

User Interface - Feature #1834

implement direct manipulation (drag and drop)

10/30/2012 01:25 PM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Eugenie Lyzenko	% Done:	100%
Category:		Estimated time:	80.00 hours
Target version:		vendor_id:	GCD
billable:	No		
Description			

History

#1 - 10/31/2012 02:43 PM - Greg Shah

- Target version set to Milestone 12

#2 - 04/30/2015 04:22 PM - Greg Shah

Implement runtime support for the following attributes:

MOVABLE
RESIZABLE
SELECTABLE

Conversion support for these should have already been implemented in [#2567](#).

Implement support for all direct manipulation events (both general and frame). See the ABL Reference, "Introduction to ABL events" section, "Direct manipulation events" sub-section.

You will need to explore the extent to which this support can be used to integrate with OS-level drag-n-drop facilities. Document the capabilities here and we will discuss this further.

#3 - 03/23/2016 12:55 PM - Greg Shah

- Target version deleted (Milestone 12)

#4 - 04/27/2017 02:04 PM - Greg Shah

The next set of direct manipulation support to be implemented is the following (see #3257-8):

Attributes

SELECTED (runtime)
SELECTABLE (runtime)
RESIZABLE (runtime)
MOVABLE (runtime)
BOX-SELECTABLE (frame and dialog, include the associated events START-BOX-SELECTION/END-BOX-SELECTION)
NUM-SELECTED-WIDGETS (frame, dialog, window)
DROP-TARGET (fill-in and editor)
NUM-DROPPED-FILES (editor)
GRID-SNAP (frame)

GRID-VISIBLE (frame)
GRID-FACTOR-HORIZONTAL (frame)
GRID-FACTOR-VERTICAL (frame)
GRID-UNIT-HEIGHT-PIXELS (frame)
GRID-UNIT-WIDTH-PIXELS (frame)
ROW-RESIZABLE (browse-only, include the associated events START-ROW-RESIZE/END-ROW-RESIZE)

Methods

END-FILE-DROP (editor, frame)
GET-DROPPED-FILE (editor, frame)

Options

DROP-TARGET (frame)

#5 - 05/19/2017 07:12 AM - Greg Shah

- Status changed from New to WIP
- Assignee set to Eugenie Lyzenko

#6 - 05/23/2017 04:02 PM - Eugenie Lyzenko

The testcases repo was updated to revision 1610. While working on drag-n-drop implementation we need to investigate how exactly this feature is implemented in original 4GL. The first test has been added based on official 4GL documentation.

#7 - 05/23/2017 10:41 PM - Eugenie Lyzenko

From the GUI perspective we need to:

1. Implement selection attribute drawing(selection box) for all selectable widgets.
2. Implement resize box drawing(special resizing box with 8 box like handles).
3. Implement selection/deselection operations.
4. Implement plug-able mouse drag handler for all widgets that are selectable/resizable.
5. Looks like embedded Java Drag&Drop features are useless for this task.

Continue investigation.

#8 - 05/24/2017 06:44 AM - Greg Shah

I agree with your assessment so far. On a positive note, it seems that the full range of possible data transfer capabilities are not needed, but I guess the ability to drop a text file on editor or fill-in and have the text inserted is important. Is that correct or are there other data transfer capabilities that can be used from the 4GL?

#9 - 05/24/2017 06:52 AM - Greg Shah

Hmmm. I was wrong about the list of widgets the drop operation applies to. Also, the GET-DROPPED-FILE, NUM-DROPPED-FILES and END-FILE-DROP are all about processing a list of filenames.

Are there any data transfer (e.g. text or other) capabilities and if so:

- Which widgets support them?
- What operations are possible?
- Does it vary by widget?

#10 - 05/24/2017 07:13 AM - Eugenie Lyzenko

Greg Shah wrote:

I agree with your assessment so far. On a positive note, it seems that the full range of possible data transfer capabilities are not needed, but I guess the ability to drop a text file on editor or fill-in and have the text inserted is important. Is that correct or are there other data transfer capabilities that can be used from the 4GL?

I think we can consider this separately. Meaning there is some 4GL specific drag-n-drop functionality like widget selection/moving/resizing and there is some system wide drag-n-drop functionality like inserting text from one fill-in or editor to another or from external text editor to 4GL entry field or editor or even dropping external file to fill-in or editor(as you noted).

What I think is the Progress is responsible for only Progress specific drag-n-drop. Here is the place where 4GL makes changes in 4GL message handling ordering and so on. And we need to implement this feature taking into account 4GL behavior.

And I think the "system wide" specific drag-n-drop features are inheriting by the fact the 4GL GUI on Windows uses system controls - entry fields, editors. In this case the system features for drag-n-drop just integrating to the 4GL "for free". But we have to implement them separately because the data related widgets(fill-in and editor) internals are implemented from scratch. We do not use system controls in GUI, implementing text drawing, data enter, caret positioning with our GUI primitives.

May be the best way will be to add d-n-d text related functionality to fill-in and editor separately. I mean considering this feature not with general direct manipulation context.

#11 - 05/24/2017 07:14 AM - Eugenie Lyzenko

Greg Shah wrote:

Hmmm. I was wrong about the list of widgets the drop operation applies to. Also, the GET-DROPPED-FILE, NUM-DROPPED-FILES and END-FILE-DROP are all about processing a list of filenames.

Are there any data transfer (e.g. text or other) capabilities and if so:

- Which widgets support them?
- What operations are possible?
- Does it vary by widget?

OK. I'll investigate this point.

#12 - 05/24/2017 06:10 PM - Eugenie Lyzenko

New investigation results:

1. The data drag-n-drop functionality is limited in Progress ABL. Currently it works only for external file system object. The user can select a set of files and drop them into widget with DROP-TARGET attribute set to TRUE. The default value is FALSE. There is no default reaction for this dropping. The editor widget does not insert the content of the dropped file in current position. This is the only ability for programmer to get the file names by(NUM-DROPPED-FILES, GET-DROPPED-FILE) and implement any action the developer wants to be implemented. For example procedure editor open the content. But it can be any other or nothing(by default)

2. The text selection drag-n-drop is seems to also not implemented in 4GL. I mean having two editable widget the user can not select a part of the text in one widget and drag this text into another. But it is possible to open widget context menu(right mouse button) and cut/copy/paste between two editable widgets; however this in not drag-n-drop operations, right?

So the point number 2 is not what we need to consider within this task. For now I'm not sure if there is any widget specific capabilities(like something form GET-DROPPED-FILE, NUM-DROPPED-FILES or END-FILE-DROP list might behave differently). Continue investigation.

#13 - 05/24/2017 08:47 PM - Eugenie Lyzenko

The testcases repo was updated to revision 1611.

Two new testcases have been added to illustrate drag-n-drop functionality for editor and fill-in widgets. The selected text drop functionality checked and found as not working in 4GL. Also no default actions for files drag-n-drop into editor or fill-in.

#14 - 05/25/2017 08:42 PM - Eugenie Lyzenko

After additional investigation and documentation checking I came in to the following implementation plan:

1. First stage is to implement only direct manipulation part.

- Check the conversion/runtime support for direct manipulation related attributes(SELECTED, SELECTABLE, RESIZABLE ...). Adding missing conversion/runtime support when necessary. This will make enough background for the next step.
- Prepare the mouse handler subsystem to handle widget selection/move/resize. This must include the UI drawing part specific to each operation like painting selection boxes for different modes. Make it to be abstract for widget, meaning single implementation method for all affected widgets.
- Implement direct manipulation related messages processing to complete the stage. Checking the implementation functionality as single system to find possible missing in implemented attributes or methods.

2. The second stage is to implement DnD support for file system object related methods coming from external windows.

- Check the conversion/runtime support for file list related attributes. Add missing features if necessary.
- Add the handler to keep track incoming drop events from external OS level for file list to handle.
- Implement dropped file list related attributes and methods(NUM-DROPPED-FILES, END-FILE-DROP(), GET-DROPPED-FILE()).
- Verify functionality preparing test that uses the file list to perform some actions with dropped files.

So I'm creating the 1834a branch and starting to implement.

#15 - 05/25/2017 09:46 PM - Eugenie Lyzenko

Created task branch 1834a from trunk revision 11152.

#16 - 05/26/2017 06:41 AM - Greg Shah

I like the plan.

Make it to be abstract for widget, meaning single implementation method for all affected widgets.

I agree the API should be abstract. I assume that the implementation will be done at the theme level (presumably with per-widget rendering).

2. The second stage...

- Check the conversion/runtime support for file list related attributes. Add missing features if necessary.

If stage 1 takes too long to finish, we may need you to do this part earlier so that all of these things convert properly (and are stubbed out so they compile).

#17 - 05/26/2017 09:25 AM - Eugenie Lyzenko

Greg Shah wrote:

2. The second stage...

- Check the conversion/runtime support for file list related attributes. Add missing features if necessary.

If stage 1 takes too long to finish, we may need you to do this part earlier so that all of these things convert properly (and are stubbed out so they compile).

OK. I can process all conversion/runtime support attributes or even methods(making stubs) on the first step of stage 1.

#18 - 05/26/2017 09:04 PM - Eugenie Lyzenko

The testcases repo was updated to revision 1612.

Testcase to illustrate possible usage of the file dropping feature has been added. Required to verify the respective attribute/methods conversion.

The task branch 1834a has been updated for review to revision 11153.

This is the very first step of the Drag-n-Drop support implementation. Two new interfaces have been added to define direct manipulation and file list drop functionality. The rule file changed to support conversion level. Also added respective unwrappers. Continue working with stub implementation to get compiled.

#19 - 05/26/2017 09:08 PM - Eugenie Lyzenko

Fixed previous note for committed branch revision(11153).

#20 - 05/29/2017 09:31 AM - Eugenie Lyzenko

Point to discuss with DnD inheritance to be implemented.

The widgets that must support file list dropping:
BrowseWidget, ControlEntity, PaneEntity

The widgets that supports direct manipulation:
PaneEntity(Frames and Widgets)

So I would suggest to:

1. PaneEntity to implement DirectManipulationInterface.
2. All classes BrowseWidget, ControlEntity, PaneEntity are derived from GenericWidget. However if we make GenericWidget implemented as DroppableInterface - we will have many classes inherited from GenericWidget that has no DnD functionality(field group, image, menu, menu item). So my plan is to introduce new class say DroppableWidget extending GenericWidget and change inheritance for BrowseWidget, ControlEntity, PaneEntity to extend DroppableWidget. All other widgets will extend GenericWidget. This way we separate drag-n-drop file list support within single class leaving not related widget's class hierarchy untouched.

Is this OK?

#21 - 05/29/2017 09:43 AM - Greg Shah

So my plan is to introduce new class say DroppableWidget extending GenericWidget and change inheritance for BrowseWidget, ControlEntity, PaneEntity to extend DroppableWidget. All other widgets will extend GenericWidget. This way we separate drag-n-drop file list support within single class leaving not related widget's class hierarchy untouched.

I prefer a different approach. We find many different cases where there is some arbitrary subset of widgets that implement a specific feature. It is not consistent enough to have all of it implemented purely based on the class hierarchy. The file dropping support is not a big enough feature set to cause us to rework the inheritance tree just for that feature.

Please implement it as its own interface. Try to put most/all of the logic in default methods (in the interface itself). Any stored data will need to be in the implementation classes and will have to be passed in to the shared/default worker methods in the interface.

#22 - 05/29/2017 01:04 PM - Eugenie Lyzenko

Greg Shah wrote:

I prefer a different approach. We find many different cases where there is some arbitrary subset of widgets that implement a specific feature. It is not consistent enough to have all of it implemented purely based on the class hierarchy. The file dropping support is not a big enough feature set to cause us to rework the inheritance tree just for that feature.

Please implement it as its own interface. Try to put most/all of the logic in default methods (in the interface itself). Any stored data will need to be in the implementation classes and will have to be passed in to the shared/default worker methods in the interface.

OK, agreed.

The task branch 1834a has been updated for review to revision 11154.

This is the stub based implementation of the involved classes and all DnD related attributes and methods. Now it can be compiled. The default logic is in interfaces. Also I have added conversion support for ROW-RESIZABLE browse widget attribute. Continue with particular implementation of the direct manipulation first.

#23 - 05/29/2017 09:32 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11155.

The server side implementation for direct manipulation has been added. Need to make another testcase to verify all setters/getter to get properly compiled. Continue working.

#24 - 05/30/2017 04:10 PM - Eugenie Lyzenko

The testcases repo has been updated to revision 1613.

The task branch 1834a has been updated for review to revision 11156.

The change includes conversion support for frame and widget DROP-TARGET option passed to widget or frame definition phrase. Also the new testcase has been written to check/demo all GRID-* related getters and setters are converting and compiling properly. So I think the base conversion support is completed(more or less runtime implemented on stub level).

The attribute DROP-TARGET is implemented as part of the BaseConfig. The default value is false. Only widgets that support this attribute are able to change/get this value. So even if widget that not supporting the option will have this inside config - the value is false and no chances to set/get it. I

think it is better than having several configs with same data member.

For the rest of the file list dropping functionality I'm planning to create special helper class not strongly attached to widget with ability to use existed server<->client calls for required data transfer. Just because we can have only one file list drop operation per session or may be even per machine user. I guess it will be right to make some separation here.

Continue with runtime implementation.

#25 - 05/30/2017 08:56 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11157.

This change introduced the suggested approach to class implementation access from interface default methods. The DROP-TARGET attribute setting/getting is completely implemented within interface defaults.

#26 - 05/31/2017 06:30 AM - Greg Shah

The attribute DROP-TARGET is implemented as part of the BaseConfig. The default value is false. Only widgets that support this attribute are able to change/get this value. So even if widget that not supporting the option will have this inside config - the value is false and no chances to set/get it. I think it is better than having several configs with same data member.

Actually, recently we have been moving to a different approach. I don't want to add overhead to the BaseConfig and GenericWidget with features that are only used by a subset of classes. I would rather have some duplicate implementation and push the features down to the exact locations in the hierarchy which use them.

#27 - 06/08/2017 06:09 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11158.

Reworked to resolve the notes for DroppableInterface implementation. Removed DROP-TARGET related flag from BaseConfig class. Also minor Javadoc errors have been fixed.

#28 - 06/08/2017 08:47 PM - Eugenie Lyzenko

This is the rest of file list dropping implementation plan. Because all real event are happening on the client side (grab files and drop into widget area) the server side widget implementation needs the communication with the ThinClient. So I'm going to introduce a set of the ClientExport calls for server side to get required data from client to server to implement END-FILE-DROP, GET-DROPPED-FILE methods and NUM-DROPPED-FILES attribute. The client session will have special helper class to handle the rest of the internals and mouse related events. The description is very generic and can be changed during opening new details or requirements but main idea is here.

For now I'm planning to make full stub implementation then shift to the main direct manipulation part 1 implementation.

#29 - 06/09/2017 07:32 AM - Greg Shah

I like your plan.

#30 - 06/09/2017 01:42 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11159.

Added new class to `../ui/client/gui` package to handle dropped files list functionality with only stub implemented. Currently the approach is considering to be used via static methods. I thought about non-static instance inside `ThinClient`. But it did not look the good because all widgets should have access for helper while drag-drop process and we will have to make many calls to TC instance while we probably need to rapidly access helped directly from widget code. So for now I have left the class internals as static.

Continue working.

#31 - 06/19/2017 09:11 PM - Eugenie Lyzenko

After doing some investigation for native 4GL behavior for direct manipulation I've made conclusion the key actor here for frames is the mouse events handler we need to refactor. The mouse events on client side will generate required triggers and change the widget's config values that will be visible on server side immediately via regular config sync. The task can be divided to 2 pretty separate parts, containers(window, frame and dialog) implementation and so-called "field level" widget implementation.

Start implementation from proper mouse handling inside frame. In addition to implementing window/frame/dialog functionality we will have more clear picture of the widget's support for mouse events and visual representation(drawing selection boxes around widgets with or without resize anchor boxes).

The handler requirements:

1. Mouse click selects frame if `FRAME:SELECTABLE` is `TRUE`
 - Mouse drag starts drawing selection box if `FRAME:BOX-SELECTABLE` is `TRUE`. This selects all selectable widgets becoming inside selection box and de-select already selected widgets in this box.
 - Mouse drag starts moving the frame if `FRAME:MOVABLE` is `TRUE`
 - Otherwise mouse drag do nothing inside frame
2. Mouse drag for frame title moves the frame
3. Mouse drag for frame resize anchor boxes resizes the frame(if the `FRAME:RESIZABLE` is `FALSE` - there is no frame resize anchor boxes)

The next step is to properly set up frame/window mouse handler.

#32 - 06/20/2017 11:04 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11160. Rebased with current trunk 11153, new revision is 11161.

Adds client side direct manipulation mouse handler base functions. It is registering for Window and Frame GUI widgets. For now it just display to console the ID of the widget for press/release/click/drag operations. The real functionality is in progress.

#33 - 06/27/2017 09:42 PM - Eugenie Lyzenko

The additional experimenting performed to find out the right approach to easily and more effectively insert mouse direct manipulation handler. The potential conflict with existing hover action handler discovered. The primary idea to move handler install code on the AbstractWidget class layer does not look right just because the code to be implemented is GUI specific and will become extra load for non-GUI mode. On the other hand ignoring handler install for all widgets disables proper event processing for non-frame based widgets like button, fill-in or rectangle. However code duplication for every GUI widget also does not look correct. Continue working to find the best approach.

#34 - 06/28/2017 10:21 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11162.

The next preparation step to implement client side of the direct manipulation. The single unified place has been found to install mouse events handler - AbstractWidget class. Need to exclude ChUI mode and skip widgets from process. Also changed the default value for selectable widget attribute to false. Continue working. Need to carefully separate regular mouse press/release/click events from ones that are related to direct manipulation. Sometimes the functionality is mutually exclusive.

Planning to add box selection code next and rectangle widget selection fix. The area inside the rectangle widget border (even not filled) belongs to the rectangle, not to the parent frame.

#35 - 06/29/2017 09:46 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11163.

This is the base skeleton functionality added to implement direct manipulation handler for widget selections. Still not having visual representation for selected widgets not for selection box itself. The output to console for debugging purpose. So this is in active development at this time. However the basic idea is here. The selection box is computing on mouse release event. And if there are widgets inside the area - it becomes selected. The mouse drag native event will be used for drawing selection dash box itself.

Also had to fix GUI rectangle to enable some mouse events processing. Continue working.

#36 - 06/30/2017 09:24 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11164.

This update starts visual effects implementation for direct manipulation handler. Now it is possible to see the selection box rectangle while selecting widgets by mouse drag operation. The handler uses immediate painting within graphics batch braces. The parent container refresh is used to clean previous rectangle. This approach is not ideal but unfortunately the Java2d does not have support for color invert mode with primitives (having this support we could avoid container repaint by using another inverted paint to restore the background).

Continue working. Need to design the approach to draw selected widgets around frames (resizable too). Planning to have the approach that does not change every widget drawing code if possible.

#37 - 07/03/2017 07:13 AM - Eugenie Lyzenko

The task branch 1834a has been rebased with current trunk 11154, new revision is 11165.

#38 - 07/03/2017 09:24 AM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11166.

Added javadoc fix for ClientsToPortsGenerator.java.

#39 - 07/03/2017 10:34 PM - Eugenie Lyzenko

- File *dnd_test0_1_fwd_20170703a.jpg* added

The task branch 1834a has been updated for review to revision 11167.

Starting with this update the implementation can become visible. The selected widgets are now enclosing with respective rectangle depending on whether the widget is resizable or not. The widget status renderer has been implemented in FrameGuiImpl class due to looks like it is theme independent. The picture attached represents the FWD testcase for direct manipulation in the current implementation status.

Continue working.

#40 - 07/04/2017 03:50 PM - Eugenie Lyzenko

The task branch 1834a has been rebased with current trunk 11155, new revision is 11168.

#41 - 07/05/2017 10:47 PM - Eugenie Lyzenko

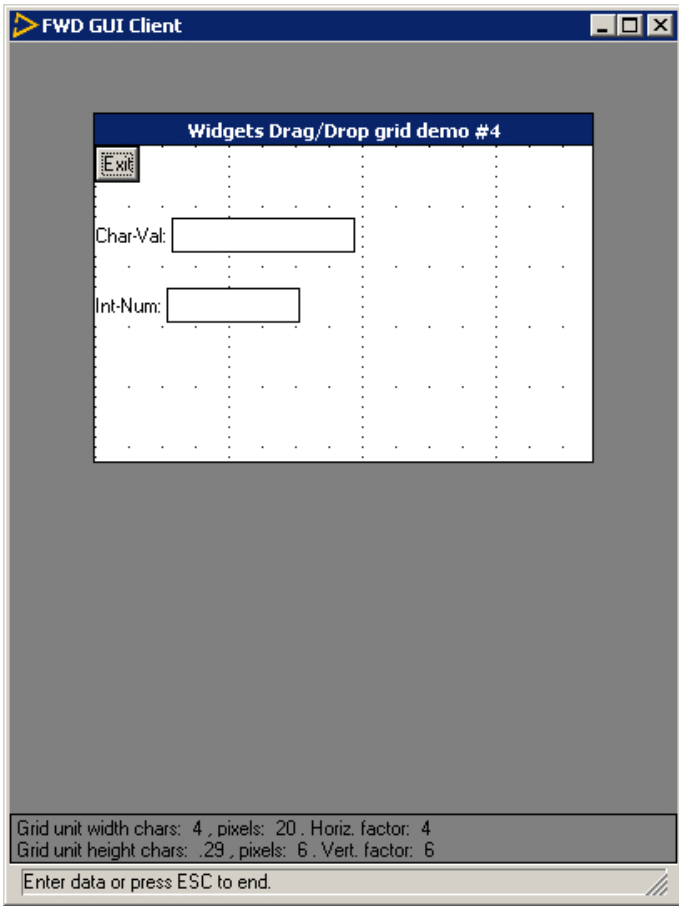
- File *dnd_test4_2_4gl.jpg* added

- File *dnd_test4_fwd_20170705a.jpg* added

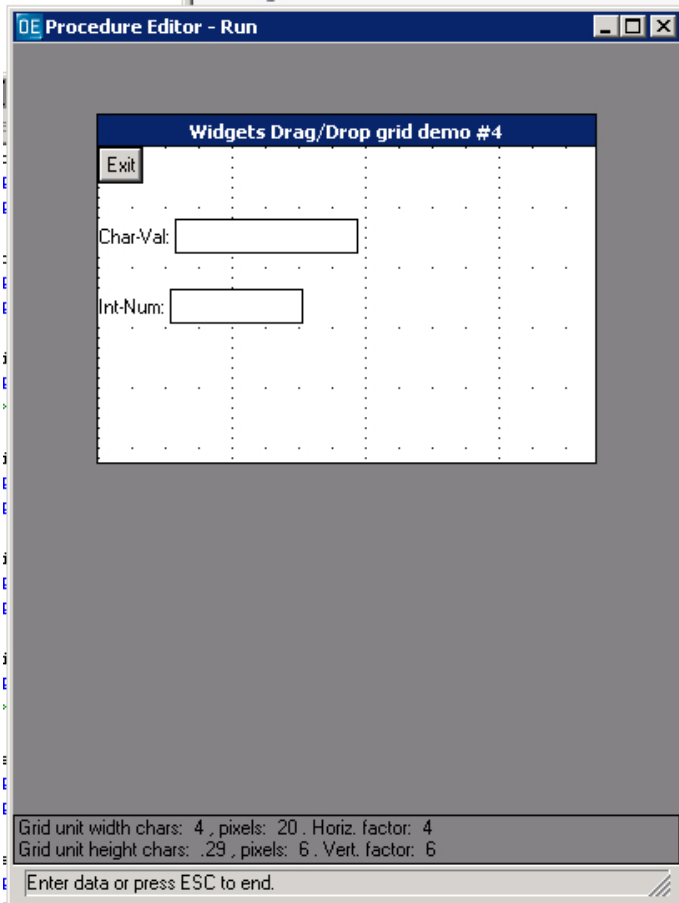
The task branch 1834a has been updated for review to revision 11169.

