

Subject: MCR10056, 4-digit Year in Info Segment Dates
Author: Gary Dixon
Date: 2019-05-28

Multics defect Ticket #89: <http://multics-trac.swenson.org/ticket/89> discusses a longer-range issue with dates in documents like .info segments, dates which typically include only a 2-digit year. Users may find such 2-digit year numbers ambiguous; the meaning of such year numbers is not immediately apparent.

`convert_date_to_binary_` interprets such year numbers using the rule: if the 2-digit year number ≥ 70 , then use "19" as a prefix for the 2-digit year in the date; otherwise, use "20" as the prefix. Thus, 01/01/70 is interpreted as 01/01/1970; 99-05-01 becomes 1999-05-01; and 16-08-11 becomes 2016-08-11.

To avoid such ambiguity in info segments, a change is needed in the help command to display dates in an info segment in 4-digit year format. For info segments created or changed in the future, the `validate_info_seg` command should report an error if it finds 2-digit year numbers in any title line date.

Proposed Changes

Changes to the help command:

This MCR proposes to change the help command to display all dates in info segment title lines using a 4-digit year format. This can be accomplished without changing existing info segments. Instead, help will extract the date item from the title line, apply `convert_date_to_binary_` rules to this date string, then use `date_time_$format` to obtain a 4-digit date string, and replace the date item displayed in the title line. The 4-digit date format used for this display will be the standard:

```
iso_date: ^9999yc-^99my-^99dm
```

which returns a date starting with a 4-digit year-of-calendar, followed by a 2-digit month-of-year, and a 2-digit day-of-month. For example, 02/03/80 becomes 1980-02-03.

The `iso_date` format, which is one of the standard keywords accepted by `date_time_$format`, was selected because this format is recommended for worldwide usage by the International Standards Organization. The keyword `iso_date` stems from this recommendation.

Changes to `validate_info_seg`:

To avoid date ambiguity in any info segments changed or created in the future, the `validate_info_seg` (`vis`) command should require that date items in titles be specified in `iso_date` format. `vis` will be changed to report a severity 3 error for each date item in a title

line not in iso_date format. The error message will ask the developer to change the date from the string given in the title line to a corrected iso_date form of that date.

Details of the Changes:

The help command and its supporting help_ subroutine use the help_rql_.pl1 module to actually display an info segment. This module parses data within a single info segment, and responds to user requests to select and display particular help data. It prompts for the next help request, and processes each request to display help information.

As it parses a title line, this module will change the date item to be in iso_date format. These updated title lines are then displayed based upon the user requests.

A problem in handling longer help request names was also corrected. This problem was found during testing of the date changes.

help_rql_.pl1 is part of >sss>bound_info_rtns_, a bound segment that includes the help command, help_ subroutine, list_help, check_info_segs, etc.

The actual change proposed for help_rql_.pl1 is shown below:

Target bound seg: bound_info_rtns_

SOURCE FILES:

pl1: help_rql_.pl1

EXTRA segs: ioa_.info

compare_ascii [lpn help_rql_.pl1] ==

Inserted in B:

```
B326          date_time_$format
B327          entry (char(*), fixed bin(71), char(*), char(*))
returns(char(250) var),
Preceding:
```

```
A323          get_line_length_$switch
```

```
A3218         operation      char (12) varying,
```

Changed by B to:

```
B3223         operation      char (24) varying, /* maxlength must be longer than
longest defined_op */
```

Inserted in B:

```
B3817          else do;                /* Replace stored date with iso_date */
B3818          Deps_.linfo (j).header = substr (Deps_.linfo (j).header,
i+1);
B3819          Deps_.linfo (j).header =
B3820          date_time_$format ("iso_date", Deps_.linfo (j).date, "",
"") || " " ||
B3821          ltrim (Deps_.linfo (j).header);
B3822          end;
Preceding:
A3812          end;
```

The `validate_info_seg` command will be changed to convert each date item in a title line to `iso_date` format, and compare that result with the original date item. If they differ, a new severity 3 error will report that an `iso_date` is required. The error will include the date as stored in the title, and the preferred `iso_date` form for that date. The user can then correct the error as appropriate.

