaningful) names.

System FileKit INI file contents:

```
(SDE)
 SITESDO=OEM.SELCI.SITE.SDO                            * Local   SDO Library.
 DISTSDO=OEM.SELCI.SZZSDIST.SDO                        * Product SDO Library.
 SDOPATH=%SYSTEM.SDE.SITESDO%; %SYSTEM.SDE.DISTSDO%    * Universal SDO search path.
```

NBJ's User FileKit INI file contents:

```
(SDE)
 USERSDO=NBJ.SELCI.SDO; NBJ.SELCI.SMF.SDO              * 2 x User SDO Libraries.
 SDOPATH=%USER.SDE.USERSDO%; %SYSTEM.SDE.SDOPATH%      * User specific SDO search path.
```

The SDO library search path order will be:

1. `NBJ.SELCI.SDO`
2. `NBJ.SELCI.SMF.SDO`
3. `OEM.SELCI.SITE.SDO`
4. `OEM.SELCI.SZZSDIST.SDO`

SDO DDName

FileKit 3.60 introduces support for a reference to an SDO as an allocated DDName.

An SDO reference may now be specified as a DDName on any FileKit utility primary command. If the specified SDO name is less than or equal to 8 characters in length and is **not** a match for a member name in the SDO library search path, then it will be treated as a possible allocated DDName.

Field Data Types

Using the CREATE STRUCTURE direct field definition syntax, SDO structures may be created with field mappings of different data types. The data type of a field determines how the field data is interpreted and displayed.

FileKit 3.60 introduces support for new data types and includes updates to some existing data types as follows:

Data Type	Description
`DECIMALUnsigned` (*`precision`*`,`*`scale`*)	Defines a field to be interpreted as a packed-decimal number without a sign nibble, allowing *precision* number of digits in total, of which *scale* number of digits will be displayed following the decimal-point. "DECUnsigned" (minimum abbreviation "DECU") is a synonym. Default for *precision* is 7. Default for *scale* is 0. For example, 123456 input to a `DECU(6)` field is packed as X'123456' (3-bytes). Note that DECIMALUNSIGNED differs from data type DECIMAL with the UNSIGNED option which preserves the sign nibble as X'F'. This is the format of a COBOL field defined as PIC 9(6) PACKED-DECIMAL. For example 123456 input to a `DEC(6) UNSIGNED` field is packed as X'0123456F' (4-bytes).
`TIME(STCKE)`	Defines an elapsed time field with the source data interpreted as STCKE format of length 16 bytes. This is a 128-bit unsigned binary value elapsed time value in the same format as that returned by an STCKE operation (i.e. an 8-bit Epoch Index, followed by 104-bit TOD clock and 16-bit Programmable Field).
`TIMESTAMP (STCKE)`	Defines a timestamp (date and time) field with the source data interpreted as STCKE format of length 16 bytes. This is a 128-bit unsigned binary value elapsed time value in the same format as that returned by an STCKE operation (i.e. an 8-bit Epoch Index, followed by 104-bit TOD clock and 16-bit Programmable Field).

FILTER/SQL File Input

A FILTER or SQL clause may be passed to Data-Edit EDIT/BROWSE and other FileKit utilities as a fileid reference, in which case the specified file contains the FILTER clause or SQL query text.

The rules governing how the text is interpreted has been updated to reflect the record format of the file. This change has been introduced specifically to allow specification of very long quoted strings in the text of the FILTER clause or SQL query.

For **RECFM=V**, a blank is inserted between the last non-blank character of a record and the first (blank or non-blank) character of the record that follows. A long quoted string literal must exist on a single line if no blanks are to be inserted.

For **RECFM=F**, **no blank** will be inserted between the last (blank or non-blank) character of a record and the first (blank or non-blank) character of the record that follows. The text of a long quoted string literal may wrap onto the following record without blanks being inserted.

Furthermore, for SQL query input only, lines beginning with "--" are treated as comment lines for compatibility with the IBM SPUFI utility.

Record-Type Assignment

when an SDO structure is used to map input records, each record will be assigned one of the record-types defined in the SDO object.

If the record data does not satisfy any of the "USE *record-type* WHEN" criteria and none of the SDO record-types is defined as being the default, then FileKit previously assigned the first record-type defined in the SDO to the record.

This default may cause unexpected results when using FileKit utilities and so this default has been changed in FileKit 3.60 so that the generic record-type "UnMapped" is used instead. The "UnMapped" record-type comprises a single character field of length equal to the record length.

FileKit Data Editor

The FileKit Data Editor provides facilities to perform browse and edit of data mapped by a structure, VSAM data sets, DB2 table rows and also data sets that are too large to be loaded entirely into available storage.

FileKit 3.60 includes a number of enhancements to the FileKit data editor.

Data Editor Line Commands

The following Data Editor line commands have been included in FileKit 3.60. These set the colour of the text displayed in all records that have the same record-type as the focus record. (Abbreviations are in parentheses.)

BLUE (BL)	TURQ (T)
GREEN (G)	WHITE (W)
PINK (P)	YELLOW (Y)
RED	

Data Editor Primary Commands

The following Data Editor primary commands have been updated or included in FileKit 3.60.

CHANGE

Syntax:

```
                 +- EQ -+
                 |      |
>>--- Change --+-+------+----- string1 ------+----- string2 ----- ... ------->
               | |      |                    |
               | +- op -+                    |
               |                             |
               +- VALID ---------------------+
               |                             |
               +- INVALID -------------------+
               |                             |
               +- ANYValue ------------------+
```

Description:

FileKit 3.60 introduces support for parameter keyword "ANYVALUE" as an alternative to *string1* on the CHANGE command.

Use of ANYVALUE will change the contents of a field to *string2* no matter what its original value. For example, the following will change all values in the field named ALBUM_ARTIST to the commercial at symbol ("@").

```
CHANGE  ANYVALUE '@'  ALL  ( ALBUM_ARTIST )
```

INSERT

Syntax:

```
             +--- 1 -----+
             |           |
>>- Insert --+-----------+---+---------------+------------- ... ------------->
             |           |   |               |
             +- n_lines -+   +- record_type -+

 >---+-----------------------+-----+------------------------------+---------->
     |                       |     |                              |
     |    +- , --------+     |     |          +- , ----------+    |
     |    v            |     |     |          v              |    |
     +- (-- field_col -+- ) -+     +- Values(-- field_value -+-) -+

 >---+-------------------------------------+---+--------------+------------><
     |                                     |   |              |
     +- RANDomize -+-----------------------+   +- SKIPerrors -+
                   |                       |
                   |    +- , --------+     |
                   |    v            |     |
                   +- (-- field_col -+- ) -+
```

Description:

FileKit 3.60 introduces support for parameter keywords "RANDOMIZER" and "SKIPERRORS" on the INSERT command.