This update adds support for frame grid drawing and makes synchronization for grid related chars/pixels parameters. Changing pixel based ones should automatically sync char bases ones and vice versa. Also the grind painting should be performed just after the frame background refresh but before any widget drawing from the frame content. The pictures below shows the FWD implementation comparing to original 4GL.

FWD version:



4GL version:



#42 - 07/06/2017 11:55 AM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11170.

Small update to handle default setting for grid related attributes. The approach is to use internal pixel based setters instead of class field init to be able to sync character based options as well.

The testcases repo has been updated to revision 1619 to include recent tests changes.

#43 - 07/06/2017 01:45 PM - Ovidiu Maxiniuc

Review of 1834a/11169.

Nice job! Please see my observation below:

1. in the default `BrowseInterface.setRowResizable(logical flag)`, if the argument is unknown, warning 4083 should be displayed. If only the (boolean) signature of the method is implemented in derived classes, the warning is not shown. The same for `DroppableInterface.setDropTarget(logical flag)`, `DroppableInterface.getDroppedFile(NumberType index)`; and other default setter methods in `DirectManipulationInterface`. The same warning is shown in `setMovable(logical)` and `setResizable(logical)` from `BaseEntity`.
2. in `PaneConfig`: do we need to keep both `gridUnitHeightChars` and `gridUnitHeightPixels` (same for width)? I think ABL only uses this a help for 4GL programmers. Internally we should have a single value (pixels/integer) and when then linearly convert it to character units when asked/set. It doesn't make sense to keep both and do the overkill management;
3. there are some javadoc fixes for `<true>true</true>` in window related classes. I also have them in 3279a, which is in tests now. Don't remove them now, but take care of the rebase with a trunk in which I have eventually already committed those changes;
4. `MouseDirectManipulation`
 - update `Module & Abstract` in file header;
 - I did not run the code of this class yet, but I can share from my GUI experience with previous projects. Remember that in case of drag, the mouse events always come in this order: `PRESS (DRAG)* RELEASE (CLICK)?`. The last - `CLICK`, only occurs if no `DRAW` was performed. In this case should be ignored. The initialization of `startPoint` should always be done in `PRESS` event. If you get notified about a `DRAG` without a prior `PRESS`, then something is wrong! Also, all rendering should be done in `DRAG`, using overlaid XOR operation, twice: once using the old position - this will remove the previous drawing and once to draw the dotted rectangle to new position, creating the actual animation. This way, only the truly affected pixels are updated, a full-redraw of the canvas with all contained widgets is not required, improving this way the UI performance. The intermediary coords are not set to widget until the `RELEASE` event. This is the moment when the canvas is first and last time repainted during this process!

If you consider so, please contact me for further discussion of the animation above.

#44 - 07/06/2017 02:53 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Also, all rendering should be done in `DRAG`, using overlaid XOR operation, twice: once using the old position - this will remove the previous drawing and once to draw the dotted rectangle to new position, creating the actual animation. This way, only the truly affected pixels are updated, a full-redraw of the canvas with all contained widgets is not required, improving this way the UI performance.

Yes, I know this great trick. But the operation should be Invert(bitwise NOT_DST), not XOR. Then making Inversion twice we get original color. But as far as I know Java2D does not have this composition mode. Am I wrong? I have tried to use `gd.setXORComposite()` and this does not work.

#45 - 07/06/2017 03:05 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Review of 1834a/11169.

1. in PaneConfig: do we need to keep both `gridUnitHeightChars` and `gridUnitHeightPixels` (same for width)? I think ABL only uses this a help for 4GL programmers. Internally we should have a single value (pixels/integer) and when then linearly convert it to character units when asked/set. It doesn't make sense to keep both and do the overkill management;

Some comments. Yes, this way we simplify the attributes setting. But getting become more complex for char based units. We will have to make pixels->chars transformation every time asking grid size in chars. Is this really that optimization we want to do?

#46 - 07/06/2017 04:54 PM - Ovidiu Maxiniuc

- File `rectangle-select-and-drag.png` added

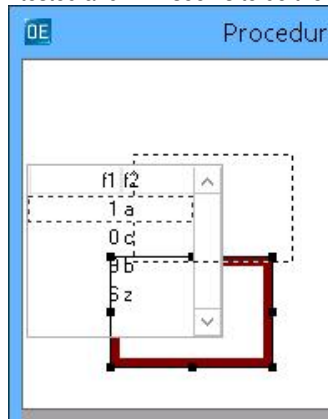
Eugenie Lyzenko wrote:

Ovidiu Maxiniuc wrote:

Also, all rendering should be done in DRAG, using overlaid XOR operation, twice: once using the old position - this will remove the previous drawing and once to draw the dotted rectangle to new position, creating the actual animation. This way, only the truly affected pixels are updated, a full-redraw of the canvas with all contained widgets is not required, improving this way the UI performance.

Yes, I know this great trick. But the operation should be Invert(bitwise NOT_DST), not XOR. Then making Inversion twice we get original color. But as far as I know Java2D does not have this composition mode. Am I wrong? I have tried to use `gd.setXORComposite()` and this does not work.

Please update testcases project. I added a quick demo: `uast/rendering/XORTest.java`. Hope this helps. I tested and ABL seems to do the same. Here is a screen-shot:



(you may need to save locally or in other tab and zoom in to see the details)

You can see that the dragged dotted rectangle intersects the brown (0x840000) rectangle widget and the resulting colour is some kind of

cyan(0x7bffff) - as result of the XOR op (0x840000 | 0x7bffff = 0xfffff, white). Also some points '2' of f2 header in the browse are turned white.

#47 - 07/06/2017 06:19 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Eugenie Lyzenko wrote:

Ovidiu Maxiniuc wrote:

Also, all rendering should be done in DRAG, using overlaid XOR operation, twice: once using the old position - this will remove the previous drawing and once to draw the dotted rectangle to new position, creating the actual animation. This way, only the truly affected pixels are updated, a full-redraw of the canvas with all contained widgets is not required, improving this way the UI performance.

Yes, I know this great trick. But the operation should be Invert(bitwise NOT_DST), not XOR. Then making Inversion twice we get original color. But as far as I know Java2D does not have this composition mode. Am I wrong? I have tried to use `gd.setXORComposite()` and this does not work.

Please update testcases project. I added a quick demo: `uast/rendering/XORTest.java`. Hope this helps.

Yes, thanks a lot! This is what I wanted to have but not found in Java2d. I think with `setXORMode()` we can avoid full container repainting.

#48 - 07/06/2017 08:57 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11171.

This is the partial notes resolution. Added missed warnings for invalid direct manipulation values. Updated file header. Also the new approach for selection rectangle drawing/erasing has been implemented using Java2d `setXORMode()`. Now we do not need to refresh full container content and repaint all widgets. Instead drawing rectangle second time restores the original pixel state.

Continue working.

#49 - 07/07/2017 10:36 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11172.

The update finished the notes resolution. The character based grid options were removed, on the fly calculation is used to get/set appropriate attributes. I guess the common usage is pixel based counterparts because only GUI mode has meaning for grid. And for GUI native units are pixels.

Also the web GUI driver was changed to include/set up XOR mode. Now it is implemented as Canvas composition "difference" mode for current color and white. Applying twice should get original color I guess. Unfortunately for now I'm not able to run web client for testcases due to strange issue with JETTY subsystem. Looks like the trunk 11154 requires directory change I'm missing.

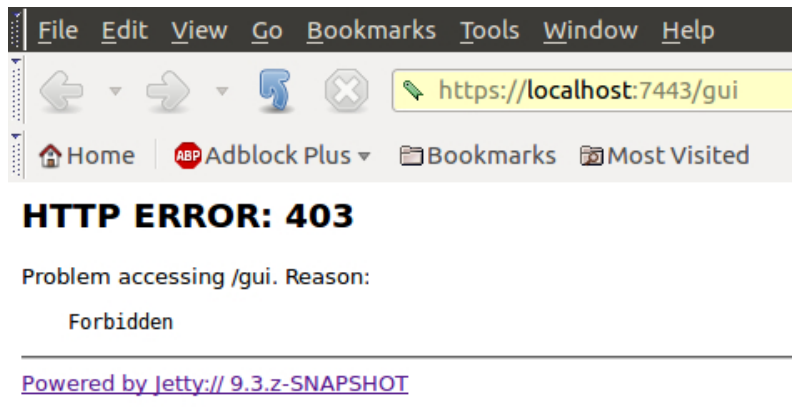
Ovidiu, do you have the ability to run testcases from uast/* within web client for your 11155 update?

#50 - 07/09/2017 05:49 PM - Eugenie Lyzenko

- File `web_client_20170707.jpg` added

This is the only log I have for web client with current trunk for hotel GUI.

Has anyone the same issue? I have added directory setups for 11155 trunk merge notification message. What I have missed in config?



#51 - 07/10/2017 08:52 AM - Ovidiu Maxiniuc

From email conversation:

On 07/06/2017 03:20 PM, Greg Shah wrote:

On 07/06/2017 08:06 AM, Eugenie V. Lyzenko wrote:

It it our concept to move all real painting from widget code into theme implementation? Or it is not mandatory and can be exceptions?

Good question.

Any drawing that is truly theme-independent, should be in common code. But we will need to test the 4GL to confirm this.

I thought about this issue. From my tests, it seems that the manipulation logic is done internally by ABL so it is not theme dependent. In fact there are no OS-level widget sub-component in Windows that an application can use in order to keep the L&F. If Microsoft would standardize something it would be the selection they use in their Office package (white squares with black border for move/resize and circles for altering the shape), which is actually duplicated by other applications that runs on Windows. However, OE/ABL have chosen not to follow those, instead they use fully black squares. This is clearly theme-independent (please see my note 46 above).

In my opinion, as long as we keep these manipulations away from normal widgets, it will be fine. We only need the geometry of the manipulated widget at the moment the gesture started, but it can be obtained using public API. If (theoretically) we will need to to make this rendering theme-dependent, we can add methods in Theme class for drawing the selection handles and moving dotted rectangle and delegate the respective actions from MouseDirectManipulation class to currently active Theme.

Eugenie, please note that there can be multiple widgets selected, and the used should be able to move them all at once. I think the current implementation does not support that. You need to notify all widgets from the current selection to 'move'. The resize gesture seems not to affect the other selected widgets.

#52 - 07/10/2017 10:32 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

From email conversation:

On 07/06/2017 03:20 PM, Greg Shah wrote:

On 07/06/2017 08:06 AM, Eugenie V. Lyzenko wrote:

It it our concept to move all real painting from widget code into theme implementation? Or it is not mandatory and can be exceptions?

Good question.

Any drawing that is truly theme-independent, should be in common code. But we will need to test the 4GL to confirm this.

I thought about this issue. From my tests, it seems that the manipulation logic is done internally by ABL so it is not theme dependent. In fact there are no OS-level widget sub-component in Windows that an application can use in order to keep the L&F. If Microsoft would standardize something it would be the selection they use in their Office package (white squares with black border for move/resize and circles for altering the shape), which is actually duplicated by other applications that runs on Windows. However, OE/ABL have chosen not to follow those, instead they use fully black squares. This is clearly theme-independent (please see my note 46 above).

In my opinion, as long as we keep these manipulations away from normal widgets, it will be fine. We only need the geometry of the manipulated widget at the moment the gesture started, but it can be obtained using public API. If (theoretically) we will need to make this rendering theme-dependent, we can add methods in Theme class for drawing the selection handles and moving dotted rectangle and delegate the respective actions from MouseDirectManipulation class to currently active Theme.

One more point. The Classic theme is the base for all others, correct? If we implement some drawing in Classic theme only - this will mean theme independent approach, right?

Eugenie, please note that there can be multiple widgets selected, and the user should be able to move them all at once. I think the current implementation does not support that. You need to notify all widgets from the current selection to 'move'. The resize gesture seems not to affect the other selected widgets.

Yes, thanks for reminder, this is known behavior and I already thinking about proper implementation.

The main point I'm confusing for now is the Web client functionality. I want the GUI changes I'm implementing to work the same way for both Swing and Web clients, especially when new features are adding to both GUI drivers. I think the primary client type the user will deal with is Web. So it is important the changes are working properly there. So can anyone clarify what is the status of the Web client for trunk 11155?

#53 - 07/10/2017 11:03 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

The Classic theme is the base for all others, correct? If we implement some drawing in Classic theme only - this will mean theme independent approach, right?

Currently, yes. But if will ever create some other Themes - like Linux or Ubuntu :), we may have to implement directly form Theme, creating a 'parallel' hierarchy. But this does not prevent us from creating an intermediary AbstractTheme class that contains Theme independent code.

Eugenie, please note that there can be multiple widgets selected, and the user should be able to move them all at once. I think the current implementation does not support that. You need to notify all widgets from the current selection to 'move'. The resize gesture seems not to affect the other selected widgets.

Yes, thanks for reminder, this is known behavior and I already thinking about proper implementation.

The main point I'm confusing for now is the Web client functionality. I want the GUI changes I'm implementing to work the same way for both Swing and Web clients, especially when new features are adding to both GUI drivers. I think the primary client type the user will deal with is Web. So it is important the changes are working properly there. So can anyone clarify what is the status of the Web client for trunk 11155?

Yup! I have the same HTTP ERROR: 403. I investigate this to see what caused it.

#54 - 07/10/2017 11:47 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

So can anyone clarify what is the status of the Web client for trunk 11155?

It it a bit strange. You need to set webClient/virtualDesktopEnabled to true in your directory.xml. For unknown reasons to me, the value was not previously read before. At least I did not change it. The WebHandler class that reads it hasn't also been changed in a while. As you can see, virtualDesktopEnabled is read with initial set to false. I didn't have it in registry so I had to add it manually. Hope this is also your case.

#55 - 07/10/2017 12:16 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Eugenie Lyzenko wrote:

So can anyone clarify what is the status of the Web client for trunk 11155?

It it a bit strange. You need to set webClient/virtualDesktopEnabled to true in your directory.xml. For unknown reasons to me, the value was not previously read before. At least I did not change it. The WebHandler class that reads it hasn't also been changed in a while. As you can see, virtualDesktopEnabled is read with initial set to false. I didn't have it in registry so I had to add it manually. Hope this is also your case.

Yes, exactly. Now the hotel GUI works fine with 11155 trunk. Thanks a lot!

Are you able to run standalone testcases with web client? The server directory is simple/server, the same repo as having uast/* testcases.

#56 - 07/10/2017 01:23 PM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

Are you able to run standalone testcases with web client? The server directory is simple/server, the same repo as having uast/* testcases.

OK. Now I'm running web client with testcases and my working branch. The last issue was wrong port to login, need to use localhost:7443/gui.

#57 - 07/10/2017 02:23 PM - Greg Shah

It is a bit strange. You need to set webClient/virtualDesktopEnabled to true in your directory.xml. For unknown reasons to me, the value was not previously read before. At least I did not change it. The WebHandler class that reads it hasn't also been changed in a while. As you can see, virtualDesktopEnabled is read with initial set to false. I didn't have it in registry so I had to add it manually. Hope this is also your case.

Yes, this is expected. Sergey implemented this feature to disable virtual desktop mode by default. The idea is that it is useful for testing but virtual desktop mode is not expected to be used for end customers.

#58 - 07/11/2017 09:54 AM - Greg Shah

From Eugenie:

The investigation for web client starting issues. The tests show additional subtasks for web client. The used in Swing java XOR display mode is not working with JavaScript approach taken for web client. For now the only way I see is to completely repaint the container to clean previous selection box. The more intelligent approach is postponed to the time when other functionality is completed. Now I'm working on different direct manipulation modes, widgets moving by drag, different selection modes (depending on mouse key modifiers) to implement complete 4GL behavior.

If you create the proper drawing/primitive APIs in the GuiDriver, then each driver can implement its own version of the compositing features as needed. The drawing code would still be common code, but each driver would be able to implement an optimized version.

#59 - 07/11/2017 10:55 PM - Eugenie Lyzenko

Greg Shah wrote:

From Eugenie:

If you create the proper drawing/primitive APIs in the GuiDriver, then each driver can implement its own version of the compositing features as needed. The drawing code would still be common code, but each driver would be able to implement an optimized version.

OK. I'm still hoping to find JS analog of the Java2D `setXORMode()` implementation.

The task branch 1834a has been updated for review to revision 11173.

This update introduces implementation of the widgets multiple selection mode by holding Ctrl key(EXTENDED key in 4GL terms) and MOVE operation. When drag starting over one of movable widgets the MOVE operation is starting. The mouse handler grabs the rectangle bound of all currently selected movable widgets and draws the dotted rectangles to show the new widgets position. When movable widget is not selected during MOVE start the widget become selected first de-selecting all previously selected widgets. The next step is to implement actual widgets coordinates change and the similar functionality for RESIZE operation.

#60 - 07/12/2017 06:16 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

I'm still hoping to find JS analog of the Java2D `setXORMode()` implementation.

Eugenie, I checked the code for XOR support. I think the existing `setXORComposite / resetComposite()` should suffice. In fact, for swing implementation I strongly recommend replacing the current implementation that uses internal proprietary API:

```
g2.setComposite(new XORComposite(new Color(0, 0, 0), NullSurfaceData.theInstance));  
g2.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER));
```

with following equivalent, portable code:

```
g2.setXORMode(Color.BLACK);  
g2.setPaintMode();
```

All that remains is to check which the XOR color is the appropriate one: BLACK or WHITE?

#61 - 07/12/2017 08:47 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Eugenie Lyzenko wrote:

I'm still hoping to find JS analog of the Java2D setXORMode() implementation.

Eugenie, I checked the code for XOR support. I think the existing setXORComposite / resetComposite() should suffice. In fact, for swing implementation I strongly recommend replacing the current implementation that uses internal proprietary API:

[...]

with following equivalent, portable code:

[...]

All that remains is to check which the XOR color is the appropriate one: BLACK or WHITE?

Yes, it works good for Swing client, the right color is White(0x00FFFFFF). Although I can not say from performance point of view it is better than using container repaint(I see the delay between mouse drag and rectangle rendering).

The problem is the missing such API for JavaScript canvas in Web client. Looks like the only we have is to play with Composition modes(the closest mode can be "difference"):

https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Compositing

The alternative approach for web client canvas could be to manually take the pixels array from area, invert them and put back.

#62 - 07/12/2017 08:57 AM - Greg Shah

Sergey: Please see the recent notes in this task and comment.

#63 - 07/12/2017 09:14 AM - Ovidiu Maxiniuc

There is a xor operation that make *Shapes are made transparent where both overlap and drawn normal everywhere else.*, but unfortunately the result is not exactly how I expected. Here is a demo:

https://www.w3schools.com/tags/playcanvas.asp?filename=playcanvas_globalcompop&preval=xor

Apparently the operation is done on the drawn mask instead of processing pixel-by-pixel :(.

I wonder if it can be adjusted to do what we need ?

#64 - 07/12/2017 09:27 AM - Ovidiu Maxiniuc

Just had an idea, I don't know how hard is to implement it.

We can use a second, temporary transparent canvas in front of the normal one. Instead of XOR, we draw on this overlay. When drag operation ends, we clean/hide/remove it. There will not be the exact color composition, but we don't destroy the current image and it would be clean and fast.

#65 - 07/12/2017 10:19 AM - Sergey Ivanovskiy

Related to the current status of the trunc 11155, all changes of the web client were described in merge notification for task branch 2683a

These new directory settings under webClient were added

```
<node class="container" name="portsrange">
  <node class="integer" name="from">
    <node-attribute name="value" value="7449"/>
  </node>
  <node class="integer" name="to">
    <node-attribute name="value" value="7459"/>
  </node>
  <node class="string" name="nameprefix">
    <node-attribute name="value" value="client"/>
  </node>
<!-- it can be skipped if the reverse proxy is a front end server -->
  <node class="string" name="forwardedHost">
    <node-attribute name="value" value="www.goldencode.com"/>
  </node>
<!-- it can be skipped since its value is always 'https' -->
  <node class="string" name="forwardedProto">
    <node-attribute name="value" value="https"/>
  </node>
</node>
<!-- the default setting is FALSE -->
<!-- This setting is required to start web client in virtual desktop mode -->
<node class="boolean" name="virtualDesktopEnabled">
  <node-attribute name="value" value="TRUE"/>
</node>
```

This change "virtualDesktopEnabled" value is false by default. has an impact on the web client.

#66 - 07/12/2017 10:24 AM - Sergey Ivanovskiy

Ovidiu Maxiniuc wrote:

Just had an idea, I don't know how hard is to implement it.

We can use a second, temporary transparent canvas in front of the normal one. Instead of XOR, we draw on this overlay. When drag operation ends, we clean/hide/remove it. There will not be the exact color composition, but we don't destroy the current image and it would be clean and fast.

Please look at the function `MouseResizable()` in `p2j.mouse.js`. The second peer canvas is used while the original window is resizing.

#67 - 07/12/2017 11:01 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Just had an idea, I don't know how hard is to implement it.

We can use a second, temporary transparent canvas in front of the normal one. Instead of XOR, we draw on this overlay. When drag operation ends, we clean/hide/remove it. There will not be the exact color composition, but we don't destroy the current image and it would be clean and fast.

The issue here is we need the original canvas to be restored every time the mouse coords are changing while mouse drag. Consider the following cycle(schematic language):

```
drag until (mouse released)
{
  restore original canvas
  draw new positioned selection box or widget's bound rectangles on the new temporary canvas
}
```

Do you think it will be faster?

#68 - 07/12/2017 11:10 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11174.

The update starts handling implementation for widget resize operation. This is first draft but base GUI ideas can be found. The resize operation has the higher priority over all others. Only one selected widget can be resized. Currently dragging with resize handle boxes produces the dotted rectangle drawing for new suggested size. All resize handles are working, each one defines single resize direction remaining unchanged until mouse release event.

There are painting optimization to be done here. Also we need to investigate resize behavior when widget is resizable but not movable. And like for widgets moving we need to add 4GL events/triggers generation, real widget resize operation implementation. Continue working.

