User Interface - Bug #7515

FILL-IN: editing dates issues

07/11/2023 04:49 PM - Vladimir Tsichevski

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

Related issues:

Related to User Interface - Bug #7642: Assigning SESSION:DATE-FORMAT has no e...	**New**
Related to User Interface - Bug #7646: FILL-IN: NPE during the PASTE of non-t...	**WIP**
Related to User Interface - Bug #7794: FILL-IN: editing numbers issues	**WIP**
Related to User Interface - Bug #8210: REGRESSION: separator should not show ...	**Test**

History

#1 - 07/11/2023 05:01 PM - Vladimir Tsichevski

At least two issues when editing date with FILL-IN.

1. When the user presses Ctrl-A to select all, then inserts a valid character (for example, 1), the selection digits must be erased, then 1 must be inserted in the beginning of the string. The resulting screen value must be 1 / /. In FWD selection is **not** erased.
2. Also, if the cursor was located in the rightmost position before the operation, the 1 digit is **not** inserted.

A simple test program (any date variable editor will do):

```
DEFINE VARIABLE d AS DATE NO-UNDO.

DEFINE BUTTON bReset LABEL "Reset values".

DEFINE FRAME f
  "Date:      " d NO-LABEL SKIP
  bReset
  WITH TITLE "#7515 test."
  .

ON CHOOSE OF bReset DO:
  d:SCREEN-VALUE = "12/29/56".
END.

APPLY 'CHOOSE' TO bReset.

ENABLE ALL WITH FRAME f.
VIEW FRAME f.
WAIT-FOR GO OF FRAME f.
```

#2 - 07/12/2023 07:19 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> In FWD selection is **not** erased.

According to the comments in the Java code, this feature is not implemented yet:

```
         // TODO: date format does not allow clearing...
```

#3 - 07/12/2023 07:34 AM - Greg Shah

Please do check ChUI for each case because the ChUI rules may have differences. Also note that ChUI has a very specific concept of insert mode that may differ from GUI.

#5 - 07/12/2023 12:20 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> 1. Also, if the cursor was located in the rightmost position before the operation, the 1 digit is **not** inserted.

This sub-issue was not confirmed in the trunk.

#6 - 07/12/2023 02:37 PM - Vladimir Tsichevski

- Status changed from New to WIP

#7 - 07/20/2023 12:37 PM - Vladimir Tsichevski

Another issue found:

In a character or date FILL-IN:

1. the cursor is in the leftmost position
2. I press the BACKSPACE key

The cursor disappears!

In OE this does not happen.

#8 - 07/24/2023 03:31 PM - Vladimir Tsichevski

Yet another issue:

In OE the SCREEN-VALUE called for an unedited date FILL-IN with unknown variable, returns / /.
In FWD the same call returns an empty string.

I am not sure how this can be fixed easily, since the screen value is calculated at the server side in the FillInWidget, and the presentation is not queried.

#9 - 07/25/2023 11:13 AM - Vladimir Tsichevski

- File 7515-9.mp4 added

Yet another issue: if a just open a date FILL-IN, place the cursor at any of the two delimiters and press /, the cursor is expected to move one character right, and nothing else should happen.

In FWD, the first date section is filled with zeroes and the cursor is positioned right from the left delimiter (regardless of the delimiter the cursor was at).

See 7515-9.mp4 for an example with the left delimiter.

#10 - 07/27/2023 08:59 AM - Vladimir Tsichevski

Yet another issue: normally pressing ? (question) should always reset screen value to an empty string.

In FWD, this does not happen if the cursor is in the last format position.

#11 - 07/28/2023 11:41 AM - Vladimir Tsichevski

Yet another issue:

Test program:

```
DEFINE VARIABLE d AS DATE NO-UNDO VIEW-AS FILL-IN FORMAT "9999/99/99".

DEFINE FRAME f d.

MESSAGE d:FORMAT.

ENABLE ALL WITH FRAME f.
VIEW FRAME f.
WAIT-FOR GO OF FRAME f.
```

In OE the value 99/99/9999 is reported (which is a bit strange). In FWD the value 9999/99/99 is reported.