By default, lines inserted by INSERT will contain random values in each field for which a randomizer object has been defined **and** no specific value has been specified.

RANDOMIZE will generate random values for the specified list of field columns (*field_col*). If specified without a parenthesised list of field columns, then RANDOMIZE will generate random values for **all** fields. If a field column does not already have a randomizer object defined, then one will be automatically created with defaults that match the field's source data type.

Explicitly specifying a null list of field columns, **RANDOMIZE()**, will suppress the default action of generating random values for fields defined with a randomizer object. Instead, the default field value will be 0 (zero) for numeric fields, null for bit and hex fields, and blanks otherwise.

SKIPERRORS will allow insert processing to continue even if an error has occurred whilst attempting to insert a field value.

REPLACELINE

Syntax:

```
>>- REPLACELine --+----------------------+---+-----------------------------+->
                  |                      |   |                             |
                  |   +- , --------+     |   |         +- , ----------+    |
                  |   v            |     |   |         v              |    |
                  +- (- field_col -+-) -+    +- Values(- field_value -+-) -+

 >---+------------------------------------+---+--------------+------------><
     |                                    |   |              |
     +- RANDomize -+----------------------+   +- SKIPerrors -+
                   |                      |
                   |    +-- , ------+     |
                   |    v           |     |
                   +- (-- field_col-+- ) -+
```

Description:

FileKit 3.60 supports parameter keywords "RANDOMIZER" and "SKIPERRORS" on the REPLACELINE command.

By default, only field values for which new values have been specified are replaced in the focus line.

RANDOMIZE will generate random values for each field in the specified list of field columns (*field_col*), provided no specific value has been specified for that field. If a specified field column does not already have a randomizer object defined, then one will be automatically created with defaults that match the field's source data type.

If specified without a parenthesised list of field columns, then RANDOMIZE will generate random values for all fields for which a randomizer object has been defined **and** no specific value has been specified.

SKIPERRORS will allow processing to continue even if an error has occurred whilst attempting to replace a field value.

SELECT

Syntax:

```
>>-- SELect --+----------------------------------------+--------------------><
              |                                        |
              |     +-- ALL  -------------------+      |
              |     |                           |      |
              +-----+-- * ----------------------+------+
              |                                        |
              |     +----------- , -------------+      |
              |     v                           |      |
              +-----+-- field_col --+--------+--+------+
              |                     |        |         |
              |                     +- Hold -+         |
              |                                        |
              |              +---- , --------+         |
              |              v               |         |
              +- - (minus) --+-- field_col --+---------+
```

Description:

FileKit 3.60 introduces support for a minus ("-") character which precedes the list of field columns. Instead of selecting the field columns for display, they will be removed from the current display of field columns.

USE

Syntax:

```
>>- USE -+- record_type -+-+-----------+-+---------------------------------+->
         |               | |           | |                                 |
         +- UNMAPPED ----+ + TEMPORARY + +- IN -+-----------+- struct_name +
         +- RECORD ------+                      |           |
                                                + STRUCTURE +

                       +-- ON ---+
                       |         |
   +-- ALWAYS ---------+---------+------------------+
   |                   |         |                  |
   |                   +-- OFF --+                  |
   |                                                |
 >-+-- WHEN -----------+----------------+-----------+-----------------------><
   |                   |                |           |
   |                   +-- expression --+           |
   |                                                |
   |                   +-- ON ---+                  |
   |                   |         |                  |
   +-- NEVER ----------+---------+------------------+
   |                   |         |                  |
   |                   +-- OFF --+                  |
   |                                   +-- ON ---+  |
   |                                   |         |  |
   +-+-------+- FOCUS -+------------+--+---------+--+
     |       |         |            |  |         |
     +- FOR -+         +-- RECord --+  +-- OFF --+
```

Description:

The USE command is used to assign (or unassign) a specific SDO record-type definition to records of a data set. FileKit 3.60 introduces support for UNMAPPED (synonym RECORD) in place of the record-type name, indicating that the unmapped record data mapping may be assigned.

XMLIMPORT

Syntax:

```
>>--- XMLIMPort ---- XmlFile ---- StructuredFile ---------------------------><
```

Description:

FileKit 3.60 introduces primary command XMLIMPORT which processes an eXtensible Markup Language (XML) data set (*XmlFile*) and converts it into a **"Tree"** formatted structured dataset (*StructuredFile*) which may be mapped using supplied mapping structure:

`"<SystemHLQ>.SZZSDIST.SDO(TREE)"`

This SDO structure contains two record-types for mapping the structured record data:

1. **TAG** comprising the following fields:

Field Name	*Description*
Id	T=Tag C=Content
Indent	Indentation (Level relative to previous)
Lev	Level (root=1)
NewL	nl=First item on the source line
Clos	Self="<xxx />" self-closing tag.
LineNo	XML source line number
Tag	Tag name
AttrN	Number of attributes.
Attr	An array of up to 32 attribute name/value pairs
Name(1)	1st Attribute Name
Value(1)	1st Attribute Value
Name(n)	nth Attribute Name
Value(n)	nth Attribute Value

2. **CONTENT** comprising the following fields:

Field Name	*Description*
Id	T=Tag C=Content
Indent	Indentation (Level relative to previous)
Lev	Level (root=1)
NewL	nl=First item on the source line
Clos	Self="<xxx />" self-closing tag.
LineNo	XML source line number
ParentTag	Name of Parent Tag
Length	Length of Content Text
Text	Content Text

Data Editor SET/QUERY/EXTRACT Options

The following Data Editor options have been included or updated in FileKit 3.60.

FOCUS

Syntax:

```
>>--- EXTract --- /FOCUS/ --------------------------------------------------><
```

Description:

EXTRACT FOCUS now also assigns the 8-byte line identifier (TTR or RBA) as a 16-byte printable hex value to REXX compound variable FOCUS.12. For example, a record at relative track number 2, block (physical record) number 1 and decimal offset 900 within the block, would return a FOCUS.12 value of "0000000002010384".

FIELD, FIELDQP and FIELDQF

Syntax:

```
>>--- EXTract --- / -+- FIELD ---+- / -------------------------------------><
                     |           |
                     +- FIELDQP -+
                     |           |
                     +- FIELDQF -+
```

Description:

For each named field column in the focus record, the EXTRACT FIELD will assign a string of blank delimited values to a REXX variable name equal to the **unqualified** field name. These values are: the field reference number, the field data type, the field name, the currently displayed field width, and the FVALUE REXX variable name.

FileKit 3.60 introduces support for EXTRACT FIELDQP and EXTRACT FIELDQF. These versions of EXTRACT FIELD assign the blank delimited field values to REXX compound variable names equal to the **partially-qualified** and **fully-qualified** field names respectively.