#69 - 07/13/2017 03:15 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11175.

Small bug fix for 11174 update with wrong array loop processing. Also added possibility to update mouse pointer when resize operation is in action.

Additional 4GL investigation shows even the widget is not movable it's position is changed if resize operation assumes the position change(for example top-left resize handle moving).

#70 - 07/13/2017 09:09 PM - Eugenie Lyzenko

Greg,

Investigating next steps for implementing direct manipulation functionality I've got a point I need to discuss future approaches. Because next changes will complicate the code and make the refactoring pretty problematic. Currently the attributes and client side GUI part is basically working. The next step is to implement reactions for direct manipulation action(widget move, resize, ...). And here we have two options:

1. Do not implement direct manipulation events making all required changes on client side just right inside direct manipulation handler(including attributes proper synchronization). This approach does not support event overriding in FWD code as it is possible in 4GL.
2. Add support for direct manipulation event. Generate proper event and post the event to target widget. Then widget will receive the event and make appropriate action(change direct manipulation attributes). This implementation way will support ability to override event by programmer, just like it happens in 4GL.

I would suggest to go the way number 2 which is longer I guess. Just because I think we need to implement everything in a right way from beginning. This way our code should work in a same way and concept as original 4GL code. What do you think? Is it acceptable?

#71 - 07/13/2017 09:19 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11176.

The update fixes event priority issues and different selection modes behavior for SELECT and EXTEND mouse artificial keys.

#72 - 07/13/2017 11:47 PM - Greg Shah

I would suggest to go the way number 2 which is longer I guess. Just because I think we need to implement everything in a right way from beginning. This way our code should work in a same way and concept as original 4GL code. What do you think? Is it acceptable?

Yes, please get it implemented to match the 4GL behavior including all of the event processing. I understand it will take longer.

#73 - 07/14/2017 10:46 PM - Eugenie Lyzenko

Greg Shah wrote:

Yes, please get it implemented to match the 4GL behavior including all of the event processing. I understand it will take longer.

OK.

The task branch 1834a has been updated for review to revision 11177.

This update starts implementing direct manipulation events handling. The base was added to handle the approach. ThinClient routes the messages to target widget. The messages generated in MouseDirectManipulation handler as reaction for different mouse events. This excludes calling the widget that does not support direct manipulation. For now the SELECTION/DESELECTION events are processing. But all the rest will go the same way. MouseDirectManipulation will be cleaned from direct change the direct manipulation attributes.

Some painting issues detected during merging to the new approach. So need to investigate if it is race condition or wrong event ordering. Planning to add the rest events handling next. The new event class - DirectManipulationEvent was introduced to cover all needs here. All new event codes/IDs were added to the Keyboard class. We will pass additional options for MOVE/RESIZE cases like pixels shift to change widget location and/or dimension.

#74 - 07/16/2017 07:01 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11178.

This is the fix for painting issue with new event based approach. The frame refresh was performed before the attribute change. This was the root cause. Now it is initiated from widget code that change the attribute. So far the issue is gone.

#75 - 07/17/2017 10:28 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11179.

This update changes the implementation approach to message based one. The selection/deselection messages handling completed, other are in progress. The attribute NUM-SELECTED-WIDGETS is now changing once selection/deselection event is coming to the target widget. Also the event format is extended to pass the flag forcing the parent container to repaint. This help to properly redraw the frame during direct manipulation. Continue working. Next step is to completely implement MOVE/RESIZE events.

#76 - 07/18/2017 08:51 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11180.

Added real handling for MOVE/RESIZE events and other direct manipulation messages. Now all real processing, attributes changes are performing once respective message is receiving. The MOVE and RESIZE changes are working. The widget resizing is not always performing OK because widget's limitation for size change. I think due to some font size change constraints. Need to be investigated additionally.

Also the code to execute appropriate triggers has been added to the branch. Now the triggers for direct manipulation event is calling. The testcases repo has been updated to revision 1624. Some triggers added to testcases to demo/debug.

Next step will be to include GRID related functionality that change MOVE/RESIZE results, check events ordering and finish GUI painting issues for Swing and Web clients.

#77 - 07/19/2017 10:38 AM - Eugenie Lyzenko

The testcases repo has been updated to revision 1625. Added another test(uast/drag_n_drop/dnd_test0_1.p) to investigate 4GL reactions on GRID-SNAP mode on or off. Found the following:

1. When grid snap is on the move resize dotted box position and dimensions are not arbitrary. They are always aligned with the minimum grid step cell. Not possible to have box bounds between grid lines(Mouse drag action handling). Both widget position and widget size are affected when widget resizing and only position is affected when widget moving.
2. Turning grid snap off at run time restores original dotted boxes behavior.
3. The box selection mode does not depend on grid-snap value.

This means the GRID-SNAP is working not only when mouse released and widgets receive the END-MOVE/END-RESIZE event but also when mouse is dragging and we can see only suggested new position and size. Planning respective FWD implementation.

#78 - 07/19/2017 01:37 PM - Eugenie Lyzenko

The note [ZZ](#) has been updated to reflect new findings.

#79 - 07/19/2017 09:59 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11181.

Adding runtime implementation for GRID-SNAP attribute. All findings for MOVE/RESIZE operations have been implemented. Continue with proper optimized container refresh for different clients(swing and web) approach and other remaining features.

#80 - 07/20/2017 10:28 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11182.

Some unification and optimization was performed to make code more readable. In addition the approach to handle cases when START-* event was consumed by NO-APPLY trigger result. In this case the appropriate END-* event will not be generated for target widget. The solution is to make the hook inside ThinClient to store consumed message inside direct manipulation handler.

The remaining events missed at this time but to be implemented are START-ROW-RESIZE/END-ROW-RESIZE. Continue working.

#81 - 07/21/2017 09:06 PM - Eugenie Lyzenko

The testcases repo has been updated to revision 1627.

Two new testcases added to investigate and properly implement START-ROW-RESIZE/END-ROW-RESIZE for browse widgets and reaction on consuming START-*/END-* related events. We can now see how it actually work on original 4GL system.

The task branch 1834a has been updated for review to revision 11183.

This update introduce starting implementation for START-ROW-RESIZE/END-ROW-RESIZE events for browse. The server side implementation completed, working on client side. Also the checking for trigger consumed events has been implemented the same way as the Progress does. The END-BOX-SELECTION processing has been reworked to reflect the 4GL reaction. Now the FWD selects/deselects appropriate widgets once the END-BOX-SELECTION event received taking into account possible event consuming.

Continue working.

#82 - 07/21/2017 10:13 PM - Eugenie Lyzenko

The more testing discovered new issue with END-BOX-SELECTION approach currently implemented. On the time the event is receiving the total number of the selected widget already has the updated value. This slightly conflicts with what we can read in official 4GL documentation about direct manipulation event processing order. 4GL receives the END-BOX-SELECTION event and only after this message the appropriate SELECTION/DESELECTION events are sending to widgets that are going to be selected or deselected. This means at the moment END-BOX-SELECTION arriving there are no changes for selection(NUM-SELECTED-WIDGET value). So the actual working is need to be investigated additionally.

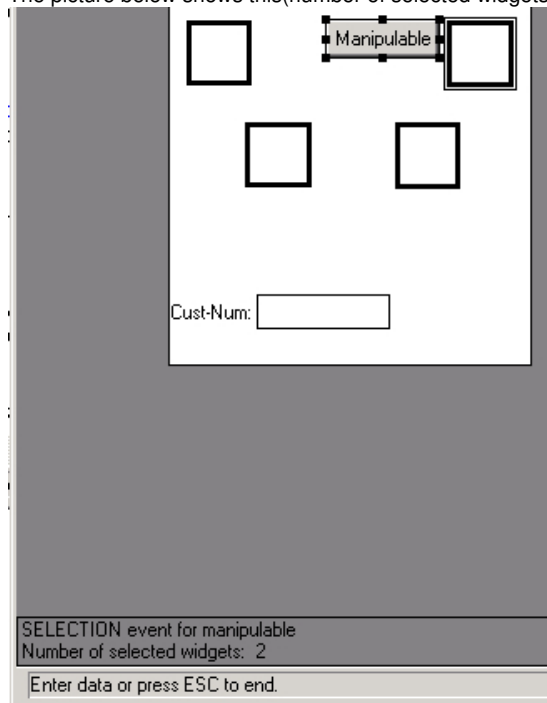
#83 - 07/22/2017 06:46 PM - Eugenie Lyzenko

- File *dnd_test0_3_0_4gl.jpg* added

The task branch 1834a has been updated for review to revision 11184.

This is the fix for END-BOX-SELECTION processing. With little changes it returns to the previous version. The testcase(*uast/drag_n_drop/dnd_test0_3.p* available in 1628) shows actually select/deselect generates before END-BOX-SELECTION as I initially suggested. So again the 4GL documentation has wrong statement.

The picture below shows this(number of selected widgets computed on END-BOX-SELECTION receiving):



#84 - 07/24/2017 10:11 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11185.

This update makes first steps for implementing START-ROW-RESIZE/END-ROW-RESIZE for browse widget. Had to rework some base classes in direct manipulation making implementation cleaner minimizing instanceof usage. The Browse widget is pretty complex one so it consists of several parts(main GUI client class and one additional GUI class per each column), several configs(one main and one additional per column). This makes direct manipulation approach to be more smart than for other widgets to do all thing in a right way. Need some time for additional investigation to find a best way to implement. Continue working.

#85 - 07/25/2017 04:17 AM - Greg Shah

Stanislav: Please review this task so that you are aware of the changes Eugenie is making that are browse related. If it makes sense, coordinate with him in regard to your work in [#3275](#).

#86 - 07/25/2017 01:16 PM - Stanislav Lomany

Yep, I have *a lot* of browse changes to commit, and I don't know how to coordinate other that committing it first. Eugenie, if you need to drag/resize something, check out how column drag and resize is implemented, that may save your time a bit.

#87 - 07/25/2017 10:47 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11186.

The update makes preparation steps for approach implementing row resizable browse feature. The idea is to register browse column widgets as direct manipulation handler but the events will be routed to main browse widget. If column receives SELECTION/DESELECTION - the only it makes is to generate new event for browse widget if the event is changing the browse selection status. For row resize events I'm planning the following. First we need to get the row number to resize(this is important to draw the dotted rectangle for new suggested row height). When mouse will start dragging over row resize area(3 pixels below top row border and 3 pixels above bottom row border) the handler get row number and direction(depending on top or bottom row border to drag). When drag finished the end row resize will be post to the browse main widget(may be via browse column widget). Continue working. Expecting to finish row resize feature in a next day.

#88 - 07/26/2017 10:35 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11187.

The update includes row resize feature complete implementation. Everything is working based on events properly routed to main browse widget. The GRID-SNAP feature is also supported. This means the new row size is aligned to the frame grid units. This means browse coordinates are transforming to the frame related when new size is applied. The testcase to check is uploaded with testcases repo 1629.

Continue working. The next step is to resolve client painting issue with dotted rectangles clean.

#89 - 07/27/2017 03:54 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11188.

This is the fix for class cast exception I have found while testing browse widget selection/deselection using CTRL extended mouse key modifier. Looks like to many instanceof statements in MouseDirectManipulation handler. I think need to find a way to avoid or minimize it because using instanceof means poor design may be.

#90 - 07/27/2017 04:12 PM - Greg Shah

I think need to find a way to avoid or minimize it because using instanceof means poor design may be.

Agreed. Please put the widget-specific code into the widget classes and call it using an interface. If it is suitable, you can add to the AbstractWidget interface, but it is also OK to add a specialized interface for direct manipulation.

#91 - 07/27/2017 10:42 PM - Eugenie Lyzenko

Greg Shah wrote:

I think need to find a way to avoid or minimize it because using instanceof means poor design may be.

Agreed. Please put the widget-specific code into the widget classes and call it using an interface. If it is suitable, you can add to the AbstractWidget interface, but it is also OK to add a specialized interface for direct manipulation.

OK.

Investigating the difference in web client working with direct manipulation. For some reasons the mouse moving events required to handle pointer change while widget resize does not work properly. Also box selecting in frame has side effect in frame duplication and wrong selection box drawing. Need to understand what is so different in web client implementation related to mouse events.

#92 - 07/28/2017 05:53 AM - Greg Shah

Ovidiu: Would you please do another code review? I think the branch is getting close to the final release candidate.

#93 - 07/28/2017 09:11 AM - Eugenie Lyzenko

Starting the rebase with 11156 trunk. Please do not change anything here until completion message.

#94 - 07/28/2017 10:17 AM - Eugenie Lyzenko

The task branch 1834a has been rebased with trunk 11156 trunk. New revision is 11189.

The old javadoc fixes were taken from 11156. I have found another javadoc issues. Will fix in 1834a shortly if there are no objections. Then the branch will be finally ready for review.

#95 - 07/28/2017 11:14 AM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11190.

The javadoc issues introduced in trunk 11156(EditorInterface.java and EditorWidget.java) have been fixed with this update.

Ovidiu, now the branch is finally ready for your look.

#96 - 07/28/2017 02:05 PM - Ovidiu Maxiniuc

Review based on rev 11190 of 1834a.

Good work! I added a short list of minor things to fix and a couple of questions below:

- PaneEntity: in setGridFactorHorizontal: what happens to vertical factor if setting it to 1, but it was already 1. Same 'mirror' question for setGridFactorVertical;
 - Line 568: unnecessary return at end of method;
- PaneConfig: minor typo line 150: *Flag to indicate if the ...*;
- BaseEntity: lines 745, 792: from my tests, you need to throw condition 4083: ***Unable to assign UNKNOWN value to attribute <attribute> on <widget id>*;
 - Line 934: Is the TODO note still valid?
- AbstractWidget: lines 805-810 (and subsequent): non-standard logging to System.out. I assume they will be cleaned up or converted in low-priority logs.
- FrameGuiImpl: line 1696: can the isDirectManipulationCapable() be used here instead of instanceof? Similar for MouseDirectManipulation:697, 1063, 1283;
- WindowGuiImpl: H080 entry is the only change in file, you rolled-back previous changes;
- Remaining issue: painting the XOR mode in JS;
- PaintPrimitives: IMHO, SET_XOR_MODE does at least for swing driver the same job as SET_XOR_COMPOSITE. We will decide whether we keep this after we find a solution for previous issue;
- DnD support classes: only conversion & runtime stubs. I guess there is a dedicated task for these.
- MouseDirectManipulation, instead of repainting the whole surface in repaintParentContainer, we can optimize by only repainting the bounding rect of the overlay selection. Another solution to be tested: when starting a manipulation gesture, save the current content of the window in a pixmap, at the time dragInProgress = true. When repaintParentContainer is needed, just refresh the content of the whole window from saved pixmap. I believe this is faster than painstakingly drawing each widget, and almost no traffic at all;
 - Line 802/816/820: yShift computed but not used, I guess this is dead code;
 - The gd instanceof GuiWebDriver appears multiple time in this class. Maybe we should add a method to be queried instead, like gd.supportsXorMode();
 - Look for typos: getMovableWidgetsBounds, coonsumed, slection, mutial, coodrinates, cooredinate, iside, brouse, resise, verical, widgees.

#97 - 07/28/2017 02:27 PM - Greg Shah

The gd instanceof GuiWebDriver appears multiple time in this class. Maybe we should add a method to be queried instead, like gd.supportsXorMode();

This makes sense.

DnD support classes: only conversion & runtime stubs. I guess there is a dedicated task for these.

What classes are you referencing here?

#98 - 07/28/2017 02:33 PM - Ovidiu Maxiniuc

Greg Shah wrote:

DnD support classes: only conversion & runtime stubs. I guess there is a dedicated task for these.

What classes are you referencing here?

Main is DroppableInterface. Methods are partially implemented in BrowseWidget, ControlEntity, PaneEntity. The interface exposes methods and attributes related to file drop.

#99 - 07/28/2017 02:51 PM - Greg Shah

I thought that the implementation of that was complete. The implementers of DroppableInterface only need to implement isDropTarget() and setDropTarget() since the rest of the methods are implemented as default methods. Those call the ThinClient for service as needed.

Looking deeper, I do see that DragDropHelper needs to be implemented. Is there something else?

#100 - 07/28/2017 05:18 PM - Eugenie Lyzenko

Greg Shah wrote:

Looking deeper, I do see that DragDropHelper needs to be implemented. Is there something else?

Everything related to client side file list drag-n-drop functionality is not implemented yet. I've planned to finish the widget direct manipulation support first.

PaneEntity: in setGridFactorHorizontal: what happens to vertical factor if setting it to 1, but it was already 1. Same 'mirror' question for setGridFactorVertical;

Nothing, if it is already 1 we do not need to change it. Or I misunderstood your question.

AbstractWidget: lines 805-810 (and subsequent): non-standard logging to System.out. I assume they will be cleaned up or converted in low-priority logs.

Yes, I used these logs to debug the mouse related code. This will be cleaned in final revision.

FrameGuiImpl: line 1696: can the isDirectManipulationCapable() be used here instead of instanceof? Similar for MouseDirectManipulation:697, 1063, 1283;

Yes, I'm planning to get rid of the poor design instanceof calls.

Remaining issue: painting the XOR mode in JS;
MouseDirectManipulation, instead of repainting the whole surface in repaintParentContainer, we can optimize by only >repainting the bounding rect of the overlay selection. Another solution to be tested: when starting a manipulation gesture, save the current content of the window in a pixmap, at the time dragInProgress = true. When repaintParentContainer is needed, just refresh the content of the whole window from saved pixmap. I believe this is faster than painstakingly drawing each widget, and almost no traffic at all;

Yes, this is still issue under development for how to implement this feature in Web client.

PaintPrimitives: IMHO, SET_XOR_MODE does at least for swing driver the same job as SET_XOR_COMPOSITE. We will decide whether we keep this after we find a solution for previous issue;

I have not introduced SET_XOR_COMPOSITE operation so can not tell for sure what is the purpose of it. For clean up previous drawing it is not perfect, XOR_MODE is better.

Line 802/816/820: yShift computed but not used, I guess this is dead code;

Thank you! This is not dead, yShift must be used with event construction.

The gd instanceof GuiWebDriver appears multiple time in this class. Maybe we should add a method to be queried instead, like gd.supportsXorMode();

Yes, like other instanceof usage, see above.

#102 - 07/28/2017 08:45 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11191.

This is the notes resolution and fix for web client mouse move handling issue. The difference for web client is the mouseActions() return value is important for web client to handle mouse events, while swing client works not considering this. Continue debugging the web client. There are the mouse pointer change and dotted rectangle painting issues there.

#103 - 07/31/2017 08:43 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

PaneEntity: in setGridFactorHorizontal: what happens to vertical factor if setting it to 1, but it was already 1. Same 'mirror' question for setGridFactorVertical;

Nothing, if it is already 1 we do not need to change it. Or I misunderstood your question.

Let me add some detail, with numeric example: suppose that we have config.gridFactorHorizontal = 1 and config.gridFactorVertical = 2. Calling at this moment setGridFactorHorizontal(1) will result in a no-op, because gridFactorHorizontal is already 1, so the content of the block of the outer if is not executed and the value of gridFactorVertical remains 2. However, I wonder if, in this case, gridFactorVertical should not be set to 1, too.

#104 - 07/31/2017 09:25 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Eugenie Lyzenko wrote:

PaneEntity: in setGridFactorHorizontal: what happens to vertical factor if setting it to 1, but it was already 1. Same 'mirror' question for setGridFactorVertical;

Nothing, if it is already 1 we do not need to change it. Or I misunderstood your question.

Let me add some detail, with numeric example: suppose that we have config.gridFactorHorizontal = 1 and config.gridFactorVertical = 2. Calling at this moment setGridFactorHorizontal(1) will result in a no-op, because gridFactorHorizontal is already 1, so the content of the block of the outer if is not executed and the value of gridFactorVertical remains 2. However, I wonder if, in this case, gridFactorVertical should not be set to 1, too.

Now I see, thanks. Yes, this is the point to clarify. I need to clarify this case in 4GL to find the real behavior. Then make appropriate changes in FWD.

#105 - 07/31/2017 11:12 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Let me add some detail, with numeric example: suppose that we have `config.gridFactorHorizontal = 1` and `config.gridFactorVertical = 2`. Calling at this moment `setGridFactorHorizontal(1)` will result in a no-op, because `gridFactorHorizontal` is already 1, so the content of the block of the outer if is not executed and the value of `gridFactorVertical` remains 2. However, I wonder if, in this case, `gridFactorVertical` should not be set to 1, too.

The investigated 4GL handling is a bit different than I thought. I have created testcase `drag_n_drop/dnd_test4_2.p` - bzd repo updated to 1630. The results are pretty strange.

1. The vertical and horizontal factors are independent as attribute values. It is NOT synchronized on 1 value. Possible to set 1 to one and any other to next value. For example 1 for horizontal and 2 for vertical, setting 1 to horizontal does not set 1 to vertical to be in sync.
2. But if at least one of the factor is 1 - the grid is displaying as if both factors are 1 (however the actual attribute values can be 1 for horizontal and 3 for vertical or vice versa).

So I need to change the implementation accordingly. We have different handling on server and client sides.

#106 - 07/31/2017 06:37 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11192.

The update has fix for grid factor handling. Now like in 4GL both vertical and horizontal attribute vales are independent and can have arbitrary values more than 0. The client side painter respectively display the grid with 1x1 if at least one of the grid factor is 1. Also the update has change for `ScrollPaneGuilmpl` with adding move and drag events to mouse action to be used in web client.

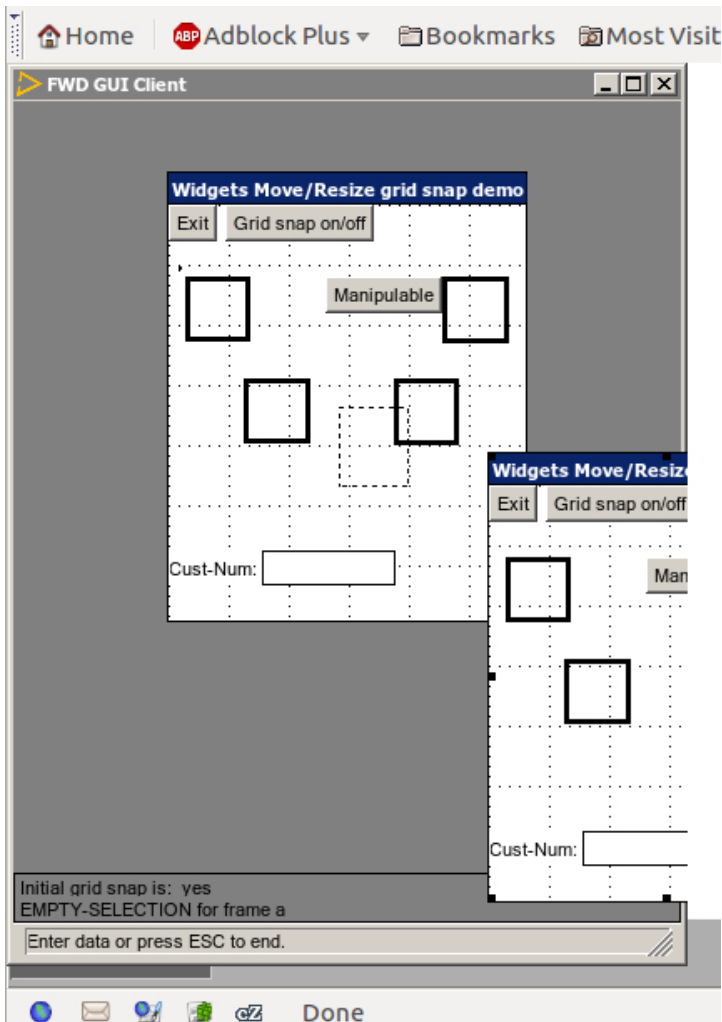
#107 - 07/31/2017 10:08 PM - Eugenie Lyzenko

- *File `dnd_test0_1_fwd_web_20170731a.jpg` added*

The task branch 1834a has been updated for review to revision 11193.

