

User Interface - Bug #7794

FILL-IN: editing numbers issues

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

Status: WIP	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	case_num:
billable: No	
vendor_id: GCD	
Description	
Related issues:	
Related to User Interface - Bug #7515: FILL-IN: editing dates issues WIP	

History

#1 - 09/12/2023 02:31 PM - Vladimir Tsichevski

- Related to Bug #7515: FILL-IN: editing dates issues added

#2 - 09/12/2023 02:33 PM - Vladimir Tsichevski

FILL-IN for an integer variable, default format, Swing: if the cursor points to a thousands delimiter, valid input is ignored. In OE the character is inserted.

#3 - 09/12/2023 02:35 PM - Vladimir Tsichevski

FILL-IN for a decimal variable, default format, Swing: if the cursor points to a thousands delimiter, an inserted digit **replaces** the digit next to the delimiter.

In OE the digit is **inserted**.

#4 - 09/13/2023 04:51 PM - Vladimir Tsichevski

Another issue:

GUI Swing, both decimal/integer editors:

In OE, a non-unknown value is padded left with spaces, so it is kind of right-aligned.

In FWD: it is not padded and is strictly left-aligned.

I wonder if there are some settings which affect the alignment which I do not know of?

#5 - 09/13/2023 05:05 PM - Greg Shah

I am remembering that some layout modes (COLON-ALIGNED?, SIDE-LABELS?) might change the alignment of the associated widget. I'm not sure if that is correct or not.

#6 - 09/14/2023 12:51 PM - Vladimir Tsichevski

Another 2 issues related to the BACKSPACE action in a decimal editor:

1. After I erase the digit 3 in the 12,345.67, the caret position is 2 instead of expected 3.
2. After I erase the digit 6 in the 12,345.67, the screen value is 12,345.07 instead of expected 12,345.70

#8 - 09/14/2023 02:24 PM - Vladimir Tsichevski

Another issues related to the DELETE action in a decimal editor:

The initial value is always 12,345.67.

1. After I erase the digit 2, the caret position is 1 instead of expected 2. This is already fixed in 7019a.
2. After I set cursor after 2 and press DELETE the caret position is 2 instead of 3. This is already fixed in 7019a.
3. After I set cursor after , and press DELETE the caret position is 2 instead of 3. This is already fixed in 7019a.
4. After I set cursor after 5 pressing DELETE has no effect.
5. After I set cursor after the dot . and press DELETE, the value is "12,345.07" instead of expected "12,345.70". This is already fixed in 7019a.
6. After I set cursor after the 7 pressing DELETE has no effect.

#9 - 09/14/2023 03:42 PM - Vladimir Tsichevski

Another issues related to the digit insertion in a decimal editor:

The initial value is always 12,345.67.

1. After inserting digits at the format positions marked as > (type over the digits 1, 2, 3 or 4 in the initial value) the caret disappears. This is already fixed in 7019a.
2. Digits can be inserted at the lowest integer part of the format (over the digit 5 in the initial value). In FWD digits are not inserted.

#10 - 09/14/2023 03:47 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

2. Digits can be inserted at the lowest integer part of the format (over the digit 5 in the initial value). In FWD digits are not inserted.

UPD: This issue exists for the >>, >>9.99 format, but does not for the >>, >99.99 format by some reasons.

#11 - 09/14/2023 04:38 PM - Vladimir Tsichevski

Yet another issue related to the digit insertion in a decimal editor:

1. The initial value has a single integer digit, which is not zero (for example, 1.00).
2. The caret is at the leftmost position (before the 1 in the example).
3. A digit is typed.

In OE, the digit typed **replaces** the digit it is over.

In FWD the new digit is **inserted** before the digit.

#12 - 09/14/2023 04:58 PM - Vladimir Tsichevski

Yet another issue which can be reproduced with 7019.p attached to the #7019 issue.

I enter -890134 into the "Value:" field and click "Set decimal". This sets the decimal editor screen value.

The error appears saying this value cannot be displayed with format displayed, and decimal value is displayed as ??????. This is expected.

Now I delete the last character in the "Value:" field, so the value is now -89013, which is now a correct value. I press "Set decimal" and got the same error referencing the **old** -890134 value, which is wrong.

I press the "Set decimal" one more time, now no error is displayed, and the value is successfully put to the decimal editor as -89,013.00.

