

## User Interface - Bug #3584

### tooltips do not work in FWD for disabled, TEXT and FILL-IN widgets

05/25/2018 10:23 AM - Constantin Asofiei

|                        |              |                        |           |
|------------------------|--------------|------------------------|-----------|
| <b>Status:</b>         | WIP          | <b>Start date:</b>     |           |
| <b>Priority:</b>       | Normal       | <b>Due date:</b>       |           |
| <b>Assignee:</b>       | Adrian Lungu | <b>% Done:</b>         | 20%       |
| <b>Category:</b>       |              | <b>Estimated time:</b> | 0.00 hour |
| <b>Target version:</b> |              | <b>case_num:</b>       |           |
| <b>billable:</b>       | No           | <b>version:</b>        |           |
| <b>vendor_id:</b>      | GCD          |                        |           |
| <b>Description</b>     |              |                        |           |

#### History

##### #1 - 05/25/2018 10:25 AM - Constantin Asofiei

The problems are these:

1. 4GL allows tooltips for TEXT widgets, too.
2. FILL-IN, TEXT are 'containers' in FWD and can't be found by findMouseSource
3. findMouseSource will not report a disabled widget - this is incorrect, as tooltips work for disabled widgets, too

So we need to mainly fix findMouseSource, but this is used in more parts (for finding mouse event targets) than tooltip support, so the changes need to be tested carefully.

##### #3 - 06/11/2019 12:26 PM - Greg Shah

- Assignee set to Adrian Lungu

##### #4 - 08/29/2019 03:37 AM - Adrian Lungu

- Status changed from New to WIP

##### #5 - 08/29/2019 07:29 AM - Adrian Lungu

- % Done changed from 0 to 20

I focused on the next issue:

3. findMouseSource will not report a disabled widget - this is incorrect, as tooltips work for disabled widgets, too

After some investigation, it seems that findMouseSource is a method which returns a single widget that is eventually the source of the mouse event. However there is a clear example which troubles this implementation: one with disabled widgets.

If one widget is disabled, 4GL suggests that the mouse event should be considered as:

1. an event occurred over the container (i.e. a frame which holds the disabled widget)
2. an event occurred over the widget

To get an idea straight away, follow the next example in **4GL**:

```
DEFINE FRAME frm WITH CENTERED TITLE "Simple test".
```

```
DEFINE VARIABLE widget1 AS CHAR  
VIEW-AS FILL-IN  
TOOLTIP "this is a tooltip".  
  
DEFINE VARIABLE widget2 AS CHAR  
VIEW-AS FILL-IN  
TOOLTIP "this is a tooltip".  
  
DISPLAY widget1 widget2 WITH FRAME frm.  
ENABLE widget1 WITH FRAME frm.  
  
ON LEFT-MOUSE-DOWN OF FRAME frm DO:  
    MESSAGE "1".  
END.  
  
wait-for window-close of current-window.
```

By hovering then clicking on widget2 will result in a shown tooltip and a message, meaning that two widgets are aware of mouse events at the same coordinates (the frame and the FillIn).

**FWD** behavior for the same example is to set the source of the mouse event to be the frame if the widget is not enabled, thus the FillIn is not aware of any mouse event as they are all redirected to the frame. `FrameGuiImp.findMouseSource` explains: // if the widget is not enabled, then the source is the frame, too

Clear enough, if the widget is no longer checked for its enable status, the widget will be aware of all mouse events, while the frame will be the source of none.

A workaround which looks appealing at this stage, is to let all the `MOUSE_ENTERED` events pass to the widget, even if it is disabled. This solution is based on the following intuitions:

1. the frame shouldn't care about the `MOUSE_ENTERED` events, as they do not have tooltips. (still to be tested, maybe for other usage of `MOUSE_ENTERED` for the frames)
2. the widget shouldn't care of any other event apart from `MOUSE_ENTERED` as long as it is disabled. (still to be tested, maybe there are widgets which do respond to the mouse even if disabled)

Further, I will surf the documentation in order to find examples which do contradict the presumptions above.