Just a some code cleanup to remove some commented out lines. The currently web client debugging is in progress. There are two issues in the web client to fix for now:

1. The mouse pointer is not properly refreshed for `BrowseColumnGuilmpl` widget, while it is not a problem to set new pointer for a `FrameGuilmpl` widget. And this is a bit strange.
2. The mouse drag(box selection mode) works bad in web client frame widget. The work area(canvas) become completely corrupted with new cloned phantom frame and widgets. The sample picture is below:



This is reaction for mouse release event. If someone has a short answer for what is probably going on here(I guess inside JavaScript code) - please share the knowledge.

#108 - 08/01/2017 11:55 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11194.

The fix for wrong mouse cursor change in row resize with BrowseColumnGuilmpl widget and web client. The root cause was insufficient deep in mouse hover handling set up for browse widget. We need to install the hover handler for browse columns widgets too to be able to change the pointer for browse column widget in web client. Also we need to disable getting custom pointer override for BrowseColumnGuilmpl for proper functionality of the hover action handler, otherwise mouse cursor change will not be possible in web client.

The fix allows to clean up a bit the direct manipulation handler code.

Continue working with web client debugging, the next subtask is to find out why the box selection causes the window/frame moving/corruption instead of drawing the dotted rectangle and widget selection.

#109 - 08/02/2017 11:44 PM - Eugenie Lyzenko

Some debugging results for web clients. The investigation highlights another issue with direct manipulation code. When mouse extended button is clicked on selected frame it must be deselected but currently not. Now fixed. The main web issue is related to full container repainting code. We certainly need to implement the XOR replacement procedure for web client. Thinking about possible alternatives.

Also the box selection mode does not work properly in web client due to possible event race conditions or improper ordering. The widget within box selection is having the select event but then something is happening causing the widget again deselecting with frame container selection. Continue debugging.

#110 - 08/03/2017 06:45 PM - Eugenie Lyzenko

I'm not an initial author of the JavaScript side so while web client debugging I have got two questions regarding JavaScript canvas procedure implemented:

1. The p2j.canvas_renderer.js has two functions: drawRect and strokeRect. Both have absolutely the same implementation. Why do we need two function names?
2. My understanding of the p2j.strokes.js implementation is the dotted line is drawing via 1x1 image of single pixel within the length of the line(if the coordinate is not in a gap between dashes). We use putImage() operating with single pixel to draw the line. Is it correct understanding.

Sergey, can you clarify, looks like you have designed this initially, right?

#111 - 08/03/2017 07:06 PM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

I'm not an initial author of the JavaScript side so while web client debugging I have got two questions regarding JavaScript canvas procedure implemented:

1. The p2j.canvas_renderer.js has two functions: drawRect and strokeRect. Both have absolutely the same implementation. Why do we need two function names?

Yes, according to their usages

```
case ops.DRAW_RECT:
case ops.FILL_RECT:
  x      = p2j.socket.readInt32BinaryMessage(message, idx + 1);
  y      = p2j.socket.readInt32BinaryMessage(message, idx + 5);
  width  = p2j.socket.readInt32BinaryMessage(message, idx + 9);
  height = p2j.socket.readInt32BinaryMessage(message, idx + 13);
  filled = (type === ops.FILL_RECT);
  extra  = " x = " + x + "; y = " + y + "; width = " + width + "; height = " + height;
  this.canvasRenderer.strokeRect(x, y, width, height, this.canvasRenderer.rawColor, filled);
  break;
```

these two names exist only because it seems that stroke should not have filled parameter but draw should have this parameter. Thus these two functions were merged in one and there are two synonyms now.

2. My understanding of the p2j.strokes.js implementation is the dotted line is drawing via 1x1 image of single pixel within the length of the line (if the coordinate is not in a gap between dashes). We use putImage() operating with single pixel to draw the line. Is it correct understanding.

It seems that you are correct. Dotted lines are drawn by pixels, but usual lines are drawn by images due to the optimization of pixels drawing.

#112 - 08/03/2017 07:10 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie Lyzenko wrote:

I'm not an initial author of the JavaScript side so while web client debugging I have got two questions regarding JavaScript canvas procedure implemented:

1. The p2j.canvas_renderer.js has two functions: drawRect and strokeRect. Both have absolutely the same implementation. Why do we need two function names?

Yes, according to their usages

[...]

these two names exist only because it seems that stroke should not have filled parameter but draw should have this parameter. Thus these two functions were merged in one and there are two synonyms now.

Another word the this.canvasRenderer.drawRect() is not used anymore?

#113 - 08/03/2017 08:40 PM - Sergey Ivanovskiy

No, there is a debug code in `CanvasRenderer.prototype.drawText = function(text, x, y, centered)` that uses this function.

#114 - 08/03/2017 11:30 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11195.

The update has small fixes and cleanup for event processing in web client. The issue caused by web client events strange behavior when mouse press-drag-release sequence generates additional mouse click event. This is not happening in Swing client and not regular event handling I guess.

Continue working on web client selection box issue. Looks close to solution. The idea is if line is drawing by 1x1 pixel image - we can do required transformation for pixel to invert the current pixel color to be able to restore original color by drawing twice.

#115 - 08/04/2017 10:44 PM - Eugenie Lyzenko

After investigation and testing I have got the selection box implementation approach. The general call will be abstracted in `GuiDriver`. The implementation will be different for swing and web client. The swing client will use the current approach with `java2d setXORMode()` usage. The investigation shows the XOR mode does not work in JavaScript based web client. So we need different approach for web client. I'm planning to use new drawing operation `DRAW_SELECTION`. The JavaScript code will save the current canvas content, restore it over previous selection box and draw new selection box. The implementation is in progress. Expecting the approach to be ready in a next working day.

#116 - 08/07/2017 02:55 PM - Eugenie Lyzenko

Some debugging/implementation results. I'm trying to implement selection box clean up code. Also to understand why XOR composition does not work.

The first thing is the bug in current `p2j.canvas_renderer.js` code:

```
CanvasRenderer.prototype.applyCompositeMode = function(imageData, x, y)
{
  if (this.compositeModeOn)
  {
    var w = imageData.width;
    var h = imageData.height;
    var screenShot = this.ctx.getImageData(x, y, w, h);
    var pixelsInBytes = 4 * w * h;
    for (var i = 0; i < pixelsInBytes; i += 4)
    {
      imageData[i] = imageData[i] ^ screenShot[i];
      imageData[i + 1] = imageData[i + 1] ^ screenShot[i + 1];
      imageData[i + 2] = imageData[i + 2] ^ screenShot[i + 2];
      imageData[i + 3] = imageData[i + 3] ^ screenShot[i + 3] ^ 255;
    }
  }
};
```

This code should perform composition for images to be put on the canvas. The bug is:

```
imageData[i] = imageData[i] ^ screenShot[i];
```

must be replaced with:

```
imageData.data[i] = imageData.data[i] ^ screenShot.data[i];
```

Not sure yet what to do with alpha channel, may be we do not even need to touch it.

The other wrong thing here and in other code is:

```
var screenShot = this.ctx.getImageData(x, y, w, h);
```

all data items in returning array(screenShot.data[0-3]) are 0. This makes transformations above useless. We are not getting XOR-ed image as result. So the question: why getImageData() does not return correct color value for the current state of the canvas?

#117 - 08/07/2017 03:02 PM - Constantin Asofiei

Eugenie, look into p2j.screen.js, a canvasRenderer defined for a window - this works with an 'offscreen' context and canvas, as drawing is not performed on the real screen - the 'offscreen' is flushed to real screen once the drawing is complete, see the call for this.canvasRenderer.clear(); at the end of Window.prototype.draw.

Now, your problem might be because you either did not call a repaint for the affected area - so the offscreen canvas is blank as it was never drawn - or the drawing has not been performed yet, but will be performed later.

#118 - 08/07/2017 11:54 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11196.

The update has implemented selection box drawing for web client. The implementation is based on fixed XOR mode functionality. The idea is: drawing strokeRect() twice on the same location/size restores the original picture. To be able to get screen capture the actual context is now passing to CanvasRenderer constructor to have the reference to get screen data. Thanks for hint to Constantin.

Due to implementation of the dotted rectangle by set of the 1x1 pixel image drawing the current approach is far away from optimal on my guess and can produce some visual artifacts(like remaining parts of previously painted dotted rectangles). But XOR mode is working now for images. May be we can use solid rectangle instead of dotted in web client to optimize the CPU usage in web browser.

Need additional testing for other code that uses selection box approach(widgets move, resize, row resize - all of them are using the same XOR-ed rectangle in swing client and need to be tested in web client).

Also I have found it is pretty difficult to debug JS code. After change and rebuild - the new version not always become visible in Firefox web debugger. And this is strange a bit.

#119 - 08/08/2017 05:43 AM - Greg Shah

After change and rebuild - the new version not always become visible in Firefox web debugger.

Sometimes you have to clear your browser's cache to pick up the changes.

#120 - 08/08/2017 06:40 AM - Ovidiu Maxiniuc

Greg Shah wrote:

After change and rebuild - the new version not always become visible in Firefox web debugger.

Sometimes you have to clear your browser's cache to pick up the changes.

See here how to quickly do this, depending on the browser you use: https://en.wikipedia.org/wiki/Wikipedia:Bypass_your_cache. I made a habit of using CTRL+F5, since I have some refresh issues with redmine on Firefox.

#121 - 08/08/2017 11:25 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11197.

This update fixes the potential issues with artificial drag event creation. The drag event is posting after 100 ms on mouse move with the button down. This means if we stop moving or release mouse button the mouse up event is generating. But if it is happening withing 100 ms there is possibility the pending mouse drag event will be generating after mouse up event. And this is certainly not correct and can cause some bugs/artifacts because the mouse drag cycle is strictly predefined.

Also I have increased the timeout for mouse drag generation to relax a bit the mouse processing loop for web client. Continue testing the selection boxes in different modes. And planning to rebase with recent trunk as next step.

#122 - 08/09/2017 12:29 PM - Eugenie Lyzenko

The task branch 1834a has been rebased with trunk 11157. New revision is 11198.

#123 - 08/10/2017 10:13 AM - Eugenie Lyzenko

The additional investigation is in progress to have better approach for javascript based web client selection box painting. I want to try the solution with additional canvas on the top of the client window with only selection box content on transparent background(coming from [note 64](#)) if no objections. The advantage is we do not need to clean old rectangle, no need to use XOR mode, just clear peer canvas and paint new rectangle not involving base client window content.

The current 1x1 pixel image based approach is not perfect and this bothers me.

#124 - 08/10/2017 06:34 PM - Greg Shah

Does 1834a support DROP-TARGET as a browse option and as a widget option? If not, please add it.

#125 - 08/10/2017 07:17 PM - Eugenie Lyzenko

Greg Shah wrote:

Does 1834a support DROP-TARGET as a browse option and as a widget option? If not, please add it.

OK. Yes, it is supported as widget attribute and option(define widget) on conversion level. The implementation is limited to server side attribute value check/set. The client side was not implemented yet.

#126 - 08/10/2017 07:24 PM - Constantin Asofiei

Greg/Eugenie: for DROP-TARGET, documentation states that only OS files can be dropped, and according to <http://knowledgebase.progress.com/articles/Article/P82639> it works in conjunction with the DROP-FILE-NOTIFY event. Can we implement this at both Web and Swing drivers?

Also, do we have real app usage of DROP-FILE-NOTIFY event? Without this, I think DROP-TARGET will be just a no-op.

#127 - 08/10/2017 11:34 PM - Eugenie Lyzenko

Implementing alternative approach for drawing selection box canvas. The point with issues is new canvas is layered behind the main window canvas instead on top of it. The changing canvas.style.zIndex to mainWindowCanvas.style.zIndex + 1 does not fix the issue. Continue working.

#128 - 08/11/2017 02:22 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Implementing alternative approach for drawing selection box canvas. The point with issues is new canvas is layered behind the main window canvas instead on top of it. The changing canvas.style.zIndex to mainWindowCanvas.style.zIndex + 1 does not fix the issue. Continue working.

zIndex should help in this case. For an example of the code you can look at MouseResizable() in p2j.mouse.js. You can check what is the actual zIndex value of this new created canvas by the Inspector tool from the browser "Developer" menu in Firefox or another browser.

#129 - 08/11/2017 03:07 AM - Sergey Ivanovskiy

Please check this style canvas.style["position"] = "fixed";. It can be important for z-order layout of canvas elements from the same group having "fixed" position.

#130 - 08/11/2017 08:46 AM - Greg Shah

do we have real app usage of DROP-FILE-NOTIFY event? Without this, I think DROP-TARGET will be just a no-op.

This should be checked. I would have expected that event without the event, the NUM-DROPPED-FILES and GET-DROPPED-FILES would return something.

Anyway, we want a matching implementation to how the 4GL does it.

#131 - 08/11/2017 09:20 AM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Please check this style `canvas.style["position"] = "fixed";`. It can be important for z-order layout of canvas elements from the same group having "fixed" position.

Yes, this is a key point. Without this the new canvas is still behind the main window even with proper `zIndex` value. Thanks for help.

#132 - 08/11/2017 09:42 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

Greg/Eugenie: for DROP-TARGET, documentation states that only OS files can be dropped, and according to <http://knowledgebase.progress.com/articles/Article/P82639> it works in conjunction with the DROP-FILE-NOTIFY event. Can we implement this at both Web and Swing drivers?

Also, do we have real app usage of DROP-FILE-NOTIFY event? Without this, I think DROP-TARGET will be just a no-op.

DROP-FILE-NOTIFY event is coming to the widget that has `DROP-TARGET == TRUE`. As default processing this event executes `END-FILE-DROP()` method to clean up resources used in this drag-n-drop operations. We need to 1. Detect the 4GL behavior and 2. Implement this in FWD(keeping event/triggers processing untouched).

I think even if there are no 4GL code now that uses it in applications we are working with - we have to create testcases that will use it. It is not so difficult because the only object the DnD is operating with are the file names. No binary data transfers.

As to web/swing drivers specific implementation - yes, some complexity is expected to handle drop operations that started outside the FWD. I guess we need to have client neutral approach in ideal. Like what I'm trying to implement for direct manipulation.

#133 - 08/11/2017 09:49 PM - Eugenie Lyzenko

- File sel_box_web_20170811b.jpg added

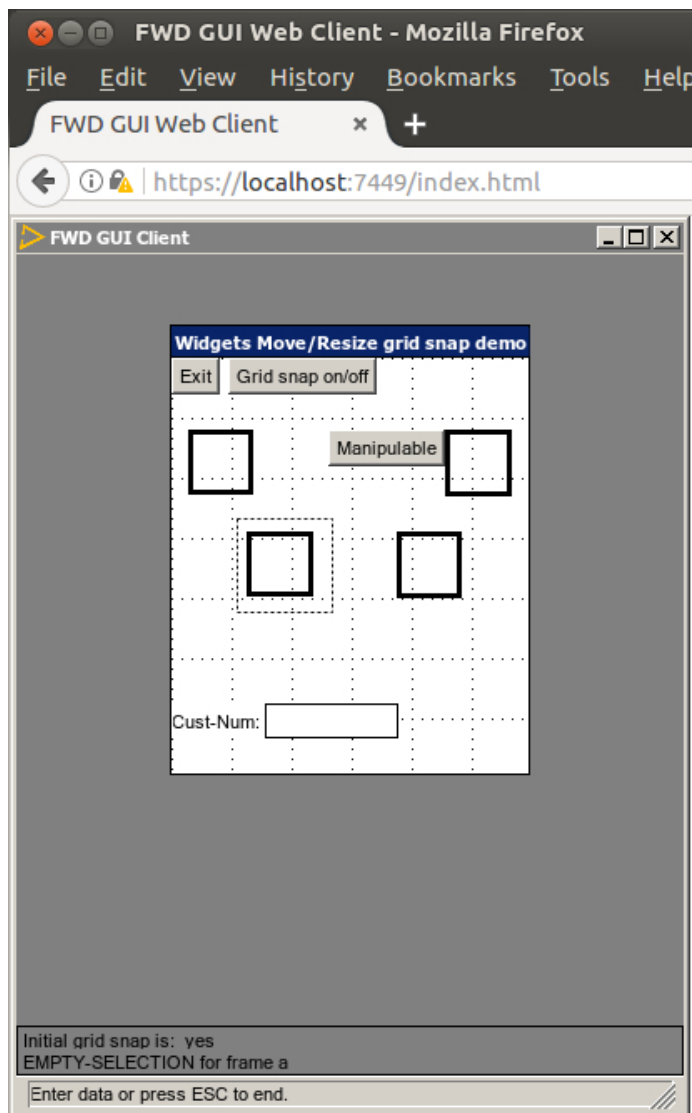
- File sel_box_web_20170811a.jpg added

The task branch 1834a has been updated for review to revision 11199.

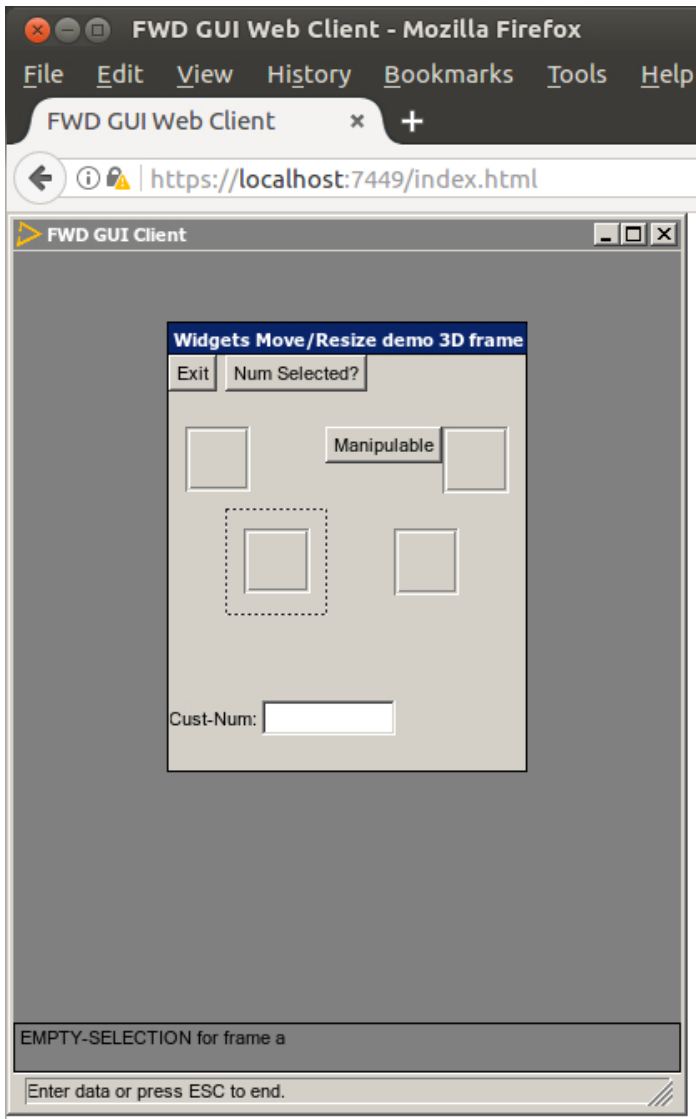
This update introduces new approach for selection boxes painting in web client. This is completely different comparing to swing client. We have special transparent canvas on the top of the window and all selection rectangles are rendering into this canvas using direct rectangle paintings. No XOR mode used in web client, no need to change regular canvas's stroke style. Instead the JS side has primitive drawSelection(..., clear) with boolean clear option that means if TRUE clear all selection canvas(if it is dirty) instead of drawing. Dirty means some selection box has been painted on. If canvas is not dirty, the clean mode does nothing. When the dragging is finished the special call is executed to clean up selection canvas and make it hidden. The z-index ordering is z-index of the top window + 10. This means I have left some room for other canvas to be on top of the window but behind the selection canvas.

About selection rectangle style. There is the base color(usually black but not mandatory) that must be set as current for selection box rectangles. To be the rectangle always visible two rectangles have been painting. The first is the solid one with color inverted to the current color. Then the dotted([2,2] in JS terms) rectangle is drawing over solid with regular color. This way we have the dotted rectangle that has 50% of the color and 50% of the inverted color. So the rectangle will always be visible even background has colored with selection rectangle color or it's inversion.

Here is the sample picture for regular frame with white background:



And here is the 3D frame with pale background:



Also I have made small fixes in p2j.mouse.js syntax errors. Please check and confirm the changes are correct.

#134 - 08/14/2017 01:00 PM - Sergey Ivanovskiy

It seems that the web client changes are correct, there are minor issues that CanvasRenderer(canvas, ctx, screenCanvas, strokesManager, fontsManager, logger) misses js docs for new parameter screenCanvas and the changes in src/com/goldencode/p2j/ui/client/gui/driver/web/res/p2j.mouse.js are correct but they are related to the code style because ; can be missed in these cases. I think there were missed by a mistake and not intentionally.

Eugenie, thanks for the fix of CanvasRenderer.prototype.applyCompositeMode = function(imageData, x, y), the first version of this function wasn't tested at all.

#135 - 08/14/2017 10:06 PM - Eugenie Lyzenko

The remaining issues/TODOs for direct manipulation and drag-n-drop functionality:

1. The frame move/resize functionality implementation.
2. Implement file-list drag-and-Drop.

Currently I have fixed the web client issue for row resize and other direct manipulation modes. Again it was related to mouse drag events artificial nature. We need to keep track on this additionally.

Now next work for remaining branch preparation is to fix the frame move and resize within current window. The logic to implement is petty clean, the point is to debug why selection box become clipped for areas outside the frame even with clipping is off. Continue working. There is no code updates for today.

#136 - 08/15/2017 03:30 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11200.

This is code clean up preparing to be released. Also the web client row resize issue has been fixed. Some refactoring during clean up to properly exclude space/skip/label from different logic not involving instanceof operator.

Continue with adding gap analysis change to prepare update for final review. Expected to be ready later today.