Unqualified Field Name
The elementary field name with no leading record-type or parent structure (group) name(s) included. For example, a variable name of "RELEASE_YYYY" for a field "RELEASE_YYYY" in group "RELEASE_DATE" in record-type mapping "TRACK".

Partially-Qualified Field Name
The field name with no leading record-type, but with any parent structure (group) name(s) included. For example, a variable name of "RELEASE_DATE.RELEASE_YYYY" for the field already described.

Fully-Qualified Field Name
The field name with leading record-type and parent structure (group) name(s) included. For example, a variable name of "TRACK.RELEASE_DATE.RELEASE_YYYY" for the field already described.

FVALUE, FVALUEQP and FVALUEQF

Syntax:

```
>>--- EXTract --- / -+- FVALUE ----+---+-------------+- / ------------------><
                     |             |   |             |
                     +- FVALUEQP --+   +- fieldname -+
                     |             |
                     +- FVALUEQF --+
```

Description:

For each field displayed in the focus record, the EXTRACT FVALUE, EXTRACT FVALUEQP and EXTRACT FVALUEQF now also generate REXX compound variables FVALUE.i, FVALUEQP.i or FVALUEQF.i respectively, where "i" is an index to the focus record field. The value assigned to variable will be the **unqualified** (FVALUE.i), **partially-qualified** (FVALUEQP.i), or **fully-qualified** (FVALUEQF.i) field name of the indexed field.

For example, consider a focus record assigned the record-type "TRACK" which contains a group field "RELEASE_DATE" comprising the 3 fields "RELEASE_YYYY", "RELEASE_MM" and "RELEASE_DD". If these 3 fields are the only fields selected for display, then the additional REXX variables set for each of the EXTRACT FVALUE variants will be as follows:

EXTRACT FVALUE:

FVALUE.1	RELEASE_YYYY
FVALUE.2	RELEASE_MM
FVALUE.3	RELEASE_DD

EXTRACT FVALUEQP:

FVALUEQP.1	RELEASE_DATE.RELEASE_YYYY
FVALUEQP.2	RELEASE_DATE.RELEASE_MM
FVALUEQP.3	RELEASE_DATE.RELEASE_DD

EXTRACT FVALUEQF:

FVALUEQF.1	TRACK.RELEASE_DATE.RELEASE_YYYY
FVALUEQF.2	TRACK.RELEASE_DATE.RELEASE_MM
FVALUEQF.3	TRACK.RELEASE_DATE.RELEASE_DD

MAXBATCHRC

Syntax:

```
    +- SET -----+
    |           |
>>-+-----------+-- MAXBATCHRC ------ value ---------------------------------><


>>--- Query ------ MAXBATCHRC ----------------------------------------------><


>>--- EXTract --- /MAXBATCHRC/  -------------------------------------------><
```

Description:

FileKit 3.60 introduces the MAXBATCHRC option which controls the maximum acceptible return code for FileKit batch (FILEKITB) processing.

FILEKITB sequentially executes the Data Editor commands passed to it via the SDEIN DD input. If the command currently being executed finishes with a FileKit return code value which is greater than the MAXBATCHRC value, then processing stops and subsequent commands are flushed.

The MAXBATCHRC setting is not saved and so should be set as the first command in SDEIN. The default value for MAXBATCHRC is 999.

FileKit File Search & Update

The FileKit File Search & Update utility is capable of performing a range of operations on both formatted and unformatted data in one or more data sets and/or library members:

1. File Search.
2. File Search, Change and Update.
3. File Search, Change and Copy to a different data set.
4. File Copy with Remap of record fields. (Formatted records only)
5. File Search, Change and Remap fields. (Formatted records only)

FileKit 3.60 includes the following enhancements to the FileKit File Search & Update Utility.

FSU - Text Report Output

FSU utility primary command support for parameter LIST=TEXT (and LIST=FMT).

LIST=FMT is default for online processing and will output the result of the FSU execution to a Data Editor display of formatted FSU report records.

LIST=TEXT is default for batch (FILEKITB) processing and will produce a printable text report.

The following is a sample FSU text report output generated by a file search on records mapped by a COBOL copybook structure. Only records containing "Bob Dylan" in the "ARTIST" field, and numeric value "17" in the "TRACK-NUM" field will be reported.

```
1FileKit 3.60      File-Search/Update        2024/04/15 16:02:33    Page      1

   >>fsu inp(USER123.SELCTRN.ZZST1DAT) using COBOL USER123.SELCTR>>
   >>N.SAM1(ZZST1CPC) find( (17 (TRACK-NUM)) & ('Bob Dylan' (ARTI>>
   >>ST)) ) list=text<<






 File Id: USER123.SELCTRN.ZZST1DAT

 Record: 168                                        (lrecl=407)

 Record Type: TRACK
 Field                         Value                                                                                                 I
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0 -
 PERSISTENT-ID                |CA174FF6527010C4
 TRACK-NUM                    | 17
 TRACK-ID                     |4249
 NAME                         |You Ain't Goin' Nowhere
                   (+100    ) |
 ARTIST                       |Bob Dylan
 ALBUM                        |The Essential Bob Dylan
 TOTAL-TIME                   |     163680
 FILE-SIZE                    |    6769903
 BIT-RATE                     |   256
 SAMPLE-RATE                  |44100
 YEAR                         |2000
 NORMALIZATION                |  3188
 DISC-NUMBER                  |  1
 ALBUM-ARTIST                 |Bob Dylan
 RELEASE-DATE                 |2000-10-31T00:00:00Z
 DATE-ADDED                   |2012-08-05T19:25:06Z
 DATE-MODIFIED                |2012-08-05T19:36:43Z
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0 -




                  --- Summary ---

        Run Type          : FIND
        Records Processed :       1070
        Files   Processed :          1
        Total   Hits      :          2
        Records Hit       :          1
        Files   Hit       :          1
        Remap  Errors     :          0
        I/O    Errors     :          0
        Change Errors     :          0
        Change Error Recs :          0
        Change Error Files:          0
        Structure Name    : USER123.SELCTRN.SAM1(ZZST1CPC)

                    --- End ---
```

FSU - Formatted FIND/CHANGE

FSU execution of FIND and CHANGE operations has been enhanced in FileKit 3.60 so that the operations may apply to fields in records mapped by different record types.