#13 - 09/15/2023 02:58 PM - Vladimir Tsichevski

Yet another issue: decimal editor: inserting the question mark ? does not reset the editor to unknown state, if the caret is in the rightmost position.

#14 - 09/15/2023 03:23 PM - Vladimir Tsichevski

- Status changed from New to WIP

Vladimir Tsichevski wrote:

Yet another issue: decimal editor: inserting the question mark ? does not reset the editor to unknown state, if the caret is in the rightmost position.

The fix: NumberFormat, I.3620. Replace

```
if (digitsRight.length() + digitPos < 0)
{
    return false;
}
```

by

```
if (digitsRight.length() + digitPos < 0)
{
    // the '?' must be processed event at the end of
    // string
    return ch == '?' && super.input(ch);
}
```

This code disables any input at the end of string, but we must provide an exception for the ? character.

#15 - 09/15/2023 03:30 PM - Vladimir Tsichevski

Yet another issue with decimal editor: initially the screen value is filled with spaces (which is incorrect by itself), and if the user positions the caret at the rightmost position (by just clicking inside the widget, for example), the input is impossible (since the caret is in the rightmost position), which is probably very surprising for the user.

#16 - 09/18/2023 07:08 AM - Vladimir Tsichevski

Yet another issue with decimal editor:

1. insert an unknown value by typing ?
2. select the ? sign
3. press Delete

Nothing is deleted, which is expected, but in FWD the selection is removed, which is not expected.

#17 - 09/18/2023 01:53 PM - Vladimir Tsichevski

Issues related to setting screen value.

Invalid character must not be tolerated, so the following values must cause the "Invalid character in numeric input" error:

1. "a"
2. "a123"
3. "123b"
4. ";123b7".

In FWD all invalid characters are silently filtered out before the string is parsed.

#18 - 09/18/2023 02:13 PM - Vladimir Tsichevski

Navigation issue: in OE in GUI mode, the CURSOR-DOWN and CURSOR-UP keys move cursor right and left. In FWD they do not.

#19 - 09/18/2023 02:19 PM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

Navigation issue: in OE in GUI mode, the CURSOR-DOWN and CURSOR-UP keys move cursor right and left. In FWD under Linux they do not.

Also, the Ctrl-CursorLeft and Ctrl-CursorRight work like HOME and END. In FWD under Linux they do not.

I am not sure whether this issue exists in FWD under Windows, so this is probably no an issue.

#20 - 09/19/2023 09:19 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

In OE, a non-unknown value is padded left with spaces, so it is kind of right-aligned.
In FWD: it is not padded and is strictly left-aligned.

UPD: initially, the screen value is right-aligned, as expected, but entering the unknown value by typing the question mark switches the editor to right-aligned mode.

#21 - 12/20/2023 12:50 PM - Vladimir Tsichevski

Greg Shah wrote:

I am remembering that some layout modes (COLON-ALIGNED?, SIDE-LABELS?) might change the alignment of the associated widget. I'm not sure if that is correct or not.

The SIDE-LABEL attribute of frame makes FILL-IN left-aligned, the COLON-ALIGNED is used with frame contents widgets only and affect positioning widgets in frame.

#22 - 12/20/2023 01:21 PM - Vladimir Tsichevski

Known issues compiled

This entry contains all know sub-issues, and will be edited as new sub-issues emerge and existing are fixed.

Screen value formatting.

1. The [#7794-27](#) issue.

Input

Cursor outside the widget area

See [#7794-32](#).

Invalid characters not ignored

In a freshly opened FILL-IN pressing an invalid character causes the screen value **reset to the zero value**. In OE invalid character is **ignored**.

Valid characters over a thousands delimiter

1. FILL-IN for an integer variable, default format, Swing: if the cursor points to a thousands delimiter, valid input is ignored.
In OE the character is inserted.
2. FILL-IN for a decimal variable, default format, Swing: if the cursor points to a thousands delimiter, an inserted digit replaces the digit next to the delimiter.
In OE the digit is inserted.
Another issues related to the digit insertion in a decimal editor:

Disappearing caret

(FIXED in 14858)

The initial value is 12,345.67.

After inserting digits at the format positions marked as > (type over the digits 1, 2, 3 or 4 in the initial value) the caret disappears.

Digits are not inserted

The initial value is 12,345.67.