#137 - 08/15/2017 04:35 PM - Eugenie Lyzenko

The task branch 1834a has been updated for review to revision 11201.

Added gap analysis changes. So it is ready for final review as candidate.

#138 - 08/15/2017 05:30 PM - Greg Shah

Code Review Task Branch 1834a Revision 11219

I like this implementation. It is especially nice that you have been able to put so much of the client implementation into MouseDirectManipulation.

1. The attributes kw_movable, kw_resizable should be set to rt_lvl_partial instead of rt_lvl_full_rest.
2. DirectManipulationInterface should just be called DirectManipulation.
3. DroppableInterface should just be called Droppable.
4. Why was BrowseColumnGuiImpl.getEffectiveMousePointer() removed?
5. Are you sure that FrameGuiImpl draws the same for all Windows themes?

- File dnd_test0_1_4gl_win10.jpg added

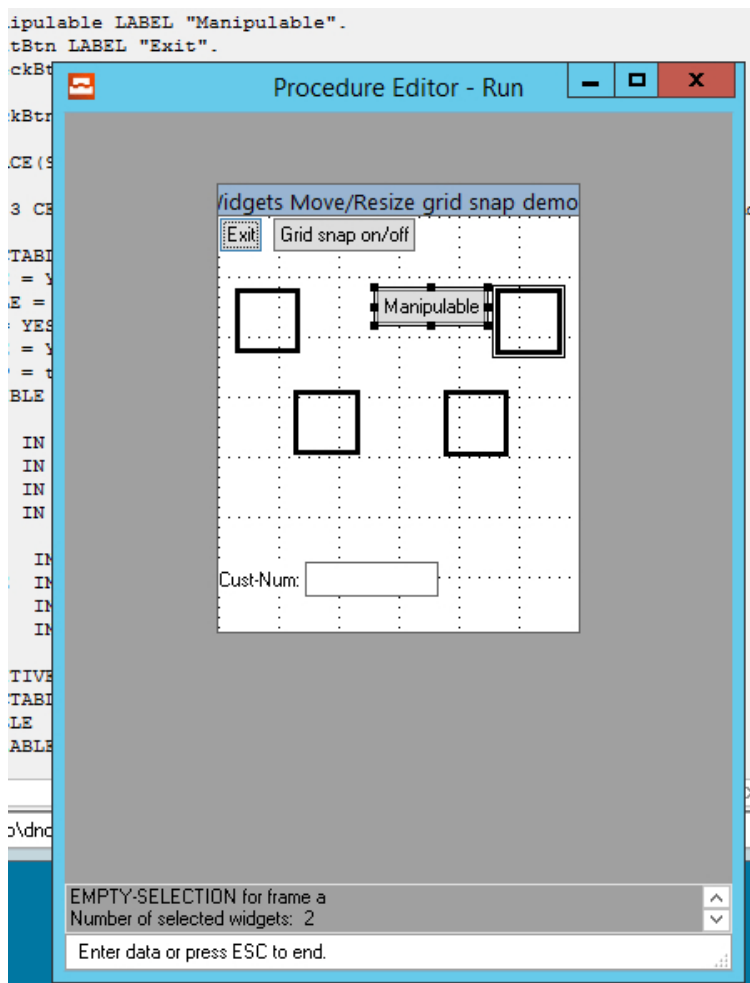
Greg Shah wrote:

4. Why was BrowseColumnGuiImpl.getEffectiveMousePointer() removed?

This method always returning default pointer does not allow to change pointer for browse column widget. On the other hand the pointer must be changed while browse row resizing is starting(mouse is over the resize area of the browse widget).

5. Are you sure that FrameGuiImpl draws the same for all Windows themes?

Do you mean the shape of the selection boxes for selected widgets? If yes - the drawing is the same for both classic and windows10 themes:



The classic themes pictures can be found above.

Or do you mean something different?

#140 - 08/15/2017 07:42 PM - Eugenie Lyzenko

Greg Shah wrote:

Code Review Task Branch 1834a Revision 11219

I like this implementation. It is especially nice that you have been able to put so much of the client implementation into MouseDirectManipulation.

1. The attributes kw_movable, kw_resizabl should be set to rt_lvl_partial instead of rt_lvl_full_rest.
2. DirectManipulationInterface should just be called DirectManipulation.
3. DroppableInterface should just be called Droppable.

The task branch 1834a has been updated for review to revision 11202.

The notes 1-3 above resolution.

#141 - 08/16/2017 07:22 AM - Greg Shah

Code Review Task Branch 1834a Revision 11202

It all looks good.

Go ahead with testing ChUI (conversion and runtime) and GUI (standalone tests, Hotel GUI and the customer application GUI).

#142 - 08/16/2017 07:30 AM - Eugenie Lyzenko

Greg Shah wrote:

Code Review Task Branch 1834a Revision 11202

It all looks good.

Go ahead with testing ChUI (conversion and runtime) and GUI (standalone tests, Hotel GUI and the customer application GUI).

OK. Started.

#143 - 08/16/2017 04:41 PM - Eugenie Lyzenko

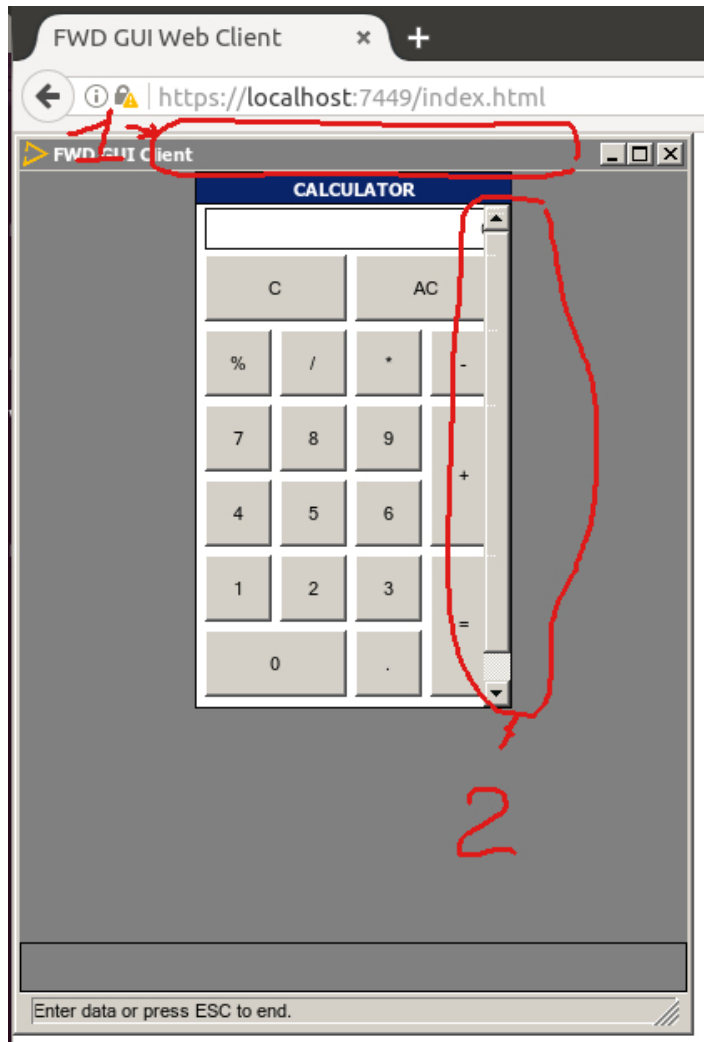
Conversion testing passed. Generated codes are identical. Continue with ChUI runtime testing and other tests.

#144 - 08/16/2017 09:24 PM - Eugenie Lyzenko

- File test_issue0.jpg added

The standalone testing is in progress.

There is an issue, not regression and we have it in the trunk:



This is the web client that has not active color window title. The second issue is layout mismatch giving vertical scrollbar that should not be here.

#145 - 08/17/2017 07:42 AM - Eugenie Lyzenko

The ChUI regression testing completed. The results: 1834a_11202_dfff287_20170817_evl.zip. No regressions. GUI test did not show any regressions too. Can be merged into trunk I think.

#146 - 08/17/2017 07:44 AM - Greg Shah

Can be merged into trunk I think.

Please merge 1834a to trunk.

#147 - 08/17/2017 07:48 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

This is the web client that has not active color window title.

This is because the mouse-in event is not received by the JS. There is a similar issue for swing. I wrote a pure-swing test (just display mouse (move) events) and some mouse-in and mouse-out are not dispatched to listeners.

The second issue is layout mismatch giving vertical scrollbar that should not be here.

This does not look like the calculator.p from testcases. Please let me know where I can find it. I would like to investigate the layout issue.

#148 - 08/17/2017 07:51 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

The second issue is layout mismatch giving vertical scrollbar that should not be here.

This does not look like the calculator.p from testcases. Please let me know where I can find it. I would like to investigate the layout issue.

demo/calc-static-chars.p and /demo/calc-static-pixels.p.

#149 - 08/17/2017 07:56 AM - Eugenie Lyzenko

Greg Shah wrote:

Can be merged into trunk I think.

Please merge 1834a to trunk.

OK. Starting.

#150 - 08/17/2017 08:02 AM - Eugenie Lyzenko

The branch 1834a has been merged into trunk as revision 11158. The branch 1834a has been archived.

#151 - 08/17/2017 08:44 AM - Eugenie Lyzenko

Created task branch 1834b from trunk revision 11158.

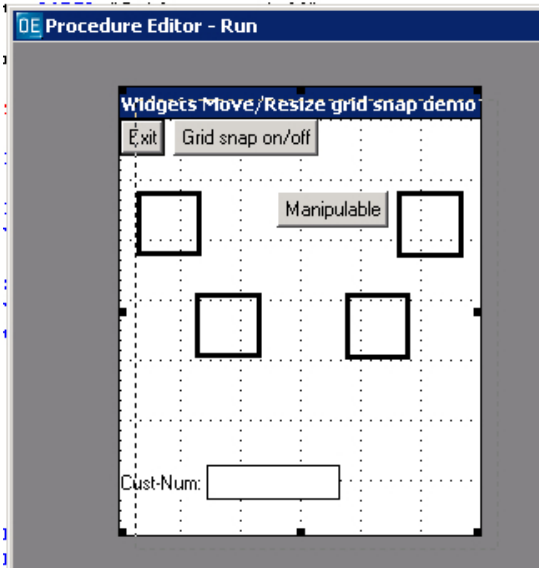
#152 - 08/17/2017 10:08 PM - Eugenie Lyzenko

Start working on the part 2. Continue with frame move/resize. The first issue I have faced is the current selection box drawing approach for swing client does not work on the window dark pale (127,127,127) background. The reason is the inverted color is the same as background color so the selection rectangle part on the window background become invisible. Working on solution.

#153 - 08/18/2017 12:05 PM - Eugenie Lyzenko

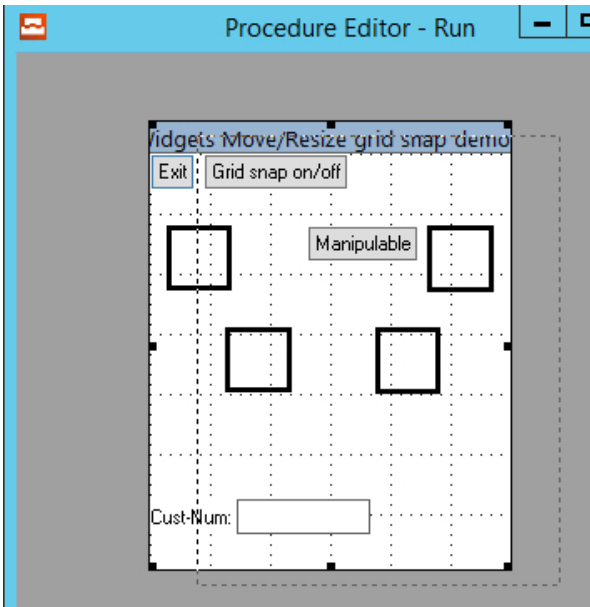
- File *set_box_on_darkgray_4gl_win10.jpg* added
- File *set_box_on_darkgray_4gl_classic.jpg* added
- File *set_box_on_darkgray_fwd_classic.jpg* added

4GL classic theme:



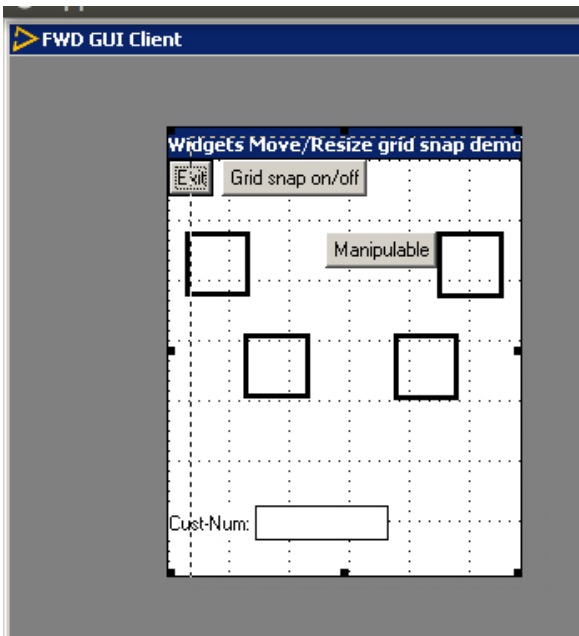
Background: 0x848285, selection: 0x7b7d7a, this is XOR operation with white addon.

4GL win10 theme:



Background: 0xa0a0a0, selection: 0x5f5f5f, this is XOR operation with white addon.

FWD classic theme:



Background: 0x808080, selection: 0x7f7f7f, this is XOR operation with white addon.

So we are correctly making selection color draw and if the picture is good, there is nothing to fix here(exclude possible background mismatch).

Note: it is better to scale view up to 800% to better see.

#154 - 08/18/2017 12:47 PM - Eugenie Lyzenko

I need some clarification from one who implemented theme management.

Why do we use:

```
pal.put (COLOR_APPWORKSPACE, new ColorRgb(128, 128, 128));
```

instead of matched color:

```
pal.put (COLOR_APPWORKSPACE, new ColorRgb(132, 130, 133));
```

Any special reason to change default color?

#155 - 08/18/2017 01:32 PM - Ovidiu Maxiniuc

Any special reason to change default color?

For Classic Theme, I tried to keep the existing colors, to avoid regression when implementing the Themes. Note that in classic scheme, the colors can be adjusted manually. for example, the COLOR_APPWORKSPACE can be altered using: Control Panel\All Control Panel Items\Display > Change Color Scheme > Advanced > Application background.

For Win10 and Win8 I used print-screens and copied the colors from Gimp. However, I later realized that Remmina does not display in full 24/32 bit Graphic, even if it is configured so. Maybe because of some compression in protocol, I do not know. I tried to fix some of these colors in 2676b branch (actually merged into 1818a). They are not really visible to naked eye so it's pretty difficult to say which are correct and which are not. The best solution is to use a remote application (the recently deprecated mspaint) to extract the exact RGB value. Working with the new VM (that's a Win8 style in set_box_on_darkgray_4gl_win10.jpg, not Win10), might simplify the capture, with less errors, also.

#156 - 08/18/2017 05:13 PM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Any special reason to change default color?

For Classic Theme, I tried to keep the existing colors, to avoid regression when implementing the Themes. Note that in classic scheme, the colors can be adjusted manually. for example, the COLOR_APPWORKSPACE can be altered using: Control Panel\All Control Panel Items\Display > Change Color Scheme > Advanced > Application background.

OK. I think here there is a way the Windows signals for opened application the systems color palette has been changed. The application receive this message and repaint itself with new colors. To leverage this we have to implement this event handler on native level(p2j.dll must be aware to keep track for color changes).

I'm going to change the classic theme default constant in FWD for application workspace if you do not mind.

For Win10 and Win8 I used print-screens and copied the colors from Gimp. However, I later realized that Remmina does not display in full 24/32 bit Graphic, even if it is configured so. Maybe because of some compression in protocol, I do not know. I tried to fix some of these colors in 2676b branch (actually merged into 1818a). They are not really visible to naked eye so it's pretty difficult to say which are correct and which are not.

I'm using gimp color picker tool to see the value of the color. To take a screenshot it is better to create a picture within system is being testing(for example in Windows print screen and paste into Windows bitmap editor). Then the picture is saving locally and downloading as file.

The best solution is to use a remote application (the recently deprecated mspaint) to extract the exact RGB value. Working with the new VM (that's a Win8 style in set_box_on_darkgray_4gl_win10.jpg, not Win10), might simplify the capture, with less errors, also.

Yes, this is what I'm doing too. The VM with Windows 2012 we currently have is a great help of course.

#157 - 08/18/2017 05:29 PM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

OK. I think here there is a way the Windows signals for opened application the systems color palette has been changed. The application receive this message and repaint itself with new colors. To leverage this we have to implement this event handler on native level(p2j.dll must be aware to keep track for color changes).

I don't think we will do this kind of support. You should not bother about this.

I'm going to change the classic theme default constant in FWD for application workspace if you do not mind.

Please do. As noted above, I have some changes for Win8, too.

I'm using gimp color picker tool to see the value of the color. To take a screenshot it is better to create a picture within system is being testing(for example in Windows print screen and paste into Windows bitmap editor). Then the picture is saving locally and downloading as file.

Just use full-color PNG for this task, JPEGs are not useful for this task.

Yes, this is what I'm doing too. The VM with Windows 2012 we currently have is a great help of course.

I think the VM has the main advantage that it renders directly to screen, and it does not need video compression. I guess it's pretty safe to capture the colors directly with gimp's picker, in this case.

#158 - 08/18/2017 08:14 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11159.

This adds frame widget moving direct manipulation code implementation and some preparation steps for frame resize implementation. The resizing is under active investigation and development so the approach can be significantly changed. Resizing is a bit complex due to resize handles are related to frame part that are not visible (neither frame title nor scroll pane). Continue working with implementing appropriate design.

#159 - 08/21/2017 07:54 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11160.

This update finishes the frame widget direct manipulation move/resize operation implementation. Tested for both Swing and Web clients. So the first phase of the drag and drop implementation has been completed.

The next step is to implement file list related drag and drop operations. First I need to investigate requirement in details and prepare implementation plan.

#160 - 08/22/2017 10:06 PM - Eugenie Lyzenko

The investigation performed to make implementation plan:

1. The good news is we can use embedded Java functionality to implement file list drag and drop. This give us good opportunity to be OS independent and do not implement OS specific native code. The good starting tutorial can be found here: <http://docs.oracle.com/javase/tutorial/uiswing/dnd/intro.html>
2. The key point for file list handling is to install custom TransferHandler for JFrame or JDialog. These two java classes we use in EmulatedWindowState extending client window emulator classes.

So the suggested implementation plan on the first look is the following:

1. Install TransferHandler for container that is our link to Java Swing classes.
2. When registered callback is executing - we need to check the mouse pointer coord, lookup the widget component and if the widget has DROP-TARGET attribute as true - we post the message DROP-FILE-NOTIFY
3. The special handler class will be introduced (say FileDropHelper) to server all internals related to file list transfers. The client will ask it for actual values for attribute NUM-DROPPED-FILES and for method result GET-DROPPED-FILE
4. No need to create new helper for every new drop target. There can be only one drop operation per system (Java VM the client is running on)
5. In ideal case the implementation will be client independent, meaning no web or swing client specifics. Or at least minimum.

This is the first view of the approach so changes are possible.

#161 - 08/23/2017 09:15 AM - Greg Shah

The key point for file list handling is to install custom TransferHandler for JFrame or JDialog. These two java classes we use in EmulatedWindowState extending client window emulator classes.

I think the plan will only work for the Swing client. Make sure to implement the interfaces in a generic way so that the driver can hide the Swing or web-specific portions. Everything else can be common code.

#162 - 08/23/2017 09:53 AM - Eugenie Lyzenko

Greg Shah wrote:

The key point for file list handling is to install custom TransferHandler for JFrame or JDialog. These two java classes we use in EmulatedWindowState extending client window emulator classes.

I think the plan will only work for the Swing client. Make sure to implement the interfaces in a generic way so that the driver can hide the Swing or web-specific portions. Everything else can be common code.

Yes, the description is for Swing client. But concept should be similar for both. I did worry a bit with JavaScript but fortunately we have a tools to implement what we need:

<https://www.html5rocks.com/en/tutorials/dnd/basics/>

<https://www.html5rocks.com/en/tutorials/file/dndfiles/#toc-selecting-files-dnd>

This give us opportunity to properly initiate the DROP-FILE-NOTIFY event which is key point to start dropped file list processing. The approach can be:

1. JavaScript handler installed in p2j.screen.js gets the end drop event and calls p2j.socket.js to send message
2. p2j.socket.js creates the message and send it to the Java side, GuiWebSocket code.
3. GuiWebSocket execute callback implemented in GuiWebDriver which in turn accesses EmulatedWindowState
4. EmulatedWindowState is the point where the code become common one.
5. On this level we can use our FileDropHelper class to work with file list objects.

The swing's client TransferHandler is installing in SwingEmulatedWindow and this also accesses to common code in EmulatedWindowState. So we have to implement client specific code due to different ways/paths the client is receiving drop target event(and because the source of this drop event is outside JVM/JavaScript)

Starting development from common code.

#163 - 08/23/2017 10:12 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11161.

This is the partial implementation for common code and swing based specific. The idea is to receive drop event, calculate the widget where the mouse was recently released, check if widget has DROP-TARGET attribute is on and initiate the drop operation. Need to add event generation and handling. Then implement web client specific. Continue working. Will start with current trunk rebase as first step next.

#164 - 08/23/2017 10:32 PM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11159. New revision is 11162.

#165 - 08/23/2017 10:54 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11163.

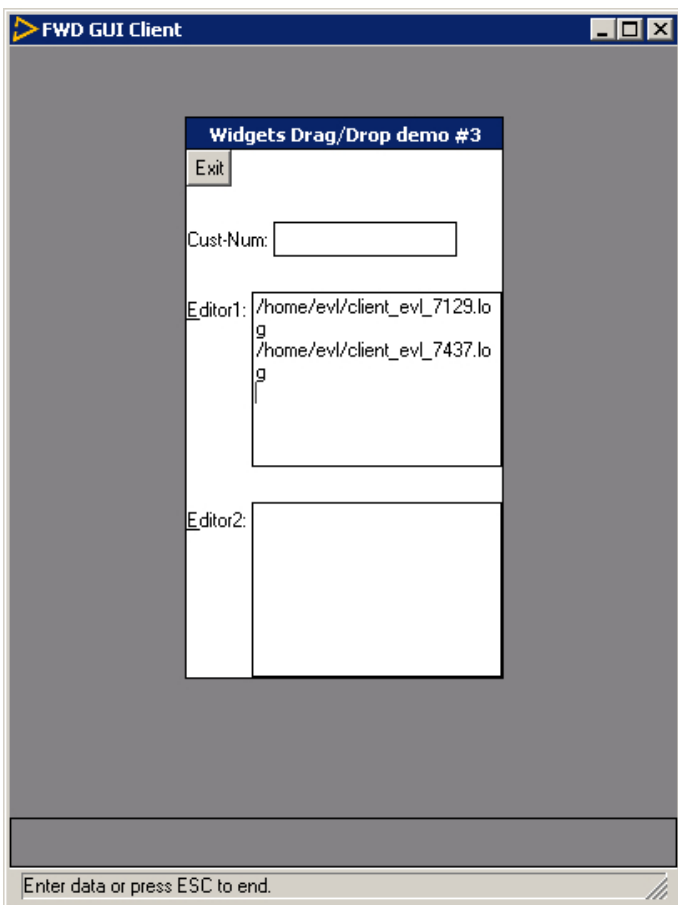
Adding Widget.java change missed in previous update. It is required to properly compile the branch.