Syntax:

```
>>-- FSU --+---------------------------------------------------------+------><
           |                                                         |
           +- | UnFormatted Data Options | -+-- | Common Options | --+
           |                              |
           +-- | Formatted Data Options | --+
```

Formatted Data Options:

```
|-- ... Other options ---- | SDE Options | ---------------------------------|
```

SDE Options:

```
          +- SDO ---+
          |         |
|- USING -+---------+- in_struct -----------------------+------------------->
          |         |                                   |
          +- COBOL -+                                   |
          +- PL1 ---+                                   |
          +- ADAta -+                                   |
          |              +-------------------------+    |
          |              v                         |    |
          +- SYMNAMes ( -+--- DFSORT symbols ---+- ) ---+

>-+----------------------------------------+--------------------------------->
  |                                        |
  |                    +----- , -----+     |
  |                    v             |     |
  +--+- RECTYPEs -+ ( --+-- rectype -+- ) -+
     |            |     |                  |
     +- VIEW -----+     +---- * -----------+

  +----------------------------------------------------+
  v                                                    |
>-+----------------------------------------------------+--------------------|
  |                                                    |
  |           +----- , -----+                          |
  |           v             |                          |
  + SELect ( --+-- field ---+-+-+----------------+- ) -+
               |              | |                |
               +---- * -------+ +- FROM rectype -+
```

Description:

Previously, FSU would use the single record-type specified on the VIEW parameter or otherwise the default record-type. Therefore, only records that were assigned to this record-type could be used in the FIND or CHANGE operation.

VIEW (synonym RECTYPES) has been enhanced to support multiple record-type specifications. It identifies the record-type mapping(s) that will be included in the FIND or CHANGE operations. By default, all record-types are included.

Multiple **SELECT** clauses may be specified, once for each record-type, to identify the record-type fields that will be included in the FIND or CHANGE operations. Note that the field selection for a particular record-type will **not** be included if VIEW is also specified and it does not identify the record-type to which the fields belong.

Using VIEW and SELECT parameters, the user may define the scope of the FIND and CHANGE operations.

Example:

The following FSU command will report on records containing a hit for ampersand ("&") in the "NAME" field only, in records mapped by the record-types "ALBUM" and "TRACK":

```
FSU INPUT  ('JGE.SELCTRN.ZZST2DAT')    USING SDO 'JGE.SELCTRN.SDO(ZZST2)'
           VIEW    ALBUM,TRACK
           SELECT (NAME FROM ALBUM)
           SELECT (NAME FROM TRACK)
           FIND   ( '&mp;')
```

Note that SELECT and VIEW parameters are applied only to FIND and CHANGE operations. They are **not** used to filter record and field output.

FSU - COPY Output

FSU execution to copy records to another data set has been enhanced in FileKit 3.60 so that output records may be filtered. Filtering can output only records for which a search string has been found, or records in which a change has occurred.

Previously, FSU would only copy all input records (or all records that satisfied a supplied input record FILTER parameter).

The following mutually exclusive parameters have been introduced to filter output records.

- **COPYALL** (the default) will copy *all* records (or all records that satisfy a specified input FILTER).
- **COPYFOUND** will copy only records that satisfy the specified FIND string specifications.
- **COPYCHANGED** will copy only records that have been changed using the CHANGE parameters.

FSU - COPY to New Library

FSU execution to copy library member records to a new library has been enhanced in FileKit 3.60 so that new library may inherit the source library SMS classes, and may be defined as a PDSE V2 with a specified maximum number of member generations.

The following parameters have been introduced specifying attributes used when allocating the new library:

- **COPYDATACLASS** uses the source library SMS DATA CLASS.
- **COPYMGMTCLASS** uses the source library SMS MANAGEMANT CLASS.
- **COPYSTORCLASS** uses the source library SMS STORAGE CLASS.
- **GENS *int*** indicates a PDSE V2 library with MAXGENS *int*.

Each of these parameters also imply parameter "NEW".

FileKit File Compare

The FileKit File Compare utility is used to compare formatted and unformatted record data belonging to 2 input data set sources and produce a comprehensive report.

FileKit 3.60 includes a the following enhancements to the FileKit File Compare Utility.

COMPFILE - Text Report Output

COMPFILE utility primary command support for parameter LIST=TEXT (and LIST=FMT).

LIST=FMT is default for online processing and will output the result of the COMPFILE execution to a Data Editor display of formatted COMPFILE report records. The report output is displayed as formatted records where different record type mappings are used to display fields in the report.

LIST=TEXT is default for batch (FILEKITB) processing and will produce a printable text report. The text report is split into pages as determined by the prevailing FileKit Data Edit PAGEDEPTH value. Each page has a standard page heading which includes a timestamp and page number.

The following is a COMPFILE command and its generated text report output. 2 files are compared with records mapped by a common SDO structure. The report shows the new/old record numbers, field names for which values differ, and the new/old values themselves.

```
COMPFILE  NBJ.SELCTRN.ZZST1DAT       NBJ.SELCTRN.ZZST1DAT.BKUP
               USING NBJ.FILEKIT.SDO(ZZST1)
          LIST=TEXT   INCFIELDS   INCMATCHED    SYNC 1TO1
```

```
1FileKit 3.60         Compare Files          2024/04/10 14:17:48    Page      1

   >>COMPFILE NBJ.SELCTRN.ZZST1DAT NBJ.SELCTRN.ZZST1DAT.BKUP USIN>>
   >>G NBJ.FILEKIT.SDO(ZZST1) LIST=TEXT INCFIELDS INCMATCHED SYNC>>
   >> 1TO1<<


  File Dataset Name                                            Lang  Mapping File
  ---- ------------------------------------------------------ ----  ------------------------------------------------------
  New  NBJ.SELCTRN.ZZST1DAT                                          NBJ.FILEKIT.SDO(ZZST1)
  Old  NBJ.SELCTRN.ZZST1DAT.BKUP                                     NBJ.FILEKIT.SDO(ZZST1)




 NEW Record: 1             OLD Record: 1             --- Record Changed  ---       (lrecl=407)

 Record Type: TRACK
 Field                        New Value                                           Old Value
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5  ----+----1----+----2----+----3----+----4----+----5
 NAME                        |Rolling In the Xeep                                 |Rolling In the Deep
                  (+50     ) |                                                   =|
                  (+100    ) |                                                   =|
 FILE-SIZE                   |    8050806                                         |         25
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5  ----+----1----+----2----+----3----+----4----+----5



 NEW Record: 2             OLD Record: 2             --- Record Changed  ---       (lrecl=407)

 Record Type: TRACK
 Field                        New Value                                           Old Value
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5  ----+----1----+----2----+----3----+----4----+----5
 FILE-SIZE                   |    7942537                                         |         39
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5  ----+----1----+----2----+----3----+----4----+----5



 NEW Record: 3             OLD Record: 3             --- Record Changed  ---       (lrecl=407)

 Record Type: TRACK
 Field                        New Value                                           Old Value
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5  ----+----1----+----2----+----3----+----4----+----5
 FILE-SIZE                   |    8774303                                         |         36
 ----+----1----+----2----+---  ----+----1----+----2----+----3----+----4----+----5  ----+----1----+----2----+----3----+----4----+----5
```

COMPFILE - HFS/ZFS File Input

The Compare File utility already supports specification of a z/OS HFS/ZFS file as the NEW and/or OLD files to be compared. FileKit 3.60 enhances this support, providing new COMPFILE primary command options that explicitly specify the format of the HFS/ZFS file records.

