# User Interface - Feature #3291

## implement SYSTEM-DIALOG-COLOR

04/27/2017 02:57 PM - Greg Shah

| | | | | |
|---|---|---|---|---|
| **Status:** | Closed | | **Start date:** | |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | Ovidiu Maxiniuc | | **% Done:** | 100% |
| **Category:** | | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | | |
| **billable:** | No | | **version:** | |
| **vendor_id:** | GCD | | | |

| **Description** |
|---|
| |

| **Related issues:** | |
|---|---|
| Related to User Interface - Bug #1830: implement SYSTEM-DIALOG-GET-FILE support | **Closed** |

## History

**#1 - 07/24/2017 08:51 AM - Ovidiu Maxiniuc**

*- Status changed from New to WIP*

*- Start date deleted (04/27/2017)*

Added conversion support in task branch 1830a, revision 11159.
The full-syntax construct like this:

```
SYSTEM-DIALOG COLOR color-number
   UPDATE OKpressed
   IN WINDOW CURRENT-WINDOW.
```

will be converted in java as:

```
ColorTable.createColorChooserDialog(colorNumber)
        .update(okpressed)
        .setWindow(currentWindow())
        .execute();
```

Notes:

- the static method call handles the mandatory parameters;
- each non-static method call (except execute) from the chaining represents an option that sets up the server-side configuration of the statement;
- finally, the execute method does the actual call on client side that will open the chooser dialog, based on options selected.

**#2 - 07/24/2017 08:52 AM - Greg Shah**

It looks good.  Go ahead with this approach.


**#3 - 07/24/2017 09:16 AM - Ovidiu Maxiniuc**

*- Related to Bug #1830: implement SYSTEM-DIALOG-GET-FILE support added*


**#4 - 11/01/2017 02:14 PM - Greg Shah**

The runtime implementation for this task was written in branch 1830b and was merged to trunk as revision 11188.

There are 2 known bugs to be resolved:

- Fill-ins for entering values directly are not working.
- After the the user makes a selection, the selection will be used by widgets that are subsequently created, but already existing widgets are not updated.

From Ovidiu in regard to the second issue:

> This is because we cache gc/gf resolvers, I think. If so, we need to drop these caches and always get the values from FontTable/ColorTable. If this is not the reason, then we have to implement some notification to let the widgets know the FontTable/ColorTable were altered and need to be repainted or even redo the layout - since new fonts may have different metrics. These changes may have a really big impact in all GUI code.


**#5 - 11/01/2017 02:15 PM - Greg Shah**

> After the the user makes a selection, the selection will be used by widgets that are subsequently created, but already existing widgets are not updated.

In the 4GL all existing widgets immediately honor any changes to the COLOR table?


**#6 - 11/14/2017 03:41 PM - Ovidiu Maxiniuc**

Greg Shah wrote:

> After the the user makes a selection, the selection will be used by widgets that are subsequently created, but already existing widgets are not updated.

> In the 4GL all existing widgets immediately honor any changes to the COLOR table?

I have done some investigations. It seems that the problem in the implementation of COLOR-TABLE:SET-RGB-VALUE(idx, rgb) which is used by the color chooser dialog to set the color after the user selects it.

In P4GL, the following code will instantly change the background color of the frame and its text to red/cyan:

```
DEFINE BUTTON prog-button LABEL "PROG".
FORM
    "some text" SPACE(2) prog-button
    WITH FRAME color-frame NO-LABELS TITLE "Update colors ..."
    FGCOLOR front-color BGCOLOR back-color VIEW-AS DIALOG-BOX.

ON CHOOSE OF prog-button IN FRAME color-frame DO:
     COLOR-TABLE:SET-RGB-VALUE(8, RGB-VALUE(255, 64, 64)).
     COLOR-TABLE:SET-RGB-VALUE(9, RGB-VALUE(0, 196, 196)).
END.
```

The same code in FWD will only update the the color table entry, followed by a synch operation with the client. This is not enough to change the color of the already rendered widgets.

**#7 - 11/15/2017 09:18 AM - Greg Shah**

> The same code in FWD will only update the the color table entry, followed by a synch operation with the client. This is not enough to change the color of the already rendered widgets.

Can we easily know then the color table changes on the client? If so, can we force some kind of reload of all color resolvers (and then force a redraw of all visible windows)?

**#8 - 03/23/2018 03:05 PM - Eric Faulhaber**

*- % Done changed from 0 to 100*

*- Status changed from WIP to Closed*

*- Assignee set to Ovidiu Maxiniuc*

Branch 1830c was committed to trunk as r11240.