An additional issue was found in `validate_info_seg` while preparing this change. Recent Multics Releases have included errata info segments describing changes needed in Multics manuals. Format of these errata files varies widely, depending upon the changes they describe. As such, `validate_info_seg` should treat them as general information content. The convention for naming these files is: `MANUAL_ID.errata.info`. Therefore, I propose adding `.errata.info` to the suffix list used by `validate_info_seg` to identify general content info segments.

`validate_info_seg.pl1` is part of `>tools>bound_lib_utility_`.

The actual change proposed in this code is shown below:

Target bound seg: `bound_lib_utility_`

SOURCE FILES:

pl1: `validate_info_seg.pl1`

EXTRA segs: `validate_info_seg.info date_and_time.gi.info an82.errata.info`

add names: `an validate_info_seg.info vis.info`

add names: `an an82.errata.info errata.an82.info Standards_SDN.info`

`compare_ascii [lpn validate_info_seg.pl1] ==`

Inserted in B:

B153

B154 `need_iso_date init ("3Date must be in iso_date format^[: change ""^a"" to ""^a""^]"),`

Preceding:

A150 `/**** SEVERITY 4 messages *****/`

A337 `need_usage, no_entries, no_entry, no_hdr_name, no_sections,`

Changed by B to:

B342 `need_iso_date, need_usage, no_entries, no_entry, no_hdr_name, no_sections,`

A488

Changed by B to:

B493 `dcl iso_date char (10);`

A823 `then dtem_string = date_time_$format ("date", cv_fstime_
((status_branch.dtem)), "", "");`

Changed by B to:

B828 `then dtem_string = date_time_$format ("iso_date", cv_fstime_
((status_branch.dtem)), "", "");`

Inserted in B:

B1558 | (index (string (current_names), ".errata.info") ^= 0)

Preceding:

A1553 | (index (string (current_names), ".error.info") ^= 0)

```
A1668      if db_sw & (code = 0)
A1669      then call ioa_ ("-Entry: ^a", date_time_$format ("date_time",
entry_clock, "", ""));
A1670      if code ^= 0
A1671      then call ERR_MSG (local.bad_date, 0, bad_date, entry_date, 0, 0);
A1672      else if header_date_sw
A1673      then if (header_clock < entry_clock)
A1674      then call ERR_MSG3 (local.entry_too_late, 0, entry_too_late);
```

Changed by B to:

```
B1673      if code = 0
B1674      then do;
B1675      iso_date = date_time_$format ("iso_date", entry_clock, "",
"");
B1676      if db_sw
B1677      then call ioa_ ("-Entry: ^a", iso_date);
B1678      if header_date_sw
B1679      then if (header_clock < entry_clock)
B1680      then call ERR_MSG3 (local.entry_too_late, 0,
entry_too_late);
B1681      if entry_date ^= iso_date
B1682      then call ERR_MSG5 (local.need_iso_date, 0, need_iso_date,
entry_date, iso_date);
B1683      end;
B1684      else call ERR_MSG (local.bad_date, 0, bad_date, entry_date, 0, 0);
```

```
A1679      if db_sw & (code = 0)
A1680      then call ioa_ ("-Header: ^a", date_time_$format ("date_time",
header_clock, "", ""));
A1681      if code ^= 0
A1682      then do;
A1683      call ERR_MSG (local.bad_date, 0, bad_date, header_date, 0, 0);
A1684      header_date_sw = "b";          /* don't compare against bad date
*/
A1685      end;
```

Changed by B to:

```
B1689      if code = 0
B1690      then do;
B1691      iso_date = date_time_$format ("iso_date", header_clock, "",
"");
B1692      if db_sw
B1693      then call ioa_ ("-Header: ^a", iso_date);
B1694      if header_date ^= iso_date
B1695      then call ERR_MSG5 (local.need_iso_date, 0, need_iso_date,
header_date, iso_date);
B1696      end;
B1697      else call ERR_MSG (local.bad_date, 0, bad_date, header_date, 0, 0);
```

Inserted in B:

```
B2247      ERR_MSG5: entry (err_ct, lino, msg, ch_arg1, ch_arg2);
B2248          argct = 5;
B2249          goto common;
```