Applicable to all (NEW and/or OLD) HFS files, the following options may be specified to determine how HFS/ZFS file data is processed by the utility. Note that, for non-HFS files, these options are ignored.

HFS options may not currently be specified independently for the NEW and OLD files.

`EOL=STD|NL|CR|LF|CRLF|LFCR|CRNL|`*`string`*

Specify the EOLIN (input end-of-line) delimiter used to identify the end of each record for unformatted (non-RECFM F or V) HFS file input. EOLIN delimiters are not included in the edited record data or record length. EOL parameter elements are as follow:

STD	-	Any standard line-end.
NL	X'15'	New Line.
CR	X'0D'	Carriage Return.
LF	X'0A'	Line Feed.
string	-	A 2-byte user specified character or hex string.

STD is default so that the EDIT operation scans the input data for any of the standard EOL combinations (not *string*), stopping when one is found. This EOL combination is used as EOLIN for the file.

`RECFM F | V (`*`off,len,origin`*`)`

Specifies that the data is to be treated as containing Fixed or Variable length format records.

RECFM F indicates that all records are of a fixed length as defined by the LRECL argument.

RECFM V allows the user to specify the location of the record length fields within the data as follows:

off	Offset of the record length field from the start of the record.
len	Length of the record length field.
origin	The start of the record data at which the record length is applied.

Default is (0,2,0) which describes variable length records that have a 2-byte record length field at offset 0, and record data at offset 2. Note that the record length value includes the length of the length field itself.

`LRECL `*`lrecl`*

Specifies the maximum record length of input HFS file records.

For RECFM F HFS files, *lrecl* is the fixed length of the records processed. If the HFS file size is not a multiple of *lrecl* value, then the last record of the file will be short.
Default *lrecl* for this type of file is 80.

For RECFM V and unformatted (EOL delimitted records) HFS files, if a record length exceeds *lrecl*, processing is stopped for that particular file.
Default *lrecl* for these types of files is 32752.

COMPFILE - Primary Command

In addition to LIST output and HFS/ZFS input file options, the COMPFILE primary command has been enhanced to support the following parameters:

`FCHUNK `*`n_cols`*

New in FileKit 3.60, each "Field" report record not only identifies the name of a mapped field, but also its offset ("zOffset"), new value ("zNewValue") and/or old value ("ZOldValue") in external format.

Applicable only to the compare of formatted records where the report output is displayed as formatted records (LIST=FMT), FCHUNK will override the default width of 50 used to display the field's new ("ZNewValue") and/or old ("ZOldValue") external value in a "Field" record.

Where the length of the external value of a given field exceeds the FCHUNK *n_cols* value, the complete value will span multiple "Field" records.

`FILTER | NFILTER | OFILTER` *`filter_clause`*

FILTER specifies additional record filtering criteria which is applied to both the NEW and OLD files to restrict the records processed by COMPFILE.

Alternatively, *filter_clause* may be specified differently for the NEW and OLD file using NFILTER and OFILTER respectively.

FILTER parameters are specified via a Data Editor filter clause which may be supplied in parentheses as part of the COMPFILE command, or referenced via *filter_fileid*, a separate sequential data set, PDS/PDSE member or HFS file. A filter file must contain the keyword FILTER followed by a valid filter clause.

`INCFIELDS`

Applicable only to the compare of formatted records, INCFIELDS indicates that report output "Field" type records may be included for any changed record, or all reported records when INCKEYFIELDS is specified for key synchronisation.

Each "Field" report record identifies the name of a mapped field, its offset ("zOffset") and its new value ("zNewValue") and/or old value ("ZOldValue") in external format.

`INCKEYFIELDS`

Applicable only to the report output displayed as plain text (LIST=TEXT), INCKEYFIELDS indicates that, when sorted or unsorted KEY synchronisation is specified, then the report display of each record must include the name and value of every key field.

This makes each inserted, deleted, changed, and possibly even matching record, easier to identify than simply by record number alone.

`ARBC | NARBC | OARBC` *`char`*

ARBC activates and specifies an arbitrary character, which is treated as a compare wildcard in both the NEW and OLD files. An arbitrary character in one record of the compare record pair will match any character in the equivalent position of the other record.

Alternatively, the arbitrary character may be apply in one direction only, or may be specified differently for the NEW and OLD file using NARBC and OARBC respectively.

Note that, when specified for a compare of formatted records, ARBC, NARBC and OARBC apply to character fields only.

Typically, OARBC is used to mask acceptable differences in your NEW file when compared with an OLD master file. To do this, simply update your OLD master by typing the specified arbitrary character over any areas where differences are expected and acceptable (such as timestamps, product release numbers, etc.), and execute COMPFILE with parameter OARBC to identify the arbitrary character used. For example, `OARBC="?"` if a question mark symbol ("?") is used as the arbitrary character in the OLD file data.

Similarly, `NARBC=X'00'` would indicate that a NULL character value in the NEW file record will match any character in the equivalent position of the OLD file record.

`DELC | NDELC | ODELC` *`char`*

All occurrences of the character specified using DELC in records belonging to both the NEW and OLD files will be removed before comparison takes place.

Alternatively, the character may be removed from records belonging to one of the files, or specified differently for the NEW and OLD file using NDELC and ODELC respectively.

Note that, when specified for a compare of formatted records, DELC, NDELC and ODELC apply to character fields only.

For example, `ODELC='-'` indicates that all hyphon symbols ("-") are to be removed from the OLD file record or record character fields before comparison occurs. `NDELC=X'00'` indicates that all NULL character values are to be removed from the NEW file record before the compare occurs.

`NEWPAGE` *`n_lines`*

Applicable only to the compare of formatted records with the report output displayed as plain text (LIST=TEXT), NEWPAGE will cause output of a new report page if *n_lines* lines are not available on the current page prior to writing report output for a new record (or synchronised record pair).

Note that the report page depth is defined by the prevailing Data-Edit **SET PAGEDEPTH** option.

`SPACE` *`n_blanks`*

Applicable only to character data (fields of character data type or unformatted record data), SPACE indicates that a record pair will be considered a match if the only difference is the number of blanks between words in the text.

Prior to comparison of data in the record pair, groups of one or more blank characters are condensed (or padded) to a fixed number of *n_blanks* blanks. The default value for *n_blanks* is **1**, meaning all occurrences of multiple blanks will be treated as a single blank.

Therefore, an equal condition is set when the record pair contains differences only as a result of there being different numbers of blanks in each record.

Specifying a value greater than 1 for *n_blanks* is meaningless. However, **SPACE=0** is significant, removing all blanks before the compare occurs, and so removing the concept of blank delimited words.