#166 - 08/24/2017 11:01 AM - Eugenie Lyzenko

- File *dnd_test3_fwd_20170824a.jpg* added

The task branch 1834b has been updated for review to revision 11164.

This is working drag-and-drop implementation for swing client. The short demo:



The code has debug calls. Also need more investigation to check the correct target widget identification with current approach and more general testing. But it works. The approach is event managed. The widget receive DROP-FILE_NOTIFY event, executes any registered trigger and call END-FILE-DROP method to clean up resources.

The branch will be rebased shortly with current trunk and work will continue with debugging and web client specific implementation.

#167 - 08/24/2017 11:11 AM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11160. New revision is 11165.

#168 - 08/24/2017 12:01 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11166.

This has reworked approach to get drop target widget candidate. Existed code to get last entered widget works better than attempt to save mouse release event. Also the common code was moved to common classes. And repaint for dropped widget is required to see the changes immediately.

Start implementing web client specific code.

#169 - 08/24/2017 10:31 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11167.

First steps to get web client drop target feature working. The JavaScript part and OS events receiving completed. The p2j.socket.js is sending message with all required data for Java part to process. Will continue with Java part next.

Some notes for JS part implementation:

1. Had to set up full document as draggable with index.html. Without this the drop operation has null as dataTransfer. Probably it is better to set up draggable more locally inside JS code. But for now I'm not sure where is this place. Need addition investigation.
2. Dropped file names are getting without path. Need to find a way to get the path to match 4gl behavior.

Continue working. Next need to implement Java part of the web client.

#170 - 08/25/2017 08:35 AM - Greg Shah

Dropped file names are getting without path. Need to find a way to get the path to match 4gl behavior.

Consider that the dropped files are on the system that runs the browser. That will sometimes be a different system than the one on which the FWD Java client runs. In production, it will usually be this way. The path name on the browser's system is not what we want to return. When the browser runs on a remote system, those files can't be accessed by the 4GL code.

I think we need to upload those files from the browser system to a tmp directory in the Java client and return the path from the Java client (including that tmp dir). Then any usage of the file name from 4GL code will actually work.

#171 - 08/25/2017 12:05 PM - Greg Shah

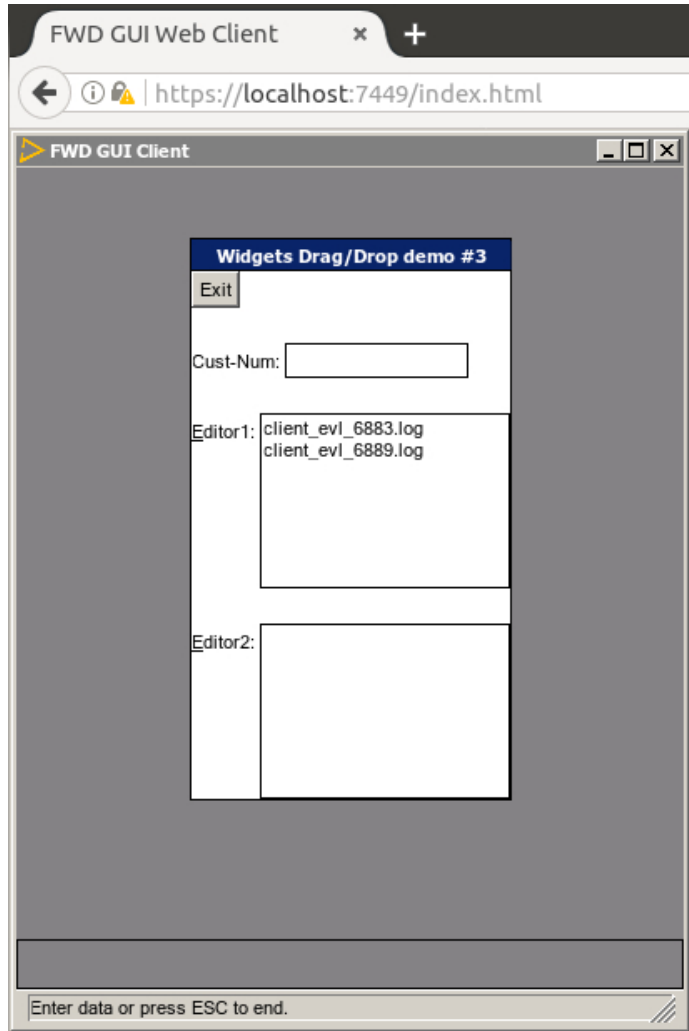
The drop-target attribute is not marked correctly in gap marking. The conversion should be marked as full in the trunk. You will need to update the gap marking in 1834b anyway. Make sure you fix this at the same time.

#172 - 08/25/2017 12:44 PM - Eugenie Lyzenko

- File `dnd_test3_fwd_web_20170825a.jpg` added

The task branch 1834b has been updated for review to revision 11168.

This introduce first working web client implementation of the file list drop handling. There is some issues like changes is not visible after drop until mouse move and code clean up/optimization and other. But main logic is here. The screen:



You can see the path-less files as I noted before. Also the gap missed status for DROP-TARGET is also fixed.

Will rebase with the recent trunk and continue working next.

#173 - 08/25/2017 12:44 PM - Eugenie Lyzenko

Greg Shah wrote:

The drop-target attribute is not marked correctly in gap marking. The conversion should be marked as full in the trunk. You will need to update the gap marking in 1834b anyway. Make sure you fix this at the same time.

OK. Done.

#174 - 08/25/2017 12:46 PM - Eugenie Lyzenko

Greg Shah wrote:

Dropped file names are getting without path. Need to find a way to get the path to match 4gl behavior.

Consider that the dropped files are on the system that runs the browser. That will sometimes be a different system than the one on which the FWD Java client runs. In production, it will usually be this way. The path name on the browser's system is not what we want to return. When the browser runs on a remote system, those files can't be accessed by the 4GL code.

I think we need to upload those files from the browser system to a tmp directory in the Java client and return the path from the Java client (including that tmp dir). Then any usage of the file name from 4GL code will actually work.

OK. Will implement this after current version become stabilized.

#175 - 08/25/2017 12:51 PM - Greg Shah

You can see the path-less files as I noted before.

I don't think this is a problem so long as we can automatically upload the files to the Java client (in a tmp dir). Perhaps some of these techniques will help:

<https://css-tricks.com/drag-and-drop-file-uploading/>

Sergey: please review the link and report your thoughts.

#176 - 08/25/2017 01:31 PM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11161. New revision is 11169.

#177 - 08/25/2017 02:09 PM - Sergey Ivanovskiy

Greg Shah wrote:

You can see the path-less files as I noted before.

I don't think this is a problem so long as we can automatically upload the files to the Java client (in a tmp dir). Perhaps some of these techniques will help:

<https://css-tricks.com/drag-and-drop-file-uploading/>

Sergey: please review the link and report your thoughts.

Ok. There is also the Firefox resource with the concrete examples how to drag and drop files.

https://developer.mozilla.org/en-US/docs/Using_files_from_web_applications

#178 - 08/25/2017 02:26 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Ok. There is also the Firefox resource with the concrete examples how to drag and drop files.

https://developer.mozilla.org/en-US/docs/Using_files_from_web_applications

One note here. We implementing operation on files dragged from OS file system, meaning outside the browser window. The drag is starting not within browser JS code in this case.

#179 - 08/25/2017 02:37 PM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

One note here. We implementing operation on files dragged from OS file system, meaning outside the browser window. The drag is starting not within browser JS code in this case.

It seems that the drag and drop is used exactly in this case when you are dragging the OS file into the target element of the browser window. This element must be able to listen drag and drop events from the supported drag and drop browser. There are several levels of drag and drop api from os api to browser api up to the application handlers, correct?

#180 - 08/25/2017 02:47 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

It seems that the drag and drop is used exactly in this case when you are dragging the OS file into the target element of the browser window. This element must be able to listen drag and drop events from the supported drag and drop browser. There are several levels of drag and drop api from os api to browser api up to the application handlers, correct?

Yes. We are able to get the dropped file names(in recent 1834b).

Now we need to have a way to copy the content of the file from browser JS code to some tmp directory created withing Java client side code. There are two ways to do this:

1. Short way is to use some JS internals already implemented in browser.
2. Long way is to make this upload in FWD, opening file for reading and sending content to Java side via our GuiWebSocket channel.

#181 - 08/25/2017 02:50 PM - Constantin Asofie

Eugenie Lyzenko wrote:

Now we need to have a way to copy the content of the file from browser JS code to some tmp directory created withing Java client side code. There are two ways to do this:

1. Short way is to use some JS internals already implemented in browser.

I don't think is OK to read the entire file content into memory and after that send it to the web socket... more, there is no knowledge if that file will be read or not by the business logic at that time.

1. Long way is to make this upload in FWD, opening file for reading and sending content to Java side via our GuiWebSocket channel.

This one I think is the way to go: read the file in chunks and send it to FWD to save it.

#182 - 08/25/2017 02:51 PM - Sergey Ivanovskiy

My understanding is that file content can be read using the browser's File API, there are several usages of this API in the 3222a and in the web tools. The content can be send to the java client side. Is there another way?

#183 - 08/25/2017 03:00 PM - Sergey Ivanovskiy

Constantin Asofiei wrote:

1. Long way is to make this upload in FWD, opening file for reading and sending content to Java side via our GuiWebSocket channel.

This one I think is the way to go: read the file in chunks and send it to FWD to save it.

File API is asynchronous that we get a completion event that the file content is ready, then we can use the chunk uploads that should work automatically. Please look at p2j.socket.js function send(data). Correct?

#184 - 08/25/2017 03:41 PM - Sergey Ivanovskiy

Although for very large files we should think more. Please look at <https://www.codeproject.com/Articles/830704/Gigabit-File-uploads-Over-HTTP>

#185 - 08/25/2017 03:54 PM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11162. New revision is 11170.

#186 - 08/25/2017 03:56 PM - Greg Shah

Sergey Ivanovskiy wrote:

Although for very large files we should think more. Please look at <https://www.codeproject.com/Articles/830704/Gigabit-File-uploads-Over-HTTP>

I think we can set a maximum upload size. We probably need 2 limits:

- Maximum single file size. Default this to 20MB.
- Cumulative per-user upload amount (all files added together should not exceed this). Default this to 100MB.

Allow the customer to override either or both with directory settings.

#187 - 08/25/2017 04:45 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11171.

Some minor clean up and fix for drop target related event processing. Web client now displays the dropped file names immediately. Continue working.

The question:

What do you think about change in web client index.html:

```
...
<body class="claro" draggable="true" style="font-family:Arial,Helvetica,Verdana,serif;font-size:${font.size}
}px; ">
  <div id="cont" style="margin-left:auto;margin-right:auto;">
...

```

Is it acceptable? Without draggable="true" the file list is null and dropping is not working.

#188 - 08/25/2017 05:56 PM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11163. New revision is 11172.

#189 - 08/28/2017 04:37 AM - Sergey Ivanovskiy

Eugenie, do you need my help to work with FileReader and Blob. I am worried because Blob.slice method has different versions between browsers: blob.mozSlice() for Firefox and blob.webkitSlice() for WebKit. The IE11 requires investigations what is the slice method used by this browser. Please look at <https://developer.mozilla.org/en-US/docs/Web/API/Blob/slice>. If the file size is big enough, then it is required to use Blob.slice on the dropped file object. It seems that <https://www.codeproject.com/Articles/830704/Gigabit-File-uploads-Over-HTTP> has a good idea to use WebWorkers.

#190 - 08/28/2017 12:50 PM - Eugenie Lyzenko

The point to discuss for file uploading approach.

1. Yes, for the cases when web client and Java client part are running on the different systems we need to upload the files to the Java part.
2. On the other hand default processing for 4GL is to do nothing. So the first question is do we really need to upload all files every time it is dropped to the target?

This can overload the web browser making them unresponsive(without additional attempts to do loading in parallel). Or this will produce significant delay between dropping and the moment when the file names/and uploaded files can be ready for web browser.

3. The 4GL working with files is limited (INPUT FROM or OUTPUT TO or LOAD-PICTURE or ...) so the usage of the dropped file is not planned to be often.

So I would suggest to postpone actual file uploading up to moment it become really required to open/load. Then we can start JS FileReader asynchronous usage.

What do you think? Can this approach be used?

#191 - 08/28/2017 04:23 PM - Greg Shah

So I would suggest to postpone actual file uploading up to moment it become really required to open/load. Then we can start JS FileReader asynchronous usage.

I don't think our javascript code will be allowed to access the files later. I think it is possible only during the drop event processing, for security reasons.

I think we must upload as the drops occur.

#192 - 08/28/2017 04:42 PM - Eugenie Lyzenko

Greg Shah wrote:

So I would suggest to postpone actual file uploading up to moment it become really required to open/load. Then we can start JS FileReader asynchronous usage.

I don't think our javascript code will be allowed to access the files later. I think it is possible only during the drop event processing, for security reasons.

I think we must upload as the drops occur.

OK. Understood. Let's go this way.

#193 - 08/28/2017 08:52 PM - Eugenie Lyzenko

Taking into account the parallel processing environments, file size issues and runtime resource limitations the implementation plan is visible as:

1. As soon the dropped files become visible the JS code performs files reading one by one and send the data associated with file name to Java side. Depending on the file size and files amount we can slice them or not.
2. The Java side is taking the data buffer, associated file name and creates the file in temporary location.
3. As soon as all files are uploaded to the Java the final drop file notify is sending to Java side indicating the dropped files are ready to use.
4. When processing of the dropped files completes and the method END-FILE-DROP() is calling the FWD on the Java side deletes uploaded files in temporary location. This point can have another behavior, I think we need to delete the files after usage otherwise the disk space can be exhausted.

Starting implementation from the JS side with some experiments and investigations.

#194 - 08/29/2017 08:48 AM - Greg Shah

4. When processing of the dropped files completes and the method END-FILE-DROP is calling the FWD on the Java side deletes uploaded files in temporary location.

I'm not sure about this. I suspect this is too early (the files might still be in use). Initially, I think we can delete the list of tmp files when the Java client exits. If we find that we need to do something different later, we can improve the approach to give the customer more choices.

#195 - 08/29/2017 10:17 PM - Eugenie Lyzenko

- File *dnd_test3_fwd_web_20170829a.jpg* added

Greg Shah wrote:

4. When processing of the dropped files completes and the method END-FILE-DROP is calling the FWD on the Java side deletes uploaded files in temporary location.

I'm not sure about this. I suspect this is too early (the files might still be in use). Initially, I think we can delete the list of tmp files when the Java client exits. If we find that we need to do something different later, we can improve the approach to give the customer more choices.

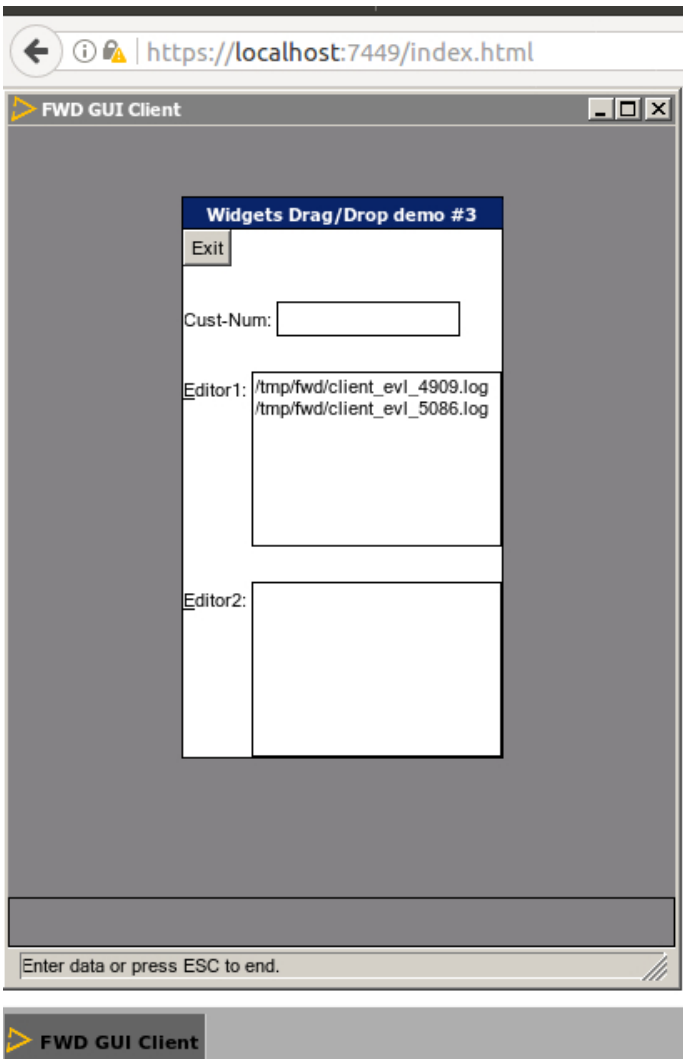
OK. Agreed.

The task branch 1834b has been updated for review to revision 11173.

This update introduces file uploading approach implementation. The implementation is not complete but the working can be tested.

1. The java temporary directory will be java system wide name + "fwd" to separate FWD related files from all others.
2. The JS reads the local file and send it as byte array. This is the point to optimize to improve uploading speed.
3. Java makes arraycopy() method to get the file data. No encoding conversation is performing, the file data is uploading as binary type and this is the as is copy of the source file. May be need to perform some data integrity check.
4. The data synchronization is to be added, as the first TODO. JS reads in async mode, so need to make sure the files are ready to be read on java side before sending DROP-FILE-NOTIFY event to target widget.
5. The uploaded files are deleted on JVM exit.

The new web client picture with temporary directory location:



Continue working. File uploading sync is the next, then handling of the big files and so on...

#196 - 08/30/2017 10:26 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11174.

The update implements file synchronization approach for web client. My investigation inclined me to choose Java side as sync point. The file can be considered as ready to use when it received by Java side and written to the temporary dir. JS has no knowledge about level of the ready status of the uploaded file. So Java side is responsible for correct usage of the file. In web client the emitting of the DROP-FILE-NOTIFY event postponed until the moment when all files in dropped packet has been written to disk.

Also two size limits are now used to estimate possibility to drop the file, 20MB for single file, 100MB for single drop packet. File not matching the criteria even will not be uploaded to the Java side. Probably we need a way to override this defaults within directory.

The question about large file. Do we really need slicing for big files? What advantage will we get taking into account the current uploading is working in parallel(async way) to the main JS thread leaving browser content able to use UI. What file size is a criteria to start splitting it into slices?

Continue working. The next step is to implement large file slicing.

#197 - 08/31/2017 10:28 AM - Greg Shah

Probably we need a way to override this defaults within directory.

Yes.

The question about large file. Do we really need slicing for big files? What advantage will we get taking into account the current uploading is working in parallel(async way) to the main JS thread leaving browser content able to use UI.

I think the concern was that the uploads might file when they are above a certain size.

What file size is a criteria to start splitting it into slices?

I imagine it may vary by browser.

Sergey: Do you have any findings on this?

#198 - 08/31/2017 10:30 AM - Greg Shah

It may be best for Sergey to take the JS file uploading part. What do you think?

#199 - 08/31/2017 10:51 AM - Eugenie Lyzenko

Greg Shah wrote:

It may be best for Sergey to take the JS file uploading part. What do you think?

I have no objection. I just want to finish the current version. I need to implement file size limits directory based configuration support. And testing some edge conditions and consider handling the cases when file uploading failures.

Also I think for large file upload we need something like uploading progress indicator(may be with ability to cancel uploading). Just for user to not to loose control on the process.

#200 - 08/31/2017 11:41 AM - Sergey Ivanovskiy

Greg Shah wrote:

What file size is a criteria to start splitting it into slices?

I imagine it may vary by browser.

Sergey: Do you have any findings on this?

We have `MaxBinaryMessageSize` that means the maximal binary message that can be send from JS to the java client via wss. If we have to send a large message that must be split into packets of messages, then the count of this messages multiplied by the JS time to send one message can have an impact on the user activity, but I didn't evaluate this effect through test cases. I can help to implement JS part only (WebWorker to slice a large file into packets). May be it makes sense to split it into `MaxBinaryMessageSize`.

#201 - 08/31/2017 11:56 AM - Sergey Ivanovskiy

Another argument can be that the available memory that can be used to allocate the binary array on the client to hold the file data. Don't know what are the OS or Browsers parameters that can control this part. It needs to test. Try to send a large file and compare the result if we send this file by its parts. We can make an experiment how large array can be allocated on the JS side.

#202 - 08/31/2017 12:21 PM - Sergey Ivanovskiy

I just found that https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/length the array length can be 2^{32} .

#203 - 08/31/2017 01:12 PM - Sergey Ivanovskiy

Eugenie, Greg, should I take this JS part to upload files that exceed 1Gb?

#204 - 08/31/2017 01:34 PM - Greg Shah

Sergey Ivanovskiy wrote:

Eugenie, Greg, should I take this JS part to upload files that exceed 1Gb?

Yes. If there is anything else remaining on the JS side of the upload, you can take that too.

#205 - 08/31/2017 01:41 PM - Sergey Ivanovskiy

It seems that the uploading files without slicing are done except these configurable parameters in the directory, correct?

```
+  /** The value of the single file size limit available for uploading (20MB). */  
+  var singleFileLimit = 20 * 1024 * 1024;  
+  
+  /** The value of the total files size limit available for uploading in single drop (100MB). */  
+  var totalFilesLimit = 100 * 1024 * 1024;
```

#206 - 08/31/2017 01:48 PM - Sergey Ivanovskiy

Eugenie, please help what 4GL examples should I use for tests with large files? There are uast/drag_n_drop?

#207 - 08/31/2017 01:56 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

It seems that the uploading files without slicing are done except these configurable parameters in the directory, correct?
[...]

Yes, exactly. I have to add reading directory file to take size limit overrides and propagate the values to JS side.

#208 - 08/31/2017 02:00 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie, please help what 4GL examples should I use for tests with large files? There are uast/drag_n_drop?

Yes, the test uast/drag_n_drop/dnd_test3.p can be used to verify dropping feature. Drop any file from OS into Editor1 and you should see the uploaded file names in java temporary directory.

