Subject: MCR10099, Enhancements to index_set Command/AF

Author: Gary Dixon

Date: October 30, 2021

MCR10098 mentions that the existing index_set command/active function may be used in analyze_multics (azm) to display a set of structures sharing the same structure declaration. However, index_set accepts only a rigid triplet of integer arguments for specifying the set of numbers to be generated.

Multics Ticket 254 suggests several enhancements for index_set that would make it easier to specify sets in the azm request environment. These include:

- Accept control arguments to:
    - Specify which integers of a set specification triplet (FIRST BOUND INCREMENT) are actually being given.
    - Separate one set specification from the next when several sets are specified in one program invocation.
    - Allow set elements to be output in a different numeric base other than decimal.

That ticket also suggests accepting input integers expressed in a base other than decimal by interpreting an optional radix indicator at the end of each integer. For example: 337o for an octal integer, 3FEx for a hexadecimal integer, etc.

azm requests often output values as octal words (e.g., octal segment numbers or octal offsets in a dump_segment output display). Therefore, many azm requests expect octal inputs to specify a structure location or to select a data structure related to a given octal segment number. Such index_set changes would make it easier to use in the azm environment.

Proposed Changes

Non-decimal input integers: The current index_set accepts only decimal integer arguments in a set specification. These are converted to numbers using the PL/I convert builtin function. index_set will be changed to call the existing cv_integer_string_ and cv_integer_string_check_ subroutines which convert an integer string to a fixed bin(35) number.

For readers unfamiliar with the existing cv_integer_string_ subroutines, they differ in capability from the PL/I convert mechanism by:

- interpreting integer strings using a default base ranging from 2 to 16;
- accepting integer strings that end with an optional radix indicator, permitting any input integer string to override the decimal default base normally used by index_set; and
- permitting arbitrary whitespace (SP and HT character) before and after the integer string, and between any plus or minus sign and the numeric digits of the integer string.

The radix indicators accepted by cv_integer_string_ are:

```
List of radix indicators:
   Indicator characters may also be given in uppercase.
b, _b
   the number is interpreted as a base two number (binary).
o, _o
   the number is interpreted as a base eight number (octal).
d, _d
   the number is interpreted as a base ten number (decimal).
x, _x
   the number is interpreted as a base sixteen number (hexadecimal).
rNN, _rNN
   the number is interpreted in the base NN which is a decimal number
   between 2 and 16 inclusive.
```

Control arguments for set specification: Add a -set control argument to separate one triplet of set specification arguments from those of the next set. -set may optionally appear before the first set specification arguments.

Add -from FIRST (-fm FIRST) to specify the first element of a set. Add -to BOUND to specify the upper bound of a set. Add -for COUNT to specify the number of elements in the set. Either -to BOUND or -for COUNT must be specified for each set. Add -by INCREMENT to specify the increment separating each set element.

Output of set elements: The index_set active function will return each set element as an unquoted string separated by a single space character from any adjacent set element. The index_set command will print elements of each set on one or more terminal lines using a line break before an element that would exceed the terminal line length. A blank line separates output of one set from that of the next set.

Control arguments changing base of output values: Add -octal (-oct), -hexadecimal (-hex) and -decimal (-dec) control arguments to specify the numeric base in which elements of the generated set are printed/returned. -decimal will remain the default output base. A radix indicator will be appended to octal and hexadecimal output strings (e.g., 337o or dfx). The existing numeric_to_ascii_base_ subroutine will be called to generate each output string. That subroutine outputs all letters in lowercase.

Add -unspec to specify output as a 12-digit octal number representing bits of unspec(number) where number is the fixed bin(35) value of the set element. For example, a set element of 223 (decimal) would be expressed in -unspec format as: 000000000337. An element of -1 would be expressed as: 777777777777. No radix indicator is appended to words in the -unspec output format.

Note that appending a radix indicator to each -octal output integer introduces a problem, because azm requests expecting octal inputs call cv_oct_check_ to interpret such arguments. The cv_oct_ and cv_oct_check_ subroutines do not currently accept strings ending with a radix indicator. This problem will be resolved in a separate MCR in the near future.