`STRIP | NSTRIP | OSTRIP` *`char`*

Applicable only to character data (fields of character data type or unformatted record data), STRIP indicates that **trailing** characters in the compare data of both the NEW and OLD files are to be stripped (so reducing the compare length) and optionally specifies the strip character *char*. For each occurrence of *char* occupying the last position of the compare data the compare length is reduced by 1.

Alternatively, a different trailing strip *char* may be specified for compare data in the NEW and OLD files using NSTRIP and OSTRIP respectively.

If STRIP/NSTRIP/OSTRIP is not specified, then no characters are stripped from the compare data. If any of these parameters are specified without *char*, the default is blank (X'40').

`STRIPL | NSTRIPL | OSTRIPL` *`char`*

Applicable only to character data (fields of character data type or unformatted record data), STRIPL indicates that **leading** characters in the compare data of both the NEW and OLD files are to be stripped (so reducing the compare length) and optionally specifies the strip character *char*. For each occurrence of *char* occupying the first position of the compare data the compare position is increased by 1 and the compare length is reduced by 1.

Alternatively, a different strip *char* may be specified for compare data in the NEW and OLD files using NSTRIPL and OSTRIPL respectively.

If STRIPL/NSTRIPL/OSTRIPL is not specified, then no characters are stripped from the compare data. If any of these parameters are specified without *char*, the default is blank (X'40').

`USING SYMNAMES (` *`DFSORT symbols`* `)`

Applicable only to the compare of formatted records, USING SYMNAMES provides an alternative to providing a source (COBOL, PL1, HALSM, SDO) structure name to format the NEW and OLD records.

The USING SYMNAMES parameter may be specified separately, once each for mapping records in the NEW and OLD files. For each USING SYMNAMES specification, DFSORT style SYMNAME syntax must be used to specify a parenthesised list of field definitions. The order in which symbol definitions are supplied dictates the order in which the fields will occur in the record type definition.

In the following example, a field name "RECNO" is defined as a 4-byte binary numeric value at position 1 of each input record of the new and old data sets. Since only mapped record fields are compared and no other fields are defined, matching will occur only on the values in this one field. The contents of the rest of the old/new record pair are not tested.

```
    COMPFILE  'NBJ.DATA.NEW'  'NBJ.DATA.OLD'     USING SYMNAMES ( RECNO,01,4,BN  )
```

The field definitions within the SYMNAMES parentheses may be supplied in-line and/or via named data sets/library members.

```
    SYMNAMES( Card,06,04,CH  Dept,46,03,CH  Amount,49,06,PD  )
    SYMNAMES( SYS1.MACLIB(EDGSMFSY)  SYS1.MACLIB(EDGSRCSY) )
    SYMNAMES( CBL.DFSORT.SYM(CBLATRAC)  TCB,*,4,BI )
```

FileKit File I/O

First introduced in release 3.50, the FileKit File I/O utility (FILEIO) has been developed to be a powerful replacement for the TSO/E REXX extension, EXECIO.

For use in FileKit REXX macros, FILEIO may be used to perform file input/output on any of the following:

- Physical sequential data set
- PDS/PDSE library member or member generation
- VSAM data set,
- Unix (HFS/ZFS) file
- DB2 table or SQL result table
- Multiple members of a library or concatenation of libraries (**BPAM**)

Where **BPAM** access to multiple members across a concatenation of multiple libraries is requested, then the user may choose to do one of the following:

- Process all matching member names from each library in sequence.
- Treat the concatenation as a **library search path**, processing only the first occurrence of each named member.

FILEIO may be thought of as being similar to the TSO/E REXX extension, **EXECIO**. Unlike EXECIO, FILEIO may be executed in REXX macros run by FILEKITB (the FileKit batch interface) or via a FileKit VTAM login.

FILEIO also has features that make it a great deal more powerful than **EXECIO**.

Just like EXECIO, FILEIO operations include the following:

- Read a single unformatted record into a named REXX variable.
- Write a single unformatted record from a REXX variable or literal.
- Read multiple unformatted records into a named REXX stem variable.
- Write multiple unformatted records from a named REXX stem variable.

However, FILEIO operations also include the following:

- Read a single record formatted by a structure (e.g. COBOL/PL1) and transfer (selected) field values into separate REXX variables.
- Write a single record formatted by a structure, populating field values from separate REXX variables.
- Perform direct reads on any VSAM (KSDS/ESDS/RRDS/VRDS) file.
- Perform direct reads on any unix (HFS/ZFS) file.
- Update, insert and delete records.
- Filter records by content using powerful selection criteria including reference by COBOL/PL1 field name.
- Extract information from SMF records and navigate their extremely complex structures.
- Generate Test Data, including sequenced or random numbers, dates/times, etc. and even personal names.
- Disguise existing data by performing "keyed" look-up/substitution

Unlike EXECIO, the FILEIO feature allows the user to refer to files directly by dataset name, without need for a prior allocation of the dataset to a **"DD" name**. However, for convenience and brevity, an (up to 8-character) user name may be assigned on the OPEN operation, meaning that from then on any FILEIO operations for that file may refer to it by its short user name.

FILEIO - Primary Command

Syntax:

```
>>--+- FILEIO -+-- filename --+----| Open/Close_Operations | -----+---------><
    |          |              |                                  |
    +- FIO ----+              +----| Input_Operations | ----------+
                              |                                  |
                              +----| Output_Operations | ---------+
                              |                                  |
                              +----| Selection_Operations | ------+
                              |                                  |
                              +----| Test_Data_Operations | ------+
                              |                                  |
                              +----| Options | -------------------+
                              |                                  |
                              +------ CMD command_text -----------+
```

Open/Close_Operations:

```
|-+- OPENread ----+--| Open_Options |-+---------------------------------+-+-|
  |               |                   |                                 | |
  +- OPENWrite ---+                   +- USING -- | Mapping_Structure |-+ |
  |               |                                                       |
  +- OPENUpdate --+                                                       |
  |               |                                                       |
  +- OPENInsert --+                                                       |
  |                                                                       |
  |                                                                       |
  +---- CLOSE ------------------------------------------------------------+
```

Open_Options:

```
|---+------------------+--+--------------------------------+--+-------+----->
    |                  |  |                                |  |       |
    +- NAME user_name -+  +- CONCATENATEDLIBRARYDIRectory -+  +- IOP -+
                          +- CLD --------------------------+
                          |                                |
                          +- LIBRARY ----------------------+
                          +- PDS --------------------------+
                          |                                |
                          +- | DB2_options |---------------+
                          |                                |
                          +- | HFS_Options |---------------+

>---+----------------+--+-------------+--+----------+------------------------|
    |                |  |             |  |          |
    +- ILIMit nrecs -+  +- LRECL nnn -+  +- APPend -+
```