#12 - 07/28/2023 03:59 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Yet another issue: normally pressing ? (question) should always reset screen value to an empty string.
>
> In FWD, this does not happen if the cursor is in the last format position.

The origin it this code in FillInGuiImpl.precessKeyEvent:

```
      if (!(dispFormat instanceof NumberFormat)            &&
          offset >= dispFormat.getScreenWidth() &&
          Keyboard.isPrintableKey(ke))
      {
         // can not add keys past the last char...
         return;
      }
```

I think, this is not a proper place for code like this. It is the format object, that has to make decisions like this. For now I will add the DateFormat to the test.

Also, we need to change the order of the first two operations in DateFormat.DateBut.input:

```
         if (presCursorPos >= getScreenWidth())
         {
            // ignore any input outside widget even '?'
            return false;
         }

         if (ch == '?')
         {
            screenValue.setLength(0);
            presCursorPos = 0;
            drawSeparators = false;

            return true;
         }
```

to let the ? be processed event at the end of the field.

#13 - 07/31/2023 07:17 AM - Vladimir Tsichevski

- Related to Bug #7642: Assigning SESSION:DATE-FORMAT has no effect on existing date instances added

#14 - 07/31/2023 07:18 AM - Vladimir Tsichevski

- Related to Bug #7646: FILL-IN: NPE during the PASTE of non-text clipboard contents added

#15 - 08/09/2023 05:47 PM - Vladimir Tsichevski

Yet another issue:

1. set SESSION:DATE-FORMAT = "ymd".
2. create an editor for a date variable with the 9999/99/99 format
3. type in the value 2000/12/29 and press TAB. The value is assigned to the variable as expected.
4. change the date format to 99999999 (same but now no delimiters)
5. The screen value is expected to change to 20001229

In FWD the value changes to 00120029 instead.

It looks like the value was formatted as yymmdddd instead of yyyymmdd.

Now if you change the format back to 9999/99/99, the screen value is correct again. That more or less proves the variable value was not changed.

#16 - 08/14/2023 04:46 PM - Vladimir Tsichevski

A date validation issue: the year value 001- is raises error 85 in OE, but is successfully parsed into value 1 in FWD. Seems, the algorithm implemented in verifyYearComponent is incorrect, the rightmost minus sign should not be trimmed silently:

```
private ErrorEntry verifyYearComponent(char[][] comp)
{
   ...
   int year = 0;
   if (hasFmtSeparator())
   {
      int end = yearBuffer.length - 1;

      // discard extreme right '-' if present
      if (yearBuffer[end] == '-')
      {
         end--;
      }
```

#17 - 08/15/2023 02:07 PM - Vladimir Tsichevski

Another issue regarding SESSION:DATE-FORMAT and existing date formats.

For all object types, compiled format instances are cached in DisplayFormatFactory.DISPLAY_FORMAT_CACHE. The cache key is composed from the object type and the format string (e.g. DATE#9999/99/99), it does not include the component order. The date component order is compiled into the cached instances. Setting this order with SESSION:DATE-FORMAT does **not** affect the cached instances, so date formatting is broken after this call.

To fix, we probably shall ad the date component order to the cache key (e.g. the above-mentioned key could now look like DATE#ymd@9999/99/99).

Note also, that changing session date component order shall cause reformatting all existing dates, which is also does not happen in FWD.

#18 - 08/16/2023 08:51 AM - Vladimir Tsichevski

Another issue: setting the date FILL-IN screen value to ///1 crashes the client with NumberFormatException:

```
Daemon Thread [Conversation [00000002:bogus]] (Suspended (exception NumberFormatException))
Integer.parseInt(String, int) line: 572
Integer.parseInt(String) line: 615
date.parseWorker(String, byte[], int) line: 4320
date.instantiateFromStringWorker(String, String, int) line: 3376
date.<init>(character) line: 766
GenericFrame.parseValue(Class<T>, character, String, boolean) line: 3775
GenericFrame.parseScreenValue(Class<T>, character, String) line: 3663
$__Proxy5(GenericFrame).setScreenValue(GenericWidget<?>, character) line: 3613
FillInWidget(GenericWidget<T>).setScreenValueNoErrorHandling(character) line: 6582
FillInWidget(GenericWidget<T>).setScreenValueInt(character) line: 6554
FillInWidget.setScreenValueInt(character) line: 990
FillInWidget(GenericWidget<T>).setScreenValue(character) line: 3348
```