Preceding:

```
A2235      ERR_MSG3: entry (err_ct, lino, msg);
```

Inserted in B:

```
B2257      ch_arg1      char (*),          /* character value
*/
B2258      ch_arg2      char (*),          /* character value
*/
```

Preceding:

```
A2242      fb_arg1      fixed bin (24),     /* binary value
*/
```

Inserted in B:

```
B2285          else if (argct = 5)
B2286          then call ioa_ (substr (msg, 2), T, ch_arg1, ch_arg2);
Preceding:
A2268          else call ioa_ (substr (msg, 2), T, ch_arg, fb_arg1, fb_arg2);
```

In addition, while testing this change, several problems were found in command, subroutine, and general information info segs: validate_info_seg.info, date_and_time.gi.info, and ioa_.info. All of these files reside in >doc>info directory.

N.B.: vis output below includes the proposed check for 4-digit dates.

```
vis >doc>info>vis.info
```

```
>doc>info>vis.info
```

```
-----COMMAND/ACTIVE FUNCTION info
```

```
LINE: 1      85-10-10      validate_info_seg, vis
Severity 3. Date must be in iso_date format: change "85-10-10" to "1985-10-10"
LINE: 3      Syntax as a command
LINE: 6      Syntax as an active function
LINE: 9      Function
LINE: 14     Arguments
LINE: 20     Control arguments
LINE: 34     Notes on validation
```

```
cpa vis.archive::"validate_info_seg.info" -orig >doc>is>validate_info_seg.info -he
A >doc>is>validate_info_seg.info (original)
B >user_dir_dir>Multics>GDixon>work>vis>vis.archive::validate_info_seg.info (new)
```

```
A1      85-10-10  validate_info_seg, vis
Changed by B to:
B1      1985-10-10  validate_info_seg, vis
```

```
Comparison finished: 1 difference, 2 lines.
```

Note that the date in vis.info was not updated to a 2019 date, because information content in this file is not changing; only the format of its title date is changing. The original date value is preserved so check_info_segs does not show the validate_info_seg interface as being changed.

```
vis >doc>info>date_and_time.gi.info
```

```
>doc>info>date_and_time.gi.info
```

```
-----GENERAL info
```

```
LINE: 1      03/08/85  Date/Time Commands/AFs
```

```
Severity 3. Date must be in iso_date format: change "03/08/85" to "1985-03-08"
```

```
LINE: 3      <untitled>
```

```
cpa vis.archive::"date_and_time.gi.info" -orig >doc>is>date_and_time.gi.info -he
```

```
A >doc>is>date_and_time.gi.info (original)
```

```
B >user_dir_dir>Multics>GDixon>work>vis>vis.archive::date_and_time.gi.info (new)
```

```
A1      03/08/85  Date/Time Commands/AFs
```

```
A2
```

```
A3      These commands/active functions return information about dates and
A4      times:
```

```
Changed by B to:
```

```
B1      2019-05-23  Date/Time Commands/AFs
```

```
B2
```

```
B3      The following commands/active functions return information about dates
B4      and times.
```

```
A17      hour
```

```
A18
```

```
A19
```

```
Changed by B to:
```

```
B17
```

```
B18
```

```
B19      hour
```

```
A26      set_time_default, sdt
```

```
Changed by B to:
```

```
B26      set_time_default, std
```

```
Comparison finished: 3 differences, 16 lines.
```