DB2_Options:

```
|-+-------+--+--------------------------+--+--------------+--+----------+---->
  |       |  |                          |  |              |  |          |
  +- DB2 -+  +- LOCKTable +- EXclusive -+  +- SKIPLocked -+  +- SCROLL -+
                          +- SHare -----+
                          +- SHR -------+

>-+---------------------------------------------+--+----------------+-------->
  |                                             |  |                |
  +-- WITH --+- CS --+-+------------------------+  +- EDITPRIMEKEY -+
             +- RR --+ |                        |
             +- RS --+ +- KEEP --+- EXclusive --+
             +- UR --+           +- UPDate -----+
                                 +- SHare ------+
                                 +- SHR --------+

>-+------------------------+-+------------------------------------------+-+-->
  |                        | |                                          | |
  +- WHERE (where_clause) -+ +- ORDER -+------+--+- (order_by_clause)  -+ |
  |                          |         |      |  |                      | |
  |                          |         +- BY -+  |                      | |
  |                          +- SORT ------------+                      | |
  |                          |                                          | |
  |                          +- SORTIndex -+--+-- index_name --+--------+ |
  |                          +- SORTIx ----+  |                |          |
  |                                           +-- Prime -------+          |
  |                                                                       |
  +--- SQL -+- ( sql_syntax ) --+-----------------------------------------+
            |                   |
            +--- sql_file ------+

>-+----------------------------------+---------------------------------------|
  |                                  |
  |            +----- , ------+      |
  |            v              |      |
  +- SELect ( -+- field_name -+- ) --+
```

HFS_Options:

```
                        +-- STD -----+
                        |            |
             +-- EOL ---+------------+----------------+
             |          |            |                |
             |          +-- CR ------+                |
             |          +-- LF ------+                |
             |          +-- NL ------+                |
             |          +-- CRLF ----+                |
             |          +-- LFCR ----+                |
             |          +-- CRNL ----+                |
             |          +-- string --+                |  +-- LRECL lrecl -+
             |                                        |  |                |
          |--+----------------------------------------+--+----------------+------------|
             |                                        |
             +-- NOEOL -------------------------------+
             |                                        |
             +-- RECFM -+- F --+----------------------+
                        +- V --+
                        +- V1 -+
                        +- V2 -+
                        +- V3 -+
```

Mapping_Structure:

```
               +- SDO ---+
               |         |
          |-+--+---------+--- copybook_or_mapping_file ---+----------------------------|
            |  |         |                                |
            |  +- COBol -+                                |
            |  |         |                                |
            |  +- PL1 ---+                                |
            |  |         |                                |
            |  +- ADAta -+                                |
            |                                             |
            +---- SYMNAMES ( DFSORT_symbols ) ------------+
```

Input_Operations:

```
          |--+-+- READ --------+--+------------+--+------------------------+-+---------|
             | |               |  |            |  |                        | |
             | +- READSEGment -+  +- rexxvar --+  +- KEY string -----------+ |
             |                                    |                        | |
             |                                    +- RBA nnnn -------------+ |
             |                                    |                        | |
             |                                    +- RECno nnnn -----------+ |
             |                                    |                        | |
             |                                    +- TTR ttr --------------+ |
             |                                    |                        | |
             |                                    +- RECLen record_length -+ |
             |                                                               |
             +-+- FVALUE ---+---+---------+----------------------------------+
             | |            |   |         |                                  |
             | +- FVALUEQP -+   +- READ --+                                  |
             | |            |                                                |
             | +- FVALUEQF -+                                                |
             |                                                               |
             +--- CREAD -----------------------+---------+-------------------+
             |                                 |         |                   |
             |                                 +- nrecs -+                   |
             |                                                               |
             +-+- SREAD --------+-- stem_name -+---------+-------------------+
             | |                |              |         |                   |
             | +- SREADSEGment -+              +- nrecs -+                   |
             |                                                               |
             +--- RECLen record_length --------------------------------------+
             |                                                               |
             +--- SKIP nrecs ------------------------------------------------+
```

Output_Operations:

```
|--+--- DELete ---------------------------------------------+----------------|
   |                                                         |
   +-+- INSert -------+--- text_literal --------------------+
   | |                |                                      |
   | +- UPDate -------+                                      |
   | |                |                                      |
   | +- WRite --------+                                      |
   |                                                         |
   +-+- INSERTV|INSV -+--- rexxvar ------------------------+
   | |                |                                      |
   | +- UPDATEV|UPDV -+                                      |
   | |                |                                      |
   | +- WRITEV |WRV --+                                      |
   |                                                         |
   +-+- INSERTX|INSX -+--- hex_literal ---------------------+
   | |                |                                      |
   | +- UPDATEX|UPDX -+                                      |
   | |                |                                      |
   | +- WRITEX |WRX --+                                      |
   |                                                         |
   +-+- SWRite -------+--- stem_name ----------------------+
   | |                |                                      |
   | +- SWRITEX ------+                                      |
   |                                                         |
   +-+- FINSert --------+----| Field_Values |--------------+
   | |                  |                                    |
   | +- FUPDate --------+                                    |
   | |                  |                                    |
   | +- FWRite ---------+                                    |
   |                                                         |
   +--- COMMIT --------------------------------------------+
```

Field_Values:

```
|--+---------------+--+--------------------+--+----------------------------+->
   |               |  |                    |  |                            |
   +- record_type -+  |   +- , -------+    |  |         +-- , -------+     |
                      |   v           |    |  |         v            |     |
                      +- ( field_name +-) -+  +- Values( field_value +-) -+

>--+-----------------------------+-----------------------------------------|
   |                             |
   +- RANDomize -----------------+
   |                             |
   |            +-- , ------+    |
   |            v           |    |
   +- RANDomize( field_name +-) -+
```

Selection_Operations:

```
|--+-+- FILTer ---------+---+--------- selection_file -----------+------+---|
   | |                  |   |                                    |      |
   | +- TOFCondition ---+   +- ( -- | Selection_Criteria | -- ) -+      |
   | |                  |                                               |
   | +- EOFCondition ---+                                               |
   |                                                                    |
   +--- MEMBer --+-- library_member_name -------------+-----------------+
   |             |                                    |                 |
   |             +-- library_member_name.generation --+                 |
   |                                                                    |
   +--- NEXTMEMber ---------------------------------------------------+
   |                                                                    |
   +--- PREVMEMber ---------------------------------------------------+
   |                                                                    |
   |               +----- , ------+                                     |
   |               v              |                                     |
   +--- SELect --+-+- field_name -+-+--+--------------------+-----------+
   |             |                  |  |                    |           |
   |             +-- * -------------+  +- FROM record_type -+           |
   |                                                                    |
   |              +------ , ------+                                     |
   |              v               |                                     |
   +--- VIEW -----+- record_type -+-----------------------------------+
   |                                                                    |
   +--- WHere (expr) -+------------------+------------------------------+
                      |                  |
                      +- IN record_type -+
```