Digits can be inserted at the lowest integer part of the format (over the digit 5 in the initial value). In FWD digits are **not** inserted.

This issue exists for the >>,>>9.99 format, but does not for the >>,>99.99 format by some reasons.

Digits in empty editor are not inserted in the rightmost position

Decimal editor: initially the screen value is filled with spaces (which is incorrect by itself), and if the user positions the caret at the rightmost position (by just clicking inside the widget, for example), the input is impossible (since the caret is in the rightmost position), which is probably very surprising for the user.

Digit inserted in wrong position

1. The initial value has a single integer digit, which is not zero (for example, 1.00).
2. The caret is at the leftmost position (before the 1 in the example).
3. A digit is typed.

In OE, the digit typed replaces the digit it is over.

In FWD the new digit is inserted before the digit.

Backspace

In decimal editor:

1. erase the digit 3 in the 12,345.67, the caret position is 2 instead of expected 3.
2. erase the digit 6 in the 12,345.67, the screen value is 12,345.07 instead of expected 12,345.70

Delete

decimal editor, the initial value is 12,345.67.

1. erase the digit 2, the caret position is 1 instead of expected 2. This is already fixed in 7019a.
2. set cursor after 2 and press DELETE the caret position is 2 instead of 3. This is already fixed in 7019a.
3. set cursor after , and press DELETE the caret position is 2 instead of 3. This is already fixed in 7019a.
4. set cursor after 5 pressing DELETE has no effect.
5. set cursor after the dot . and press DELETE, the value is "12,345.07" instead of expected "12,345.70". This is already fixed in 7019a.
6. set cursor after the 7 pressing DELETE has no effect.

Decimal separator

Inserting the decimal separator character in the decimal editor does not position the cursor **next to the decimal separator** ([#7794-24](#)).

Region delete

1. the difference probably caused by a bug in OE: [#7019-92](#).

Question mark

Decimal editor: inserting the question mark ? does not reset the editor to unknown state, if the caret is in the rightmost position. See [#7794-13](#) and [#7794-29](#).

Deleting question mark

(FIXED in 14858) decimal editor:

1. insert an unknown value by typing ?
2. select the ? sign
3. press Delete

In OE nothing is deleted, which is expected, but in FWD the selection is removed, which is not expected.

h3. Setting screen value

Fixing a wrong value does not work

Yet another issue which can be reproduced with 7019.p attached to the [#7019](#) issue.

1. enter -890134 into the "Value:" field and click "Set decimal". This sets the decimal editor screen value.
2. The error appears saying this value cannot be displayed with format displayed, and decimal value is displayed as ?? This is expected.
3. delete the last character in the "Value:" field, so the value is now -89013, which is now a correct value. Press "Set decimal" and got the same

error referencing the old -890134 value, which is wrong.

4. press the "Set decimal" one more time, now no error is displayed, and the value is successfully put to the decimal editor as -89,013.00.

Passing invalid values

Invalid character must not be tolerated, so the following values must cause the "Invalid character in numeric input" error:

1. "a"
2. "a123"
3. "123b"
4. ";123b7".

In FWD all invalid characters are silently filtered out before the string is parsed.

Alignment

1. GUI Swing, both decimal/integer editors:
 1. In OE, a non-unknown value is padded left or right with a space, so right-aligned and left-aligned values always look different.
 2. In FWD: values are not padded.
2. Right-aligned editor becomes left-aligned after an unknown value once was entered to it.

Navigation

1. In OE in GUI mode, the CURSOR-DOWN and CURSOR-UP keys move cursor right and left. In FWD they do not.
2. In OE the Ctrl-CursorLeft and Ctrl-CursorRight work like HOME and END. In FWD Swing under Linux they do not.

I am not sure whether this issue exists in FWD under Windows, so this is probably not an issue.

Documentation

1. Wrong Javadocs for `NumberType.getDecimalSeparatorIndex()` (see [#7794-23](#)).

#23 - 02/05/2024 07:09 AM - Vladimir Tsichevski

May be, this task is not the proper place to report this.

The Javadocs for the return type of `NumberType.getDecimalSeparatorIndex()` is obviously incorrect (probably, was copy-pasted from some other method):

```
/**
 * Parse the numeric format string and return the digit string on
 * the right if it exists.
```

```

*
* @param    fmt
*           The format string to parse. Must be a valid
*           Progress-compatible numeric format.
*
* @return   The digit text on the right side or null
*           if no text exists there.
*/
public static int getDecimalSeparatorIndex(String fmt)
{
    return obtainParsed(fmt, null).decimalPoint;
}

```

#24 - 02/07/2024 02:32 PM - Vladimir Tsichevski

Another issue: inserting the decimal separator character in the decimal editor should position the cursor **next to the decimal separator**.