For date_and_time.gi.info, the date is updated to 2019 because a bug in the information content of the file is being repaired (short name of set_time_default was corrected).

```
vis >doc>info>ioa_.info
```

```
>doc>info>ioa_.info
```

```
-----SUBROUTINE w/entries info
```

```
LINE: 1      09/23/85      ioa_
```

```
Severity 3. Date must be in iso_date format: change "09/23/85" to "1985-09-23"
```

```
LINE: 3      <untitled>
```

```
LINE: 28      Notes on Control Strings
Severity 1. This form is preferred:
           Notes on control strings
Severity 3. This section does not belong here.
LINE: 36      Entry points in ioa_
Severity 2. These lines end in white space:  28,  32

-----SUBROUTINE entry info
LINE: 40      :Entry: ioa_: nnl:  09/10 81  ioa_, ioa_$nnl
Severity 3. Date must be in iso_date format: change "09/10" to "1901-09-10"
LINE: 43      Function
LINE: 48      Syntax
Severity 2. Name "ioa_, ioa_$nnl" not present in Syntax line.
LINE: 52      Arguments
Severity 2. These lines end in white space:  54,  55,  58,  59

-----SUBROUTINE entry info
LINE: 63      :Entry: general_rs: 09/10/81 ioa_$general_rs
Severity 3. Date must be in iso_date format: change "09/10/81" to "1981-09-10"
LINE: 66      Function
LINE: 71      Syntax
LINE: 78      Arguments
Severity 2, line 78. Paragraph size (16 lines) exceeds limit of 15.
Severity 2. These lines end in white space:  74,  80,  83,  92,  97

-----SUBROUTINE entry info
LINE: 108     :Entry: general_rs_control_string: 08/29/85 ioa_$general_rs_control_string
Severity 3. Date must be in iso_date format: change "08/29/85" to "1985-08-29"
LINE: 111     Function
LINE: 116     Syntax
LINE: 124     Arguments

-----SUBROUTINE entry info
LINE: 153     :Entry: ioa_stream: ioa_stream_nnl:  09/10/81 ioa_$ioa_stream,
ioa_$ioa_stream_nnl
Severity 3. Date must be in iso_date format: change "09/10/81" to "1981-09-10"
LINE: 156     Function
LINE: 161     Syntax
Severity 2. Name "ioa_$ioa_stream_nnl" not present in Syntax line.
LINE: 165     Arguments
Severity 2. These lines end in white space:  167,  170,  171,  174

-----SUBROUTINE entry info
LINE: 179     :Entry: ioa_switch: ioa_switch_nnl:  09/10/81 ioa_$ioa_switch,
ioa_$ioa_switch_nnl
Severity 3. Date must be in iso_date format: change "09/10/81" to "1981-09-10"
LINE: 182     Function
LINE: 189     Syntax
Severity 2. Name "ioa_$ioa_switch_nnl" not present in Syntax line.
LINE: 193     Arguments
Severity 2. These lines end in white space:  197,  198,  202

-----SUBROUTINE entry info
LINE: 206     :Entry: rs: rsnl: rsnp: rsnpnl:  09/10/81 ioa_$rs, ioa_$rsnl,
ioa_$rsnp, ioa_$rsnpnl
Severity 3. Date must be in iso_date format: change "09/10/81" to "1981-09-10"
LINE: 208     Function
LINE: 217     Syntax
```

```

Severity 2. Name "ioa_$rsnnl, ioa_$rsnp, ioa_$rsnpnnl" not present in Syntax line.
LINE: 221  Arguments
LINE: 237  Notes
Severity 3. These lines of section exceed 71 chars:  239,  240,  241,  242
Severity 2. These lines end in white space:  223,  224,  227

```

The compare_ascii output for ioa_.info changes to correct the above errors is too voluminous to show here; but all changes are trivial in nature (as described by the errors above).

Testing

Changes to validate_info_seg and help were tested against typical command, subroutine, and general information info segs. The results with the changed vis command are shown above. Notice the severity 3 error messages reporting that "Date must be in iso_date format: ...".

Changes to help were tested against the same command, subroutine, and general information info segs. The following help output demonstrates the change in date items for the main title line of ioa_.info, and for each of the entrypoint titles:

```

help -pn >doc>info>ioa_.info
>doc>info>ioa_.info
(10 lines follow, 27 in introduction;  222 lines, 6 entry points in info)
1985-09-23  ioa_

```

The ioa_ subroutine is used for formatting a character string from fixed-point numbers, floating-point numbers, character strings, bit strings, and pointers. The character string is constructed according to the control characters entered in a "control string" and a variable list of arguments that are either edited into the output string in character form, or are used in some way to control the formatting of the string. The entire procedure is similar to formatted output in PL/I or FORTRAN.