#209 - 08/31/2017 02:06 PM - Sergey Ivanovskiy

Ok, thank you. It seems that we need to do some changes in WebClientProtocol.AppendMessageTask because the file content is gathered here and finally copied into the byte array. It should not work for large files.

#210 - 08/31/2017 02:23 PM - Sergey Ivanovskiy

I think we should use streams here.

#211 - 08/31/2017 10:17 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11175.

This update adds ability to configure file limits for uploads via directory. The location to change is container:

```
<node class="container" name="webClient">
...
  <node class="string" name="uploadSingleFile">
    <node-attribute name="value" value="2M"/>
  </node>
  <node class="string" name="uploadTotalFiles">
    <node-attribute name="value" value="10M"/>
  </node>
...
</node>
```

The size is string value can be suffixed with base units "M" meaning megabytes, "K" kilobytes or nothing meaning bytes.

Continue working. Need to check/cleanup code.

#212 - 09/01/2017 05:00 AM - Sergey Ivanovskiy

Eugenie, you added MSG_SET_UPLOAD_FILE_SIZE_LIMITS message type. It is useful if this limit is changed during the application run time,

correct? I am testing the changes in WebClientProtocol.AppendMessageTask now in order to handle large files and found that my java client code throws these exceptions

```
File /tmp/fwd/safari_wss_error.rtf uploading requested, readyCounter: 1
File safari_wss_error.rtf saving completed, readyCounter: 0
Posting SE_DROP_FILE_NOTIFY event
File /tmp/fwd/Practical_Thin_Server_Architecture_with_Dojo.pdf uploading requested, readyCounter: 1
File Practical_Thin_Server_Architecture_with_Dojo.pdf saving completed, readyCounter: 0
Posting SE_DROP_FILE_NOTIFY event
File /tmp/fwd/tech_scanner-resolution.pdf uploading requested, readyCounter: 1
File tech_scanner-resolution.pdf saving completed, readyCounter: 0
Posting SE_DROP_FILE_NOTIFY event
File /tmp/fwd/servlet-2_4-fr-spec.pdf uploading requested, readyCounter: 1
File servlet-2_4-fr-spec.pdf saving completed, readyCounter: 0
Posting SE_DROP_FILE_NOTIFY event
File /tmp/fwd/Хатха-Йога для начинающих.djvu uploading requested, readyCounter: 1
Sep 01, 2017 11:17:03 AM org.eclipse.jetty.websocket.common.Parser notifyWebSocketException
WARNING:
org.eclipse.jetty.websocket.api.MessageTooLargeException: Binary message size [32910] exceeds maximum size [32894]
    at org.eclipse.jetty.websocket.api.WebSocketPolicy.assertValidBinaryMessageSize(WebSocketPolicy.java:128)
    at org.eclipse.jetty.websocket.common.Parser.assertSanePayloadLength(Parser.java:130)
    at org.eclipse.jetty.websocket.common.Parser.parseFrame(Parser.java:480)
    at org.eclipse.jetty.websocket.common.Parser.parse(Parser.java:252)
    at org.eclipse.jetty.websocket.common.io.AbstractWebSocketConnection.readParse(AbstractWebSocketConnection.java:675)
    at org.eclipse.jetty.websocket.common.io.AbstractWebSocketConnection.onFillable(AbstractWebSocketConnection.java:505)
    at org.eclipse.jetty.io.AbstractConnection$ReadCallback.succeeded(AbstractConnection.java:273)
    at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:95)
    at org.eclipse.jetty.io.ssl.SslConnection.onFillable(SslConnection.java:186)
    at org.eclipse.jetty.io.AbstractConnection$ReadCallback.succeeded(AbstractConnection.java:273)
    at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:95)
    at org.eclipse.jetty.io.SelectChannelEndPoint$2.run(SelectChannelEndPoint.java:93)
    at org.eclipse.jetty.util.thread.strategy.ExecuteProduceConsume.produceAndRun(ExecuteProduceConsume.java:246)
    at org.eclipse.jetty.util.thread.strategy.ExecuteProduceConsume.run(ExecuteProduceConsume.java:156)
    at org.eclipse.jetty.util.thread.QueuedThreadPool.runJob(QueuedThreadPool.java:654)
    at org.eclipse.jetty.util.thread.QueuedThreadPool$3.run(QueuedThreadPool.java:572)
    at java.lang.Thread.run(Thread.java:748)

org.eclipse.jetty.websocket.api.MessageTooLargeException: Binary message size [32910] exceeds maximum size [32894]
    at org.eclipse.jetty.websocket.api.WebSocketPolicy.assertValidBinaryMessageSize(WebSocketPolicy.java:128)
    at org.eclipse.jetty.websocket.common.Parser.assertSanePayloadLength(Parser.java:130)
    at org.eclipse.jetty.websocket.common.Parser.parseFrame(Parser.java:480)
    at org.eclipse.jetty.websocket.common.Parser.parse(Parser.java:252)
    at org.eclipse.jetty.websocket.common.io.AbstractWebSocketConnection.readParse(AbstractWebSocketConnection.java:675)
    at org.eclipse.jetty.websocket.common.io.AbstractWebSocketConnection.onFillable(AbstractWebSocketConnection.java:505)
    at org.eclipse.jetty.io.AbstractConnection$ReadCallback.succeeded(AbstractConnection.java:273)
    at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:95)
    at org.eclipse.jetty.io.ssl.SslConnection.onFillable(SslConnection.java:186)
    at org.eclipse.jetty.io.AbstractConnection$ReadCallback.succeeded(AbstractConnection.java:273)
    at org.eclipse.jetty.io.FillInterest.fillable(FillInterest.java:95)
    at org.eclipse.jetty.io.SelectChannelEndPoint$2.run(SelectChannelEndPoint.java:93)
    at org.eclipse.jetty.util.thread.strategy.ExecuteProduceConsume.produceAndRun(ExecuteProduceConsume.java:246)
    at org.eclipse.jetty.util.thread.strategy.ExecuteProduceConsume.run(ExecuteProduceConsume.java:156)
    at org.eclipse.jetty.util.thread.QueuedThreadPool.runJob(QueuedThreadPool.java:654)
    at org.eclipse.jetty.util.thread.QueuedThreadPool$3.run(QueuedThreadPool.java:572)
    at java.lang.Thread.run(Thread.java:748)
```

It means that something goes wrong because the binary data should not exceed the size 32894 given by these settings

```
<node class="integer" name="maxBinaryMessage">
  <node-attribute name="value" value="32894"/>
</node>
```

Did you encounter such kind of exceptions in 1834b? I am investigating the reasons now.

#213 - 09/01/2017 06:51 AM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie, you added MSG_SET_UPLOAD_FILE_SIZE_LIMITS message type. It is useful if this limit is changed during the application run time, correct?

As noted above it is designed to be used for initial set up the JS respective options with override values from directory when the client is starting.

But of course you can use it at run time to change the values, every time the JS is ready to accept the messages.

I am testing the changes in WebClientProtocol.AppendMessageTask now in order to handle large files and found that my java client code throws these exceptions

[...]

It means that something goes wrong because the binary data should not exceed the size 32894 given by these settings

[...]

Did you encounter such kind of exceptions in 1834b? I am investigating the reasons now.

I have not tried yet. Change the single file limit to small value(say 10K) and the uploading should be refused.

#214 - 09/01/2017 07:31 AM - Sergey Ivanovskiy

- File Хатха-Йога для начинающих.djvu added

Ok, don't understand this extra 16 bytes that was added to the wss message? Please update src/com/goldencode/p2j/ui/client/driver/web/res/p2j.socket.js, committed rev. 11176.

Eugenie, I attached this file that you can check this exception related to Editor. Let the job be more fun.

```
java.lang.reflect.InvocationTargetException
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:498)
  at com.goldencode.p2j.util.MethodInvoker.invoke(MethodInvoker.java:124)
  at com.goldencode.p2j.net.Dispatcher.processInbound(Dispatcher.java:757)
  at com.goldencode.p2j.net.Conversation.block(Conversation.java:412)
  at com.goldencode.p2j.net.Conversation.waitMessage(Conversation.java:348)
  at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1170)
  at com.goldencode.p2j.net.Queue.transact(Queue.java:641)
  at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
  at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
  at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
  at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
  at com.sun.proxy.$Proxy8.waitFor(Unknown Source)
```

```
at com.goldencode.p2j.ui.LogicalTerminal.waitFor (LogicalTerminal.java:6346)
at com.goldencode.p2j.ui.LogicalTerminal.waitFor (LogicalTerminal.java:6116)
at com.goldencode.testcases.drag_n_drop.DndTest3.lambda$execute$0 (DndTest3.java:52)
at com.goldencode.p2j.util.Block.body (Block.java:604)
at com.goldencode.p2j.util.BlockManager.processBody (BlockManager.java:6985)
at com.goldencode.p2j.util.BlockManager.topLevelBlock (BlockManager.java:6776)
at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:343)
at com.goldencode.p2j.util.BlockManager.externalProcedure (BlockManager.java:317)
at com.goldencode.testcases.drag_n_drop.DndTest3.execute (DndTest3.java:32)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:498)
at com.goldencode.p2j.util.Utills.invoke (Utills.java:1337)
at com.goldencode.p2j.main.StandardServer$MainInvoker.execute (StandardServer.java:1899)
at com.goldencode.p2j.main.StandardServer.invoke (StandardServer.java:1394)
at com.goldencode.p2j.main.StandardServer.standardEntry (StandardServer.java:513)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:498)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:124)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:757)
at com.goldencode.p2j.net.Conversation.block (Conversation.java:412)
at com.goldencode.p2j.net.Conversation.run (Conversation.java:232)
at java.lang.Thread.run (Thread.java:748)
Caused by: java.lang.ArrayIndexOutOfBoundsException: 1061
at com.goldencode.p2j.ui.client.gui.EditorGuiImpl.goToNewLine (EditorGuiImpl.java:1961)
at com.goldencode.p2j.ui.client.Editor.splitLine (Editor.java:2216)
at com.goldencode.p2j.ui.client.gui.EditorGuiImpl.splitLine (EditorGuiImpl.java:2150)
at com.goldencode.p2j.ui.client.Editor.placeChar (Editor.java:2188)
at com.goldencode.p2j.ui.client.Editor.lambda$insertString$0 (Editor.java:1339)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket (ThinClient.java:15096)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket (ThinClient.java:15040)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket (ThinClient.java:14943)
at com.goldencode.p2j.ui.client.Editor.insertString (Editor.java:1335)
at com.goldencode.p2j.ui.chui.ThinClient.insertEditorString (ThinClient.java:4526)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:498)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:124)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:757)
at com.goldencode.p2j.net.Conversation.block (Conversation.java:412)
at com.goldencode.p2j.net.Conversation.waitForMessage (Conversation.java:348)
at com.goldencode.p2j.net.Queue.transactImpl (Queue.java:1170)
at com.goldencode.p2j.net.Queue.transact (Queue.java:641)
at com.goldencode.p2j.net.BaseSession.transact (BaseSession.java:271)
at com.goldencode.p2j.net.HighLevelObject.transact (HighLevelObject.java:211)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore (RemoteObject.java:1473)
at com.goldencode.p2j.net.InvocationStub.invoke (InvocationStub.java:145)
at com.sun.proxy.$Proxy10.trigger (Unknown Source)
at com.goldencode.p2j.ui.chui.ThinClient.trigger (ThinClient.java:12558)
at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers (ThinClient.java:18014)
at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers (ThinClient.java:17818)
at com.goldencode.p2j.ui.chui.ThinClient.invokeTriggers (ThinClient.java:17704)
at com.goldencode.p2j.ui.chui.ThinClient.processProgressEvent (ThinClient.java:16894)
at com.goldencode.p2j.ui.chui.ThinClient.processEventsWorker (ThinClient.java:16125)
at com.goldencode.p2j.ui.chui.ThinClient.pop (ThinClient.java:15133)
at com.goldencode.p2j.ui.chui.ThinClient.eventBracket (ThinClient.java:15116)
at com.goldencode.p2j.ui.chui.ThinClient.eventDrawingBracket (ThinClient.java:15040)
at com.goldencode.p2j.ui.chui.ThinClient.applyWorker (ThinClient.java:14801)
at com.goldencode.p2j.ui.chui.ThinClient.waitForEvent (ThinClient.java:14070)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker (ThinClient.java:11822)
at com.goldencode.p2j.ui.chui.ThinClient.waitForWorker (ThinClient.java:11401)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor (ThinClient.java:11349)
at com.goldencode.p2j.ui.chui.ThinClient.waitFor (ThinClient.java:11303)
at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke (Method.java:498)
at com.goldencode.p2j.util.MethodInvoker.invoke (MethodInvoker.java:124)
at com.goldencode.p2j.net.Dispatcher.processInbound (Dispatcher.java:757)
at com.goldencode.p2j.net.Conversation.block (Conversation.java:412)
at com.goldencode.p2j.net.Conversation.waitForMessage (Conversation.java:348)
```

```
at com.goldencode.p2j.net.Queue.transactImpl(Queue.java:1170)
at com.goldencode.p2j.net.Queue.transact(Queue.java:641)
at com.goldencode.p2j.net.BaseSession.transact(BaseSession.java:271)
at com.goldencode.p2j.net.HighLevelObject.transact(HighLevelObject.java:211)
at com.goldencode.p2j.net.RemoteObject$RemoteAccess.invokeCore(RemoteObject.java:1473)
at com.goldencode.p2j.net.InvocationStub.invoke(InvocationStub.java:145)
at com.sun.proxy.$Proxy12.standardEntry(Unknown Source)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:361)
at com.goldencode.p2j.main.ClientCore.start(ClientCore.java:158)
at com.goldencode.p2j.main.ClientDriver.start(ClientDriver.java:250)
at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
at com.goldencode.p2j.main.ClientDriver.process(ClientDriver.java:144)
at com.goldencode.p2j.main.ClientDriver.main(ClientDriver.java:313)
[09/01/2017 14:24:06 MSK] (StandardServer.invoke:SEVERE) {00000003:00000012:bogus} Abnormal end!
```

#215 - 09/01/2017 08:04 AM - Sergey Ivanovskiy

May be this extra 16 bytes are related to the protocol format that must carry the payload data. <https://tools.ietf.org/html/rfc6455> 5.2. Base Framing Protocol, but in that case i don't understand why the code works fine in another cases and it seems it was tested to send and receive files.

#216 - 09/01/2017 08:19 AM - Sergey Ivanovskiy

Eugenie, please review the changes in 11178, 11177. We need to implement uploading files from file channels, but now we use arrays as data sources.

```
else if (message[offset] == MSG_FILE_UPLOADING)
{
    offset = offset + 1;

    // file name length
    int nameLength = readMessageInt16(message, offset);
    offset = offset + 2;

    // file name
    String fileName = readMessageText(message, offset, offset + 2 * nameLength);
    offset = offset + 2 * nameLength;

    // file size
    int fileSize = readMessageInt32(message, offset);
    offset = offset + 4;

    // file content itself
    byte[] fileData = readByteArray(message, offset, fileSize);

    // content can be empty
    if (fileData != null)
    {
        // ask java to store the file locally in temp directory
        this.callbacks.saveDropTargetFile(fileName, fileData, fileSize);
    }

    handled = true;
}
```

#217 - 09/01/2017 09:10 AM - Sergey Ivanovskiy

I will take this part to implement MSG_FILE_UPLOADING from file channels on the java client side too if there are no objections and js part related to file slicing.

#218 - 09/01/2017 10:48 AM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11179.

This is minor code cleanup(DragDropHelper) and changing support status in gap rules for attributes/methods to partial instead of stub.

The debugging log is still there for DragDropHelper. We can remove them when DnD support implementation completes but now it is pretty useful.

Also the user_interface.rules rule file had double adding entries for frame option map, removed.

#219 - 09/01/2017 11:45 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

The task branch 1834b has been updated for review to revision 11179.

This is minor code cleanup(DragDropHelper) and changing support status in gap rules for attributes/methods to partial instead of stub.

The debugging log is still there for DragDropHelper. We can remove them when DnD support implementation completes but now it is pretty useful.

Also the user_interface.rules rule file had double adding entries for frame option map, removed.

What is about the issue in the note 214?

#220 - 09/01/2017 11:59 AM - Sergey Ivanovskiy

Another question related to this implementation is that we have a message type MSG_FILE_UPLOADING with its 32-bit field file size. Why this field is 32bit length? Usually the file size is hold by 64bit number, correct? Blob.size is 64 bit number.

<https://developer.mozilla.org/en-US/docs/Web/API/Blob/size>

#221 - 09/01/2017 12:11 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie Lyzenko wrote:

The task branch 1834b has been updated for review to revision 11179.

This is minor code cleanup(DragDropHelper) and changing support status in gap rules for attributes/methods to partial instead of stub.

The debugging log is still there for DragDropHelper. We can remove them when DnD support implementation completes but now it is pretty

useful.

Also the user_interface.rules rule file had double adding entries for frame option map, removed.

What is about the issue in the note 214?

I think it is not related to DnD itself. Instead it relates to the code in GUI Editor string processing. Probably the issue is a combination of the Cyrillic chars with space char or other, or the filename was improperly transferred via web socket. You can debug and see what is actually happening in EditorGuiImpl. I can say from my personal experience having the file name with Cyrillic chars and spaces is a bad practice. It will cause more or less severe issues, this is just a question of time.

#222 - 09/01/2017 12:16 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Another question related to this implementation is that we have a message type MSG_FILE_UPLOADING with its 32-bit field file size. Why this field is 32bit length?

I thought it is high enough for files to have 4G limit(32-bit int). I can not imagine the case when one will drop the bigger file via this web client drop operation. This is a kind of crazy I think. But if you think we need 64-bit number - feel free to change.

#223 - 09/01/2017 12:28 PM - Sergey Ivanovskiy

Ok, understand now. There is the design issue that WebClientProtocol.AppendMessageTask saves the whole message into the temporary file and then its content is handled. In the case of MSG_FILE_UPLOADING the content is copied to the new named file in the temporary directory under the new fwd folder. It seems that in this case the file can be gathered on the fly. I don't think about this new way.

#224 - 09/01/2017 01:18 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

What is about the issue in the note 214?

The investigation:

1. The font width array we taking from JS has 0-255 elements range only:
2. The code(EditorGuiImpl):

```
...
    for (int i = 0; i < subtext.length(); i++)
    {
        mw += widths[0][subtext.charAt(i)]; // Line 1961 where exception happens
    }
...

```

3. When the subtext become having Cyrillic char the subtext.charAt(i) value becomes over 255, 1061 for the first letter of your file(X).
4. We have out of the array range (1061 > 255)
5. Replace all Cyrillic letters in filename with US Latin1 range counterpart and issue must go away.
6. I guess this issue is not one of our priority and adding Cyrillic support can be postponed.

#225 - 09/01/2017 01:30 PM - Greg Shah

I guess this issue is not one of our priority and adding Cyrillic support can be postponed.

Actually, I think this is a serious issue we need to resolve now. It will affect any unicode characters that are outside of the extended ASCII range. We definitely have customers where this is an issue. Please rework the array to process unicode characters.

#226 - 09/01/2017 05:20 PM - Eugenie Lyzenko

- File *dnd_test3_fwd_web_unicode_20170901a.jpg* added

Prepared the fix for Unicode chars.

The idea. Instead of looping through every char in substring we can compute full substring by `getTextWidth()`. I think this is fine because:

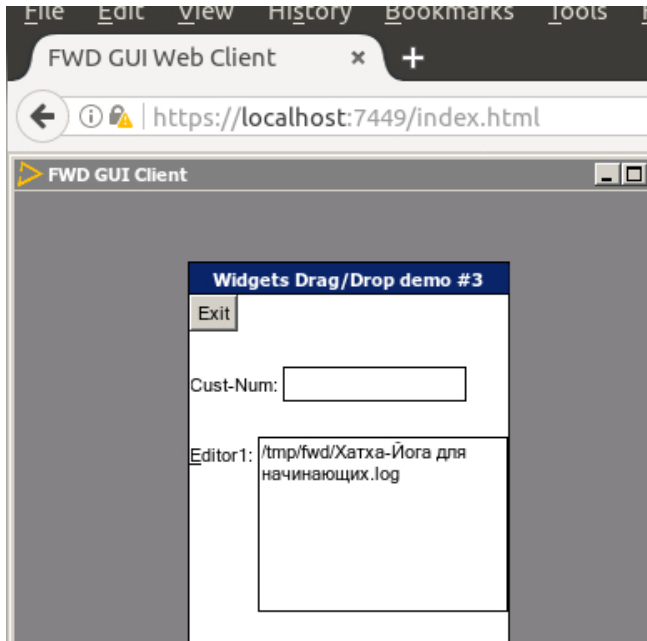
1. Java does not provide method to get width of the chars in array outside 255 element in `java.awt.FontMetrics`.
2. Moreover getting full Unicode range(65536) chars can be too expensive even for high performance system when calling very often.

The change in `EditorGuilImpl`:

```
...
    int mw = gf.font().font.legacyMaxWidth * 2;
    int editWidth = editor.physicalDimension().width;
    final int[] subtextWidth = new int[1];
    // getting full substring width instead of looping through every char
    ThinClient.getInstance().independentEventDrawingBracket(this,
        () -> subtextWidth[0] = gd.getTextWidth(subtext, gf.font().font) );
    mw += subtextWidth[0];
...

```

The result(I have renamed one of my local file instead of using original djvu):



Swing client is also OK. So if no objection I'll commit the fix into 1834b branch.

#227 - 09/01/2017 06:18 PM - Sergey Ivanovskiy

Committed revision 11180 changed(added new methods) DragDropHelper, ClientProtocolHooks to use file channels for file uploading. Planning to complete JS part with file slicing at this Sunday.

#228 - 09/01/2017 08:09 PM - Eugenie Lyzenko

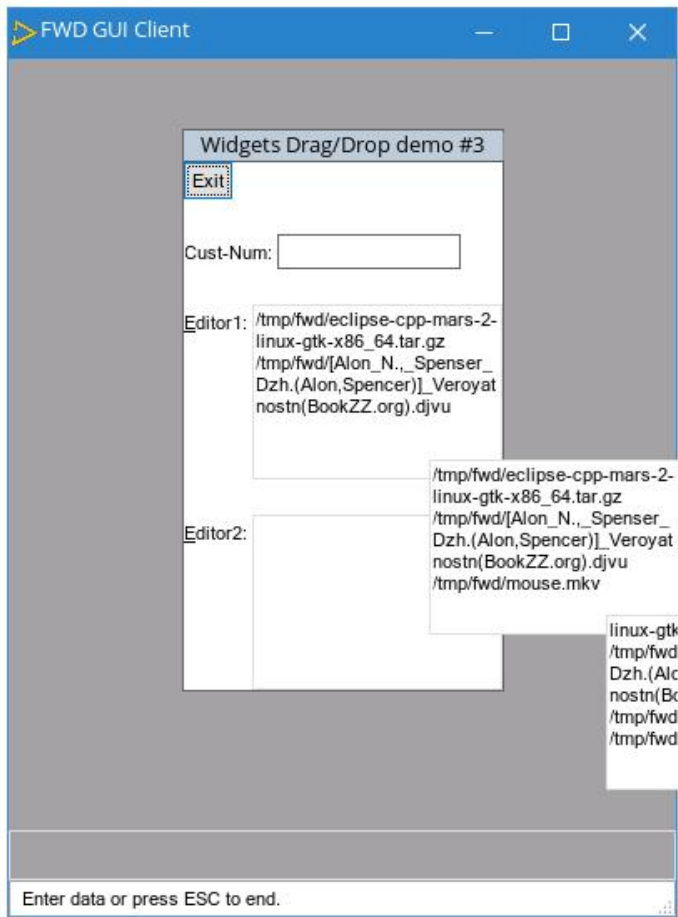
The task branch 1834b has been updated for review to revision 11181.