#19 - 08/16/2023 09:53 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Another issue: setting the date FILL-IN screen value to ///1 crashes the client with NumberFormatException:

Found at least one error in date.parseWorker(..):

1. first all possible kinds of delimiters are replaced by the minus sign delimiter:

   ```
         String cleaner = clean.replaceAll("[./]", "-");
   ```

2. next, it is noted that a minus sign in the year component means a negative value, and is not a delimiter.

That means that a dot or a slash delimiters can convert into such a non-delimiter minus.

#20 - 08/16/2023 10:01 AM - Vladimir Tsichevski

Another thing I do not understand in in date.parseWorker(..):

```
      for (int idx = 0; idx < len; idx++)
      {
         if (cleanest.charAt(idx) == '-' &&
             (idx > 0 || idx == len - 1 || (idx != 0 && cleanest.charAt(idx + 1) != '-')))
         {
```

The condition

```
idx > 0 || idx == len - 1 || (idx != 0 && cleanest.charAt(idx + 1) != '-'))
```

is effectively reduces to

```
idx > 0 || len == 1
```

, which I suppose was not intended. Constantin, can you comment this, please?

#21 - 08/16/2023 10:40 AM - Vladimir Tsichevski

Another issue to check: for the 99999999 format, the screen values shall never be auto-completed.

Note: in trunk, it is impossible to check this, since it is impossible to enter any unnormalized value, since the delete selection feature is not implemented in trunk currently.

#22 - 08/16/2023 01:37 PM - Vladimir Tsichevski

Another (anti?)issue: date validation for date format 99999999:

In OE, any screen value, which includes a space, is invalid.
In FWD it is valid and is successfully parsed into date object.

The problem is that in OE two errors are emitted: the error 85 (invalid date input) or the error 80 (the month of a date must be from 1 to 12), and I cannot deduce the algorithm which error is emitted in which case. For example, why the problem with the month is reported event if a space is in the day component? And I need to know this if I want to implement the behavior in FWD.

The question is: should we try to fix this problem or leave this as is?

#23 - 08/16/2023 03:19 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Another (anti?)issue: date validation for date format 99999999:
>
> In OE, any screen value, which includes a space, is invalid.
> In FWD it is valid and is successfully parsed into date object.
>
> The problem is that in OE two errors are emitted: the error 85 (invalid date input) or the error 80 (the month of a date must be from 1 to 12), and I cannot deduce the algorithm which error is emitted in which case. For example, why the problem with the month is reported event if a space is in the day component? And I need to know this if I want to implement the behavior in FWD.
>
> The question is: should we try to fix this problem or leave this as is?

UPD: seems, the date parsing method for ymd component order is as follows:

1. the value is trimmed left
2. if the value length is 8, it is parsed with the original format (as yyyymmdd)
3. if the length is 7, then the rightmost character is dropped, and the result is parsed with the 999999 format (as yymmdd)
4. if the length is 6, then value is parsed with the 999999 format (as yymmdd)
5. if the length is 5, then value is parsed with the 99999 format (as yymmd)
6. if the length is 4, then value is parsed with the 9999 format (as mmdd), and the current year is used for the year.
7. if the length is 3, then value is parsed with the 999 format (as mmd), and the current year is used for the year.
8. if the length is less than 3, then the error 85 (invalid date input) is raised.

This algorithm seems to be similar to that implemented in date.addSeparators(..), according to the Javadocs for this method.

TODO: need to test the OE behavior for the mdy and dmy orders.