In FWD this operation has no visible effect.

#25 - 02/08/2024 05:08 PM - Vladimir Tsichevski

Alignment-related issue 2: Right-aligned editor becomes left-aligned after an unknown value once was entered to it.

The origin is in the difference between the two NumberBufGui constructors:

```

/**
 * Constructs default presentation used for clear operation.
 */
public NumberBufGui()
{
    super();
}

public NumberBufGui(NumberType var)
{
    super(var);

    // check if we have right aligned fill-in as front end
    rightAlignedGuiFillIn = editableSource instanceof FillInGuiImpl &&
        ((FillInGuiImpl)editableSource).isRightAligned();
}

```

The rightAlignedGuiFillIn is **not** initialized in the first constructor. Copy-pasting the code from the second constructor to the first solves this.

PS: I would prefer to convert rightAlignedGuiFillIn from a field to a method, so we will nether bother about the field proper initialization.

#26 - 02/09/2024 08:44 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

Alignment-related issue 2: Right-aligned editor becomes left-aligned after an unknown value once was entered to it.

The origin is in the difference between the two NumberBufGui constructors:

[...]

The rightAlignedGuiFillIn is **not** initialized in the first constructor. Copy-pasting the code from the second constructor to the first solves this.

PS: I would prefer to convert rightAlignedGuiFillIn from a field to a method, so we will nether bother about the field proper initialization.

The 7019c rev. 14979 includes the fix.

#27 - 02/09/2024 11:29 AM - Vladimir Tsichevski

Incorrect screen value formatting:

If the format string includes a minus sign character, and the FILL-IN is right-aligned, one character is always reserved for the sign, and space character is displayed there for positive numbers.

In FWD, no space is reserved in case the value is positive.

This issue exists at least in GUI.

#28 - 02/09/2024 02:38 PM - Vladimir Tsichevski

Another issue: then a (right-aligned) FILL-IN is just displayed, the screen value must be empty, hence neither cursor movements nor selection are allowed.

In FWD the visible screen value is non-empty and filled with spaces. Cursor movement is allowed, selection is not visible, but is non-empty internally. Selection deletion results in client crash due to assertion errors.

#29 - 02/12/2024 11:17 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

Yet another issue: decimal editor: inserting the question mark ? does not reset the editor to unknown state, if the caret is in the rightmost position.

More correct fix replace NumberFormat.java, l. 1169 by:

```
screen = Utils.isChUI()  
    ? StringHelper.leftAlignText("?", getScreenWidth())  
    : "?";
```

This assures the correct values for GUI and ChUI modes both.

#30 - 02/12/2024 11:30 AM - Vladimir Tsichevski

Yet another issue: clearing FILL-IN in GUI must set the screen value to ?, but in FWD the value is right-padded with spaces. Padding should occur in ChUI only.

The fix is in [#7794-29](#).

#31 - 02/12/2024 11:33 AM - Vladimir Tsichevski

Vladimir Tsichevski wrote:

Yet another issue: decimal editor: inserting the question mark ? does not reset the editor to unknown state, if the caret is in the rightmost position.

The correct fix is using the screen length instead of screen value length.

In theory, both length must always equal, but in current implementation the screen info can be longer, and the extra characters are filled with garbage. This garbage is used in some current code.

#32 - 02/20/2024 12:53 PM - Vladimir Tsichevski

- *File 7794-32.png added*

Yet another issue related to digit input at the rightmost position in ChUI mode.

In OE the cursor can be positioned right from the rightmost format position. In GUI mode this causes no problems with displaying such cursor, but in ChUI the cursor (highlighted character) would go outside the widget area. IMO because of that in OE the cursor visually stays at the last character, but logically is located right from this position: you cannot input characters in this state, and you have to move logical cursor left one position to be able to. Note: after moving the logical cursor visually nothing changes: the last character is still highlighted.

But in FWD the cursor goes outside the widget, as shown here:



Files

7794-32.png

815 Bytes

02/20/2024

Vladimir Tsichevski