7 more lines. More help? list_entry_points

```

Entry points in ioa_:
1981-09-10  ioa_,                      1981-09-10  ioa_$ioa_switch,
            ioa_$nnl                  ioa_$ioa_switch_nnl
1981-09-10  ioa_$general_rs           1981-09-10  ioa_$rs,
1985-08-29  ioa_$general_rs_control_string  ioa_$rsnnl, ioa_$rsnp,
1981-09-10  ioa_$ioa_stream,          ioa_$rsnpnnl
            ioa_$ioa_stream_nnl

```

```

Returning to: 1985-09-23  ioa_
UNTITLED (7 lines): More help?  q

```

The list_entry_points request of help was the request that failed to work, because the operation variable in which it was stored for comparison had a small maxlength. (See the cpa output for help_rql_pl1 for the change to the operation variable.

Documentation

No interfaces are changing, so no user documentation updates are needed for help.info. Since the validate_info_seg command is changing, its associated validate_info_seg.info will have 4-digit dates stored in its title lines. This change was shown above.

However, this MCR proposes a change to the format used for all dates in info segment titles. This change should be described in Standards System Designer's Notebook (AN82-00). The following ak82.errata.info will be added as part of this change.

```
pr vis::an82.errata.info
```

```
2019-05-25  Errata for Standards Manual
```

```
Function: This info segment provides errata and addenda information
for the Multics manual AN82-00 titled:
```

```
STANDARDS
SYSTEM DESIGN'S NOTEBOOK
```

```
2019-05-25  Information for MR 12.7:
```

```
pg 11-2 under "SYNTAX OF INFO SEGMENTS"
```

```
Replace the "Title" paragraph with the following:
```

```
Title
```

```
Some rules apply to all three types of info segments. The
first line in each info segment (or entrypoint description within
a subroutine info segment) must be a brief title line, beginning
with the date of the last modification. The title should be
appropriate for a table of contents; for command or subroutine
descriptions it will give the name(s) of the program including
abbreviations. The date should be in iso_date format.
```

```
Examples of title lines:
```

```
2019-05-24  AN82 Errata for Standards Manual
1985-10-10  validate_info_seg, vis

1985-09-23  ioa_
...
:Entry: ioa_: nnl: 1981-09-19  ioa_, ioa_$nnl
```

The last example above shows titles for a subroutine with several entrypoints. The first title line uses the standard title format for the introductory section of the info segment; this section describes the purpose and common characteristics of all the subroutine's entrypoints. Subsequent sections describe each

entrypoint or group of entrypoints sharing a common calling sequence. Each section begins with an :Entry: title line naming the entrypoint(s) followed by a typical title line with iso_date and the full subroutine name(s) described by that section. The :Entry: and colon-separated names allow the help command to skip to the description of a particular entrypoint, and to produce output for its list_entry_points request.

The info_seg.gi.info segment also documents info seg format, in more detail than the Standards SDN manual. This info seg will be changed to document use of iso_date format in header lines. Other minor errors found by validate_info_seg will be corrected at the same time.

```
vis [lpn info_seg.gi.info -lb info]
```

```
>doc>is>info_seg.gi.info
```

```
-----GENERAL info
```

```
LINE: 1      04/12/85  Info Segment Standards
Severity 3. Date must be in iso_date format: change "04/12/85" to "1985-04-12"
LINE: 3      Introduction
LINE: 8      The following topics are addressed in th
Severity 4. Section not preceded by 2 blank lines.
LINE: 15     Notes
Severity 4. Section not preceded by 2 blank lines.
LINE: 19     Types of info segs
LINE: 34     Basic info seg format
LINE: 47     Paragraphs
LINE: 57     Header line
LINE: 72     Sections
LINE: 77     Section titles
Severity 4. Section not preceded by 2 blank lines.
LINE: 88     Using help
LINE: 105    List of info seg organization techniques
LINE: 119    Using standard section titles
LINE: 123    Command/active function info segs
Severity 4. Section not preceded by 2 blank lines.
LINE: 132    List of standard headings for commands/a
LINE: 252    Using multiple info writeups
LINE: 257    Multiple-entry info segs
LINE: 274    Using subroutine entry points
LINE: 279    Subroutine infos
LINE: 301    List of standard headings for subroutine
LINE: 318    Info naming conventions
Severity 3. These lines of section exceed 71 chars: 174
Severity 2. These blank lines contain white space: 194, 196, 199
Severity 2. These lines end in white space: 74, 89, 100, 101, 114, 139, 142,
198, 206
```

cpa [lpn info_seg.gi.info -lb info] ==

A1 04/12/85 Info Segment Standards

Changed by B to:

B1 2019-05-28 Info Segment Standards

A58 An info seg begins with a heading line, consisting of a date on which
 A59 it was last modified and a brief title. Change the date whenever you
 A60 alter any part of the info seg; the format of the date is mm/dd/yy.
 A61 Follow it with two spaces and the long (and short, if any) name of the
 A62 module being described; for example,
 A63 07/17/84 working_dir, wd

Changed by B to:

B58 An info seg begins with a header line, consisting of a date on which
 B59 it was last modified and a brief title.

B60

B61 Change the date whenever you alter any part of the info seg. The date
 B62 is in iso_date format. Type "date_time_\$format -ca iso_date" for a
 B63 description of this format, or see sample header lines below.

B64

B65 Follow the date with two spaces and the long (and short, if any)
 B66 name of the module being described; for example,

B67

B68 1985-10-10 validate_info_seg, vis

B69 1985-09-23 ioa_

B70

A66 The header line on a general information (GI, gi) info seg is the date

Changed by B to:

B73

B74 The header line of a general information (.gi) info seg is the date

A69 07/20/84 The Multics Star and Equal Conventions.

Changed by B to:

B77

B78 1986-07-30 Star and Equal Conventions (general information)

A74 paragraphs which follow. Any number of paragraphs can comprise a

Changed by B to:

B83 paragraphs which follow. Any number of paragraphs can comprise a

A89 Information, from the info segs, can be displayed via help in

Changed by B to:

B98 Information, from the info segs, can be displayed via help in

A100 of help. In order to display the data obtained in a meaningful

A101 manner some suggested organizational techniques and some rules for

Changed by B to:

B109 of help. In order to display the data obtained in a meaningful

B110 manner some suggested organizational techniques and some rules for

A114 separates information for subroutine entrypoints to allow
 Changed by B to:
 B123 separates information for subroutine entrypoints to allow

A139 optional arguments and control arguments are given in the Syntax
 Changed by B to:
 B148 optional arguments and control arguments are given in the Syntax

A142 braces {}). The actual names of arguments and control arguments
 Changed by B to:
 B151 braces {}). The actual names of arguments and control arguments

A174 Function: sends a message to another user or prints messages in any
 A175 mailbox to which you have sufficient access.
 Changed by B to:
 B183 Function: sends a message to another user or prints messages in
 B184 any mailbox to which you have sufficient access.

A194
 A195 -brief, -bf
 A196
 A197 -brief,
 A198 -bf
 A199
 Changed by B to:
 B203
 B204 -brief, -bf
 B205
 B206 -brief,
 B207 -bf
 B208

A206 argument has a lengthy description, cross-reference it to the
 Changed by B to:
 B215 argument has a lengthy description, cross-reference it to the

A263 :Info: <long_name>: <short_name>: <date> <title>
 A264 Precede this line by two blank lines; leave two spaces after <date>.
 A265 The <long_name> and short_name> names are also add names on the
 A266 segment itself, with a suffix of info; the info suffix does not
 Changed by B to:
 B272 :Info: <long_name>: <short_name>: <iso_date> <title>
 B273 Precede this line by two blank lines; leave two spaces after
 B274 <iso_date>. The <long_name> and short_name> names are also add names
 B275 on the segment itself, with a suffix of info; the info suffix does not

A284 07/20/84 command_query_
 Changed by B to:
 B293 1984-07-20 command_query_

A291 :Entry: command_query_: 07/20/84 command_query_\$command_query_
Changed by B to:
B300 :Entry: command_query_: 1984-07-20 command_query_\$command_query_

Comparison finished: 16 differences, 71 lines.

Version History

Date	Revision	Author	Comment
2019-05-25	1.0	Gary Dixon	Initial revision of this MCR.
2019-05-26	1.1	Gary Dixon	Modify MCR based upon Documentation changes suggested by Eric Swenson.
2019-05-28	1.2	Gary Dixon	Modify info_seg.gi.info to require iso_date format in info seg header lines.