Short name: The accepted short name for the -index control argument is -ix. Following this precedent, a short name will be added for the index_set command/AF: index_set, ixs

The build script file for this change proposal is shown below. The .bind file is changing to add the ixs short name for index_set.

```
Description:
 A) Rewrite code in index_set to support binary/octal/hex input integers, and
    permit -octal and -hexadecimal output values with trailing radix indicators.
    Decimal input/output remains the default.  Add control arguments permitting
    free-form set specifications.
 B) Revise index_set.info to describe the code revisions.

Installation_directory:  >udd>m>gd>w>ixs;

 Build_script:     ixs.mb;

 Bound_obj:        bound_full_cp_              IN: sss  UPDATE;
     bindfile:     bound_full_cp_.bind          REPLACE;
     source:       index_set.pl1                REPLACE            compiler: pl1 -ot;

 Info:             index_set.info              IN: sss.info  REPLACE;
                     add_name:  ixs.info;
```

Testing

The index_set.pl1 program was heavily revised in code sections dealing with processing input arguments and producing output values. Processing of input arguments still generates the triplet of FIRST BOUND INCREMENT values defining each set. The algorithm for expanding such triplet specs into a set of integer elements is identical between original and revised index_set programs. Therefore, testing focused on:

- Checks that various combinations of input integers and control arguments generate the expected set specification triplet.
- Checks that decimal integers are properly output for each generated set element; or that integers of correct base are generated with appropriate radix indicator when -octal or -hex were given.
- Checks that -unspec output format works properly for both command and active function output.
- Two types of set specifications are possible: those in which FIRST <= BOUND; and those in which FIRST > BOUND. For both types, testing ensured that:
    - correct triplet type is identified;
    - sign of INCREMENT is adjusted as needed for that triplet type;
    - correct limits are imposed on triplet values; and
    - correct elements are then generated by that triplet type.

Biggest testing challenge was to ensure that all combinations of set specification arguments are properly producing a correct set specification triplet. A hidden -debug 1 control argument was added to display each set number and specification triplet before generating the output set from that specification. 16 different combinations are identified for set specification controls. Code handling each combination was verified to generate the correct set spec triplet; or to report the correct error message describing an incorrect combination of arguments.

All output bases and -unspec format were tested to ensure correct output strings are displayed/returned, with values having a correct radix indicator attached to non-decimal integer strings.