Selection_Criteria:

```
     +-----------------------------------------------------------+
     v                                                           |
|--+-+- INclude rec_type --+--------------+--+----------------+-+-+--------->
   |                       |              |  |                |   |
   |                       +- WHere expr -+  +- LIMit n_hits -+   |
   |                                                              |
   | +-----------------------------------------------------------+ |
   | v                                                           | |
   +-+- EXclude rec_type --+--------------+--+----------------+-+-+
                           |              |  |                |
                           +- WHere expr -+  +- LIMit n_hits -+

>--+--------------------+---------------------------------------------------|
   |                    |
   +- Stopafter n_hits -+
```

Test_Data_Operations:

```
                                               (1)
|-- RANDomize -- field_name -+----------------+- | Randomize_Options | -----|
                             |                |
                             +- FOR rec_type -+
```

(1) See "Randomize Options" for Test Data - FSU

Options:

```
                  +------ , --------+
                  v                 |
|--+-- OPTions ---+- ALIAS ---------+----------+----------------------------|
   |              +- ASCII ---------+          |
   |              +- BLANKWHENZERO -+          |
   |              +- CREATED -------+          |
   |              +- CURSIZE -------+          |
   |              +- DSORG   -------+          |
   |              +- DIRectory -----+          |
   |              +- GENeration ----+          |
   |              +- IOP -----------+          |
   |              +- NOIOP ---------+          |
   |              +- KEY -----------+          |
   |              +- KEYLEN --------+          |
   |              +- KEYPOS --------+          |
   |              +- LASTMOD -------+          |
   |              +- LIBNAME -------+          |
   |              +- MEMBER --------+          |
   |              +- MEMLIST -------+          |
   |              +- RANDomize -----+          |
   |              +- RECID ---------+          |
   |              +- RECno ---------+          |
   |              +- RECTYPe -------+          |
   |              +- RESET ---------+          |
   |              +- SEGTYPe -------+          |
   |              +- SPEED ---------+          |
   |              +- USER  ---------+          |
   |              +- ZEROS ---------+          |
   |                                           |
   |              +------ , --------+          |
   |              v                 |          |
   +-- MODRPL ----+- BWD -----------+----------+
                  +- FWD -----------+
                  +- FULLKEY -------+
                  +- GENKEY --------+
                  +- KEYEQ ---------+
                  +- KEYGE ---------+
                  +- LASTREC -------+
                  +- XRBA ----------+
```

Description:

FileKit 3.60 introduces a number of new syntax keywords including support for formatted records, randomized value output, DB2 tables, HFS/ZFS files, Combined Library directory and member input, Library Search path, and many other more features.

A comprehensive description of each FILEIO keyword and sample usage may be found in the "FileKit Data Editor" manual.

Examples:

1. Simple read and output of the first record of a library member allocated to a DDName.

```
address "CBLSDATA"                /* REXX */
"alloc f(INDD) shr reuse da('SYS1.MACLIB(ACB)')"
"fileio  INDD             read       REC1"
say     "SYS1.MACLIB(ACB) Record 1:" REC1
"fileio  INDD             close"
```

2. Read and output of the first record of each library member that matches a specified member name mask. A library path, allocated to DDName "SOURCE", is searched so that the first occurrence of a matching member name is used.

   "LIBNAME" and "MEMBER" options are used to extract the library and member name of the current member being processed for DDName "SOURCE".

```
address "CBLSDATA" /* REXX */

  /* Library search path */
"alloc f(SOURCE) reuse shr da( 'MY.COBOL.SOURCE.REL130' ",
                             " 'MY.COBOL.SOURCE.REL120' ",
                             " 'MY.COBOL.SOURCE.REL110' ",
                             " 'MY.COBOL.SOURCE.REL100' )",

"fileio SOURCE(INP*,OUT*)  open  ConcatenatedLibraryDirectory"
"fileio SOURCE  options LIBNAME MEMBER"

do forever
  "fileio SOURCE  read  MYREC"

  say SOURCE.1.LIBNAME"("SOURCE.1.MEMBER")" "Record 1:" MYREC

  "fileio SOURCE  nextmember"
  if rc <> 0 then leave
end

"fileio  SOURCE close"
```

3. Read all library member records that do not start with "*" (asterisk), and write them to a member of a different library.

```
address "CBLSDATA" /* REXX */

"fileio MY.LIB(TEST)  open"
"fileio MY.LIB(TEST)  where  (record ¬>> '*')"
"fileio MY.LIB(TEST)  sread  MYSTEM"     /* Stem READ. */
"fileio MY.LIB(TEST)  close"

"fileio MY.DUP(TEST)  swrite MYSTEM"     /* Stem WRITE. */
"fileio MY.DUP(TEST)  close"
```

4. Use a COBOL copybook structure to map input records so that only records identifying the 1st album track are read (i.e. the "TRACK-NUM" field is equal to 1), then write these records to another data set.

```
address "CBLSDATA" /* REXX */
STRUCT = "COBOL NBJ.SELCTRN.SAM1(ZZST1CPC)"     /* Mapping structure */

"fileio NBJ.SELCTRN.ZZST1DAT  openread  name III  using" STRUCT
"fileio NBJ.SELCTRN.TRACK1    openwrite name OOO"

"fileio  III where (TRACK-NUM = 1)"

do forever
  "fileio III read   MYREC"; if rc = 4 then leave
  "fileio OOO writev MYREC"
end

"fileio III close"
"fileio OOO close"
```

5. Generate random data for specific fields mapped by a COBOL copybook structure.

```
STRUCT = "COBOL NBJ.SELCTRN.SAM1(ZZST1CPC)"     /* Mapping structure */

"fileio NBJ.SELCTRN.TEST  openwrite  name MUSIC  using" STRUCT

"fileio MUSIC  rand PERSISTENT-ID hex"
"fileio MUSIC  rand TRACK-NUM     range 1 30"
"fileio MUSIC  rand ARTIST        MAlphaNum"  /* Mixed case alphanum */
"fileio MUSIC  rand NAME          Vocab length 50"

"fileio MUSIC  option randomize"
"fileio MUSIC  writerepeat 1000"           /* Generate 1,000 records */
"fileio MUSIC  close"
```

Maintenance Applied

The following table identifies maintenance to CBL Product Suite 3.50 that has been applied at source to CBL Product Suite 3.60.

Details of each SYSMOD may be found at the CBL web page entitled "CBL Product Suite 3.50 Maintenance Summary". Individual descriptions are referenced by links to this web page in the following table.

Service Package Id	SYSMOD
X2022010	RS35001
X2024095	RS35002
X2024096	RS35003