#24 - 08/17/2023 04:03 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Vladimir Tsichevski wrote:
>
>> Another (anti?)issue: date validation for date format 99999999:
>>
>> In OE, any screen value, which includes a space, is invalid.
>> In FWD it is valid and is successfully parsed into date object.
>>
>> The problem is that in OE two errors are emitted: the error 85 (invalid date input) or the error 80 (the month of a date must be from 1 to 12), and I cannot deduce the algorithm which error is emitted in which case. For example, why the problem with the month is reported event if a

space is in the day component? And I need to know this if I want to implement the behavior in FWD.

The question is: should we try to fix this problem or leave this as is?

UPD: seems, the date parsing method for ymd component order is as follows:

1. the value is trimmed left

UPD: the left spaces are not trimmed, but ignored, the position of date components is not shifted.

#25 - 08/17/2023 04:42 PM - Vladimir Tsichevski

The behavior of OE I cannot explain:

1. I use the 99999999 format, and ymd component order.
2. I use the following on leave trigger to convert screen value and assign to the variable:

```
ON LEAVE OF d DO:
  ASSIGN d.
  MESSAGE YEAR(d) MONTH(d) DAY(d).
END.
```

3. I enter the 0102 value and press TAB.
4. I got the Invalid date input (85) on the screen, **but**
5. I got the variable properly **assigned** as 2023/01/02.

I gather this means the following:

1. OE has different rules for verification and conversion to a date, verification rules are more strict.
2. verification error does not prevent date assignment.

Do we have to implement the same behavior in FWD?

#26 - 08/17/2023 05:19 PM - Vladimir Tsichevski

Greg, I need help.

It seems, OE uses different methods for date validation and date conversion:

Both 3010124 and 301012 values are converted to 2030/10/12 (yyyy/mm/dd), the last character (4) in the first spec is ignored.
But OE does not report errors for the first spec, and reports error 80 (the month not in the 1-12 range) for the second.

This is most probably an error in OE, so I am confused, which behavior shall we implement here?

#27 - 08/22/2023 09:18 AM - Vladimir Tsichevski

Yet another issue: the empty screen value for the 99/99/99 format must be   / /, in trunk it is   / /   (the last field filled with spaces).
There should be never trailing spaces, they are always trimmed in OE.

#28 - 08/22/2023 12:28 PM - Vladimir Tsichevski

The 7019a revision 14691 contains the following fixes:

1. #7515-1 bullet 1: missing feature implemented
2. #7515-1 bullet 2:
3. #7515-7
4. #7515-9 partly fixed
5. #7515-10
6. #7515-15
7. #7515-16
8. #7515-17
9. #7515-21
10. #7515-23
11. #7515-27

#29 - 08/22/2023 12:30 PM - Vladimir Tsichevski

All fixes were tested for the following date formats: 9999/99/99, 99/99/99, 99999999, 999999.
The component order used was ymd.
The driver was GUI Swing.

#30 - 09/11/2023 05:37 PM - Vladimir Tsichevski

- File 7019.p added

Another issue: when the user presses TAB, and focus moves from the date FILL-IN to the next tabbed widget, this widget receives the ENTRY event two times.

I suspect, the origin is how FWD handles the TAB key, when the FILL-IN screen value is incorrect.

The scenario:

1. The user types in an incorrect value into a date FILL-IN

2. The user presses TAB to move focus from the date FILL-IN.

The correct behavior is:

1. An error message is displayed.
2. On LEAVE trigger is fired for the date FILL-IN
3. On ENTRY trigger is not for for the next focusable widget
4. The date FILL-IN stays focused.

All these happen in FWD either.
From the sources I gather, that to achieve this, the following code is executed:

```
            // preliminary test for date based fill-in
            boolean forceFocusReturn = false;
            if (isDateBased && isFormatCheck() && config.appDataPres != null)
            {
               ErrorEntry[] res = config.appDataPres.validateFormat();
               forceFocusReturn = res != null && res.length > 0 && res[0].num > 0;
            }

            // the following call should be conditional

            if (groupMode)
               FocusManager.tryFocusChange(getNextAfterGroup());
            else
               ThinClient.nextTabStop(this);

            // need to return focus back to fix the fill-in screen value
            if (forceFocusReturn)
            {
               requestFocus();
            }
```

The focus is moved to the next widget and immediately returned back. The idea looks suspicious to me.