Documentation

Main documentation for index_set is its info segment. That segment has been rewritten as part of this proposal.

```
help index_set -all

>user_dir_dir>Multics>GDixon>w>ixs>index_set.info   (176 lines in info)

2021-10-15  index_set, ixs

Syntax as a command:  ixs SET1 ... -set SETn  {-control_arg}

Syntax as an active function:  [ixs SET1 ... -set SETn  {-control_arg}]

Function:  returns one or more integers, separated from each other by
spaces.  Output integers have fixed(35, 0) precision and may be
expressed as decimal, octal, or hexadecimal character strings.
Desired integers are selected by giving a SET specification.

Arguments:
   If several sets are given without -set control arguments, each
   specification includes all three arguments below.  If only one set
   is specified, FIRST and INCREMENT are optional.
FIRST
   a starting integer for the SET.   (default: 1)
BOUND
   an ending integer BOUND for the SET.
INCREMENT
   gives the increment between set elements.  Elements of the SET
   descend in value if a negative increment is given.
   If FIRST > BOUND, the INCREMENT is automatically negated if
   needed.  (default: 1)

   See "Examples" below for use of the various specification formats.
   See also: "Notes on set specification".

Control arguments:
-set BOUND
   gives a SET specification consisting of only a bound integer.
   Defaults are used for the first element of the SET, and for the
   increment between SET elements.
-set FIRST BOUND
   gives a SET specification consisting of a first element and bound.
   The default is used for the increment between SET elements.
-set FIRST BOUND INCREMENT
   gives a complete SET specification: first element, bound (last
   element), and increment between elements.
```

```
-set -specification_control_args...
   starts a new set specification.  It may be omitted for the first
   set specification.  It is followed by one or more of the set
   specification control arguments in the next section.  These may
   appear in any order, but must include either -to BOUND or
   -for COUNT to specify the length of the set.

Control arguments (set specification):
   The following control arguments may be used to specify elements
   desired in the SET.  Either -to or -for must be given for each SET
   to specify its length.
-from FIRST, -fm FIRST
   gives the first element of a SET.  (default: FIRST=1)
-by INCREMENT
   gives the increment between SET elements.  Elements of the SET
   descend in value if a negative increment is given.  If
   FIRST > BOUND and INCREMENT > 0, then INCREMENT is automatically
   negated.  (default: INCREMENT=1)

-to BOUND
   gives the bound (ending integer) in the range spanned by the SET.
   If an increment is given, it should be positive.  If
   FIRST > BOUND, the INCREMENT will automatically be negated to
   generate descending integers.
-for COUNT
   gives the count of elements desired in the set.  Specify a negative
   increment to produce a set of descending integers.  Both -to and
   -for may not appear in the same SET specification.

Control arguments (output radix):
   One of the following control arguments may be given to specify
   radix (numeric base) used when creating the output integers.
   Non-decimal output integers have the appropriate radix indicator
   appended to the number.
-octal, -oct
   output octal integer character strings.
-decimal, -dec
   output decimal integer character strings. (default)
-hexadecimal, -hex
   output hexadecimal integer character strings.
-unspec
   output each fixed bin(35) element of the SET as an unsigned
   12-digit octal value: the bits from unspec(element).

Notes on input integers:
A radix indicator may be appended to any input integer to specify its
numeric base.  Decimal is assumed if no radix indicator is given.

   INDICATOR   NUMERIC BASE              EXAMPLES
    b or _b      base  2 or binary         1101b
    o or _o      base  8 or octal          21o
    d or _d      base 10 or decimal        17d or 17
    x or _x      base 16 or hexadecimal    11x
   rN or _rN     where N is base value
                  between 2 and 16         44r5  (=24d)

A negative value may be given using a leading minus (-) character:
   -23
```

```
For octal input, a negative value may also be given using a 12-digit
octal format in which the left-most digit is 5, 6, or 7:
   -1o is same as 777777777777o

Digits for bases 11 through 16 may be entered using letters for digit
values beyond 9:
   0 1 2 3 4 5 6 7 8 9 A B C D E F
with letters accepted in either uppercase or lowercase.

For hexadecimal input, a negative value may also be given using a
9-digit hex format in which the left-most digit is: 8, 9, A, B, C, D,
E, or F:
   -2x is same as FFFFFFFFEx

Notes on set specification:
The prior version of index_set accepted a simpler SET specification
consisting of 3 integers without control arguments:
  F  gives the FIRST (or starting) element of the SET.
  B  gives the BOUND (or final) element of the SET.
  I  gives the INCREMENT between SET elements.
This simpler specification is accepted only if no control arguments
are used.

If only one SET is requested, the specification can be given as 1, 2,
or 3 integers in the following order:
    ORDER       MEANING
      B        all integers in range from 1 through B
   F  B        all integers in range from F through B
   F  B  I     every Ith integer in range from F through B

If n SET specifications are given and control arguments are not used,
each specification must contain all three integers:
   F1  B1  I1  ...  Fn  Bn  In

Examples:  Both integer-only SET specifications and specifications
using control arguments are shown below.

  ixs 7
  1 2 3 4 5 6 7

  ixs 3 7
  3 4 5 6 7

  ixs 7 3 2
  7 5 3

  ixs -set 7 3 2
  7 5 3

  ixs -set -from 7 -to 3 -by 2
  7 5 3

  ixs 3 -for 7
  3 4 5 6 7 8 9

  ixs 337o -for 10
  223 224 225 226 227 228 229 230 231 232
```

```
  ixs 337o -for 10 -octal
  337o 340o 341o 342o 343o 344o 345o 346o 347o 350o

  ixs 337o 350o -oct
  337o 340o 341o 342o 343o 344o 345o 346o 347o 350o

The following example shows three SETs defined in one command.
  ixs 5   -set -to 2 -from 7   -set 1 -to -5 -by 2
  1 2 3 4 5

  7 6 5 4 3 2

  1 -1 -3 -5

The following example shows -unspec output format.
  ixs 337o 350o -unspec
  000000000337 000000000340 000000000341 000000000342
  000000000343 000000000344 000000000345 000000000346
  000000000347 000000000350
```

Version History

Date	Revision	Author	Comment
2021-10-29	1.0	Gary Dixon	Initial draft of this MCR.