The attached 7019.p can be used to demonstrate this.

#31 - 09/12/2023 07:37 AM - Vladimir Tsichevski

Yet another issue: in 7019.p I assign the date variable on the LEAVE event of the date FILL-IN:

```
ON LEAVE OF d DO:
  MESSAGE "'" d:SCREEN-VALUE "'".
  // Note: the value is *not* normalized here yet!
  ASSIGN d NO-ERROR.
  IF NOT ERROR-STATUS:ERROR THEN
  DO:
    newValue:SCREEN-VALUE = STRING(d).
    MESSAGE YEAR(d) MONTH(d) DAY(d).
  END.
END.
```

In OE, if the date screen value is invalid, the error status is set, and the conditional block is not executed.
In FWD this block **is** executed, so the error status is most probably not set.

#32 - 09/12/2023 02:31 PM - Vladimir Tsichevski

- Related to Bug #7794: FILL-IN: editing numbers issues added

#34 - 11/18/2023 02:23 PM - Vladimir Tsichevski

Running MAJIC tests revealed at teast two regressions in ChUI in 7019a:

1. incorrect behavior of date FILL-IN for unknown date value: the gso_245 test (and many other tests): the WrkDate field expected to be blank, but it has date delimiters visible.
2. The gso_171 test: the Dist date is 01/01/-099 instead of expected 01/01/-999.

The first regression seems to be fixed in 7019a rev. 14846. The second is WIP.

#35 - 11/21/2023 04:26 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> 1. The gso_171 test: the Dist date is 01/01/-099 instead of expected 01/01/-999.

Fixed in 7019a rev. 14847: regression in formatting dates: the minimal digit numbers must be reduced by 1 for negative year values.

#37 - 01/03/2024 12:10 PM - Vladimir Tsichevski

Another minor issue:

OE: when the cursor points to a delimiter, and the user inputs the **same** delimiter character, the cursor moves one position right. Otherwise the input is ignored and beep is emitted.
FWD: any of two delimiters (which can be different) causes the cursor movement.

#38 - 01/22/2024 04:10 PM - Vladimir Tsichevski

- Related to Bug #8210: REGRESSION: separator should not show in date editing FILL-IN in ChUI when FILL-IN is cleared added

#39 - 01/30/2024 07:22 AM - Vladimir Tsichevski

Another issue: in GUI, when the user presses ?, the screen value is reset to an empty string, so the cursor is set to the leftmost position, and no cursor movement is possible.

In FWD the cursor can be moved right to position 1 with either Right-Arrow or End.

#40 - 01/30/2024 07:48 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

> Another issue: in GUI, when the user presses ?, the screen value is reset to an empty string, so the cursor is set to the leftmost position, and no cursor movement is possible.
>
> In FWD the cursor can be moved right to position 1 with either Right-Arrow or End.

Update: the following code in FillInGuiImpl:

```
      if (actionCode == Keyboard.KA_CURSOR_RIGHT)
      {
         if (offset == lastOffset)
         {
            // this needs to be treated explicitly... otherwise the DisplayFormat classes will
            // not allow the cursor to get past the last position via cursor-right key
            setCursorOffset(offset + 1);

            // the cursor will remain on the current position, but will be drawn right-side
            adjustEditScroller(Keyboard.KA_CURSOR_RIGHT, false);
            if (editScroller == rightAnchor)
            {
               editScroller.setAnchorIndex(offset + 1);
            }
            refresh();

            return;
         }
      }
```

Explicitly allow to set the cursor past the right screen value. I can not reproduce anything like this in OE GUI.

The code existed in the very first bzr version in 2014. Constantin, can you comment this? Probably, this code shall be moved to FillInImpl?

Files

7515-9.mp4	7.68 KB	07/25/2023	Vladimir Tsichevski
7019.p	5.26 KB	09/11/2023	Vladimir Tsichevski