Fix for array invalid index issue in EditorGuiImpl. As note above the idea is to replace loop with full string width calculation.

#229 - 09/03/2017 10:34 AM - Sergey Ivanovskiy

- File *MsgsInterleaving.png* added

The canvas image can be broken if its window is moved during the file uploading.



#230 - 09/03/2017 11:25 AM - Sergey Ivanovskiy

Please review committed revision 11182 that changed p2j.socket.js to send a packet of partial messages without recursive calls, added file slicing,

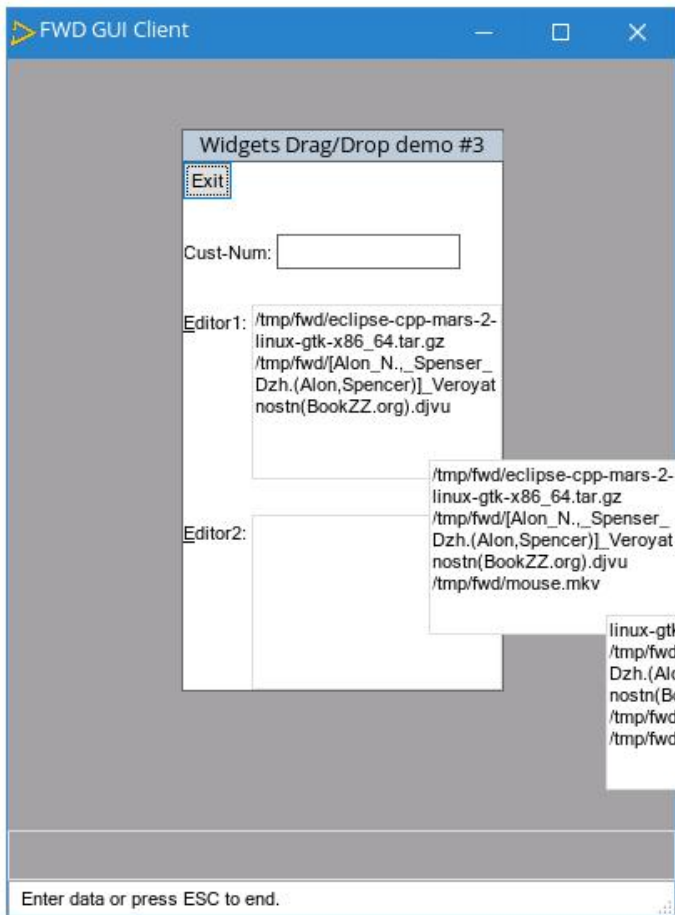
WebWorker API is not used. To reproduce the broken image I used these settings

```
<node class="string" name="uploadSingleFile">  
  <node-attribute name="value" value="200M"/>  
</node>  
<node class="string" name="uploadTotalFiles">  
  <node-attribute name="value" value="1000M"/>  
</node>
```

#231 - 09/03/2017 12:14 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

The canvas image can be broken if its window is moved during the file uploading.



Yes, this is known issue see note [#107](#).

I think not related to drag-and-drop itself.

#232 - 09/03/2017 03:12 PM - Sergey Ivanovskiy

Eugenie, I don't understand, please explain why sendDropFileNotify should arrive before uploadFileToJava message on drag event function mouseDropHandler(evt)

```
// detect the drop target candidate
if (targetWindow !== undefined)
{
    targetWidget = targetWindow.findMouseSource(evt,
                                                targetWindow.mouseAwareWidgets["mouseenter"]);
    // after all files uploaded finalize drop target by noting it can use files
    p2j.socket.sendDropFileNotify(targetWindow.id, targetWidget, fileNames);
}

// this ensures the drop file notify message arriving before file uploading starts
for (var i = 0, f; f = filesToUpload[i]; i++)
{
    // sending file content with names associated
    p2j.socket.uploadFileToJava(f);
}
```

#233 - 09/03/2017 06:40 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie, I don't understand, please explain why sendDropFileNotify should arrive before uploadFileToJava message on drag event function mouseDropHandler(evt)
[...]

This is how the current file ready approach works.

1. Sending DROP-FILE-NOTIFY message from JS to Java initializes file uploading counter on the Java side. Meaning if we have N file to upload the counter become N. Every additional file is waiting to be uploaded increase this counter. This does not initiate sending DROP-FILE-NOTIFY from Java side to target widget.
2. When file has been uploaded and ready to be read by the Java side the counter is decreasing by 1.
3. So when all requested files have been uploaded and ready the counter become 0. And this is a signal the Java can send actual DROP-FILE-NOTIFY to the target widget within Java client.
4. Placing p2j.socket.sendDropFileNotify before file uploading ensures the file counter has correct values before any upload starts. Otherwise due to asynchronous upload it is possible to have 0 counter value(or even negative) when not all files uploaded and ready.

This is the way I have implemented mutex semaphore in this case.

#234 - 09/04/2017 09:05 PM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Please review committed revision 11182 that changed p2j.socket.js to send a packet of partial messages without recursive calls, added file slicing, WebWorker API is not used. To reproduce the broken image I used these settings
[...]

I'm OK with changes in 1834b.

#235 - 09/05/2017 07:52 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

The idea. Instead of looping through every char in substring we can compute full substring by `getTextWidth()`. I think this is fine because:
1. Java does not provide method to get width of the chars in array outside 255 element in `java.awt.FontMetrics`.
2. Moreover getting full Unicode range(65536) chars can be too expensive even for high performance system when calling very often.

The change in `EditorGuiImpl`:
[...]

I'm OK with measuring the entire text instead of measuring letter by letter. Some questions/notes:

1. I think you can use `getTextWidthNative` call with the same effect
2. I've added this comment when I changed to measure the text char-by-char:

```
// TODO: in word-wrap mode, this optimization may lead to false-negatives, as the  
// formula above does not adjust for the kerning space between the text's letters...
```

Can you check typing unicode chars (i.e. russian letters) in word-wrap mode?

3. there are also these APIs which return individual char width, in ASCII range: `FontMetricsHelper.charWidth` and `FontMetricsHelper.getWidths` - we need to review the usage of these APIs, as I think we might have problems when Unicode is involved.

#236 - 09/05/2017 08:43 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

Eugenie Lyzenko wrote:

- The idea. Instead of looping through every char in substring we can compute full substring by `getTextWidth()`. I think this is fine because:
1. Java does not provide method to get width of the chars in array outside 255 element in `java.awt.FontMetrics`.
 2. Moreover getting full Unicode range(65536) chars can be too expensive even for high performance system when calling very often.

The change in `EditorGuiImpl`:
[...]

I'm OK with measuring the entire text instead of measuring letter by letter. Some questions/notes:

1. I think you can use `getTextWidthNative` call with the same effect

Yes, finally it will also call `gd.getTextWidth(String, FontDetails)`

2. I've added this comment when I changed to measure the text char-by-char:

[...]

Can you check typing unicode chars (i.e. russian letters) in word-wrap mode?

Unfortunately cyrillic letters are not displaying when typing inside `EditorGuiImpl`, and the caret is not moving. Some constraints to accept typing outside 255 char number?

#237 - 09/05/2017 08:50 AM - Eugenie Lyzenko

Constantin Asofiei wrote:

Eugenie Lyzenko wrote:

- The idea. Instead of looping through every char in substring we can compute full substring by `getTextWidth()`. I think this is fine because:
1. Java does not provide method to get width of the chars in array outside 255 element in `java.awt.FontMetrics`.
 2. Moreover getting full Unicode range(65536) chars can be too expensive even for high performance system when calling very often.

3. there are also these APIs which return individual char width, in ASCII range: `FontMetricsHelper.charWidth` and `FontMetricsHelper.getWidths` - we need to review the usage of these APIS, as I think we might have problems when Unicode is involved.

Yes, using these function we need to take into account the only range is valid fo them is 0-255. For Unicode chars we need the alternative.

#238 - 09/05/2017 09:32 AM - Eugenie Lyzenko

Constantin,

Look at Editor, line 954:

```
...
// handle all printable characters and '\n'
if (Keyboard.isPrintableKey(key) ||
    key == Keyboard.KA_RETURN ||
    key == Keyboard.KA_NEW_LINE)
{
...

```

The method `Keyboard.isPrintableKey()` returns true only for chars in range 0x20-0xFF. So Unicode range is not supported to be typed inside Editor, only when changing content by other way than typing, like direct changing the value by file name drop operation.

#239 - 09/05/2017 09:38 AM - Constantin Asofiei

Eugenie Lyzenko wrote:

Constantin,

Look at Editor, line 954:

[...]

The method `Keyboard.isPrintableKey()` returns true only for chars in range 0x20-0xFF. So Unicode range is not supported to be typed inside Editor, only when changing content by other way than typing, like direct changing the value by file name drop operation.

I think the same issue will be with FILL-IN.

Greg: do you want to do this in a different task? Because I would like to see how triggers and maybe other UI stuff works with Unicode.

#240 - 09/05/2017 09:41 AM - Greg Shah

Greg: do you want to do this in a different task? Because I would like to see how triggers and maybe other UI stuff works with Unicode.

Yes. I think it is best to split that work off. Please create a new task for this and make me a watcher.

#241 - 09/05/2017 09:42 AM - Greg Shah

I think Sergey is done with his part of the work (the JS file upload).

Eugenie: is there anything more to do on this task?

#242 - 09/05/2017 10:03 AM - Eugenie Lyzenko

Greg Shah wrote:

I think Sergey is done with his part of the work (the JS file upload).

Eugenie: is there anything more to do on this task?

We need to:

1. Clean up code from debug calls to System.err.println(). This is fast.
2. I think we need to add possibility to cancel direct manipulation drag processing on pressing ESC key. This relates to drag initiated box selection, widget moving, widget/row resizing. I think it is not so time consuming to implement this.

Do we need to point 2 to be added?

#243 - 09/05/2017 10:19 AM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

Greg Shah wrote:

I think Sergey is done with his part of the work (the JS file upload).

Eugenie: is there anything more to do on this task?

2. I think we need to add possibility to cancel direct manipulation drag processing on pressing ESC key. This relates to drag initiated box

selection, widget moving, widget/row resizing. I think it is not so time consuming to implement this.

Greg, I've just re-tested this feature in 4GL. It does not work as I've described.

So the only thing to do is point 1 above(remove debug calls). We can add point 2 as additional option as bonus to our implementation later if we decide it is necessary.

#244 - 09/05/2017 10:30 AM - Greg Shah

Since the ESC behavior doesn't exist in the 4GL, we will avoid it for now.

Please do the cleanup.

#245 - 09/05/2017 11:24 AM - Eugenie Lyzenko

Greg Shah wrote:

Since the ESC behavior doesn't exist in the 4GL, we will avoid it for now.

OK.

Please do the cleanup.

The task branch 1834b has been updated for review to revision 11184.

The debug calls to log the drag-n-drop details have been removed. Also updated the status of the drag-n-drop related attributes and methods in gap analysis rules.

#246 - 09/05/2017 12:25 PM - Greg Shah

Code Review Task Branch 1834b Revision 11184

I've checked in some fixes for typos and coding standards.

1. The `ThinClient.initializePost()` call to `DragDropHelper.init()` should only be done if `(!OutputManager.instance()).isChui()`.
2. In `ClientProtocolHooks`, I don't think all these imports are needed:


```
import java.io.*;
import java.nio.channels.FileChannel;
import java.util.*;
```

Also the FileChannel should be imported using *.

3. Please change EVL to SBI for those files where the changes were written by Sergey. Otherwise we won't know who to ask for questions.

4. The code in DragDropHelper.init() is specific to the GUI web client. Is it feasible to hide this part of the implementation in the GUI web driver instead of having the DragDropHelper implement this?

I don't want to make the implementation more complicated. But if you can come up with a way that hides the Swing parts in the Swing driver and the web parts in the web GUI driver, then the DragDropHelper will become generic.

5. DragDropHelper, GuiWebDriver, GuiWebSocket should import nio using *.

6. In FrameGuiImpl the import java.awt.event.MouseEvent should be marked with a comment as we do elsewhere in common code (non-Swing code).

#247 - 09/05/2017 12:38 PM - Eugenie Lyzenko

Greg Shah wrote:

3. Please change EVL to SBI for those files where the changes were written by Sergey. Otherwise we won't know who to ask for questions.

Sergey, would you please add SBI note for the respective changes in all files you had to modify. No one better knows the list of these files and modifications to describe.

#248 - 09/05/2017 05:46 PM - Eugenie Lyzenko

Greg Shah wrote:

Code Review Task Branch 1834b Revision 11184

I've checked in some fixes for typos and coding standards.

1. The ThinClient.initializePost() call to DragDropHelper.init() should only be done if (!OutputManager.instance().isChui()).

2. In ClientProtocolHooks, I don't think all these imports are needed:

[...]

Yes, all of them are needed.

Also the FileChannel should be imported using *.

3. Please change EVL to SBI for those files where the changes were written by Sergey. Otherwise we won't know who to ask for questions.

4. The code in DragDropHelper.init() is specific to the GUI web client. Is it feasible to hide this part of the implementation in the GUI web driver instead of having the DragDropHelper implement this?

I don't want to make the implementation more complicated. But if you can come up with a way that hides the Swing parts in the Swing driver and the web parts in the web GUI driver, then the DragDropHelper will become generic.

5. DragDropHelper, GuiWebDriver, GuiWebSocket should import nio using *.

6. In FrameGuiImpl the import java.awt.event.MouseEvent should be marked with a comment as we do elsewhere in common code (non-Swing code).

The task branch 1834b has been updated for review to revision 11185.

This is the notes resolution except point 4 with partial resolution. The swing mode just do nothing in DragDropHelper.init(). More deep reworking will require the DragDropHelper to be split into web and swing parts. Also the web client related file uploading code must be isolated in web version too for DragDropHelper to be generic.

Do you want this work to be done in current 1834b branch?

#249 - 09/05/2017 10:39 PM - Eugenie Lyzenko

The task branch 1834b has been updated for review to revision 11186.

This is the note 4 resolution. All internals related to set up temporary directory location/limits are now moved to the GUI web client driver making DragDropHelper class to be generic one.

#250 - 09/06/2017 08:07 AM - Greg Shah

Code Review Task Branch 1834b Revision 11186

I'm fine with the changes.

We are trying to get 3330a and 3222a into the trunk before 1834b. Unless there is something else to do, pause work on this for now. When the other two branches are merged to trunk, you can rebase and start testing.

#251 - 09/06/2017 09:27 AM - Eugenie Lyzenko

Greg Shah wrote:

Code Review Task Branch 1834b Revision 11186

I'm fine with the changes.

We are trying to get 3330a and 3222a into the trunk before 1834b. Unless there is something else to do, pause work on this for now. When the other two branches are merged to trunk, you can rebase and start testing.

OK.

#252 - 09/21/2017 05:51 PM - Eugenie Lyzenko

As preparation step the task branch 1834b has been rebased with trunk 11164. New revision is 11188(I saw the branch was updated to 11187 since my last changes).

#253 - 09/22/2017 03:29 PM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11165. New revision is 11189.

#254 - 09/25/2017 05:29 PM - Eugenie Lyzenko

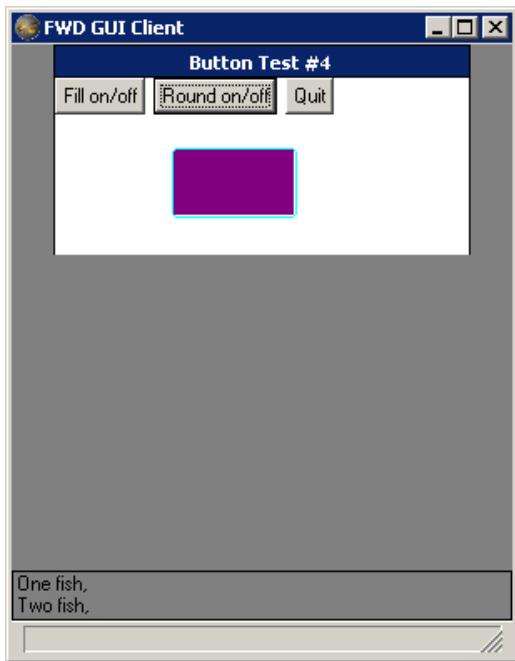
The task branch 1834b has been rebased with trunk 11166. New revision is 11190.

I'm going to start regression testing to prepare merging 1834b into trunk. I think runtime ChUI test will be enough. In addition to standalone tests.

#255 - 09/25/2017 10:05 PM - Eugenie Lyzenko

- File *gui_button_test4_issue_fwd_20170925.jpg* added

GUI standalone tests has no regressions. But I see the issue presented in trunk as well. The rounded rectangle has incorrect filling/sizing. See the picture attached(./button/gui_btn_test4.p):



#256 - 09/26/2017 01:15 AM - Sergey Ivanovskiy

Eugenie Lyzenko wrote:

Greg Shah wrote:

3. Please change EVL to SBI for those files where the changes were written by Sergey. Otherwise we won't know who to ask for questions.

Sergey, would you please add SBI note for the respective changes in all files you had to modify. No one better knows the list of these files and modifications to describe.

Eugenie, committed revision 11191, 11192 added comments for these files WebClientProtocol.java, p2j.socket.js, DragDropHelper.java, GuiWebDriver.java, GuiWebSocket.java and ClientProtocolHooks.java.

#257 - 09/26/2017 05:23 AM - Eugenie Lyzenko

Sergey Ivanovskiy wrote:

Eugenie, committed revision 11191, 11192 added comments for these files WebClientProtocol.java, p2j.socket.js, DragDropHelper.java, GuiWebDriver.java, GuiWebSocket.java and ClientProtocolHooks.java.

Is 11192 also related to comment changes? Correct? No real code modification?

I'm doing rebase and restart regression testing.

The task branch 1834b has been rebased with trunk 11167. New revision is 11193.

#258 - 09/26/2017 05:59 AM - Ovidiu Maxiniuc

Eugenie Lyzenko wrote:

GUI standalone tests has no regressions. But I see the issue presented in trunk as well. The rounded rectangle has incorrect filling/sizing. See the picture attached(./button/gui_btn_test4.p):

Although the roundness was an issue (and will be until we add a primitive that draws the complete round-rect of variable thickness for web client), I recall the filling to be fine. I tested then with ./uast/rectangle/rectangle-tests.p. Please check the following changes in ClassicTheme.java:

- replace line 4610 with `int ROUNDNESS = cfg.groupBox ? 5 : 11;`
- replace line 4618 with `gd.fillRoundRect(x + lt - 1, y + lt - 1, width - 2 * lt + 1, height - 2 * lt + 1, ROUNDNESS);`.

I haven't tested with the web client yet.

#259 - 09/26/2017 06:20 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Eugenie Lyzenko wrote:

GUI standalone tests has no regressions. But I see the issue presented in trunk as well. The rounded rectangle has incorrect filling/sizing. See the picture attached(./button/gui_btn_test4.p):

I haven't tested with the web client yet.

The Web client does not have this issue.

#260 - 09/26/2017 06:50 AM - Eugenie Lyzenko

Ovidiu Maxiniuc wrote:

Eugenie Lyzenko wrote:

GUI standalone tests has no regressions. But I see the issue presented in trunk as well. The rounded rectangle has incorrect filling/sizing. See the picture attached(./button/gui_btn_test4.p):

Although the roundness was an issue (and will be until we add a primitive that draws the complete round-rect of variable thickness for web client), I recall the filling to be fine. I tested then with ./uast/rectangle/rectangle-tests.p. Please check the following changes in ClassicTheme.java:

- replace line 4610 with `int ROUNDNESS = cfg.groupBox ? 5 : 11;`
- replace line 4618 with `gd.fillRoundRect(x + lt - 1, y + lt - 1, width - 2 * lt + 1, height - 2 * lt + 1, ROUNDNESS);`

Yes, confirm this change fixes the issue.

Greg,

Should I include this fix into 1834b branch?

#261 - 09/26/2017 07:22 AM - Eugenie Lyzenko

There are also 23 new javadoc errors introduced in yesterday trunk(11166) from branch 3222a. I can also fix them to include into 1834b. Is it OK?

#262 - 09/26/2017 07:53 AM - Greg Shah

Should I include this fix into 1834b branch?

Yes.

There are also 23 new javadoc errors introduced in yesterday trunk(11166) from branch 3222a. I can also fix them to include into 1834b. Is it OK?

Yes.

I've noticed quite a few javadoc problems with the OracleJDK, even in trunk. If you can resolve them quickly, it would be great.

#263 - 09/26/2017 08:00 AM - Eugenie Lyzenko

Greg Shah wrote:

I've noticed quite a few javadoc problems with the OracleJDK, even in trunk. If you can resolve them quickly, it would be great.

Do you mean these?:

```
[exec] [ant:javadoc] /home/evl/testing/majic/p2j/src/com/goldencode/p2j/admin/client/application/home/HomePresenter.java:1174: warning - @result is an unknown tag.  
[exec] [ant:javadoc] /home/evl/testing/majic/p2j/src/com/goldencode/p2j/admin/client/application/home/accounts/users/UsersPresenter.java:666: warning - Tag @link: reference not found: ResetPresentersEvent  
[exec] [ant:javadoc] /home/evl/testing/majic/p2j/src/com/goldencode/p2j/admin/client/application/home/accounts/users/UsersPresenter.java:666: warning - Tag @link: reference not found: PresenterWidget  
[exec] [ant:javadoc] /home/evl/testing/majic/p2j/src/com/goldencode/p2j/ui/WindowConfig.java:252: warning - Tag @link: can't find getWindowState in com.goldencode.p2j.ui.ConfigHelper
```

If yes, I'm going to fix them too. If other - please point me what you mean.

#264 - 09/26/2017 08:11 AM - Greg Shah

Do you mean these?:

Yes.

#265 - 09/26/2017 10:39 AM - Eugenie Lyzenko

The task branch 1834b has been updated, rebased with trunk 11168. New revision is 11195.

This is round rectangle fix and javadoc fixes. Going to start regression testing.

#266 - 09/26/2017 02:44 PM - Eugenie Lyzenko

The task branch 1834b has been rebased with trunk 11169. New revision is 11196.

The regression testing is restarting.

#267 - 09/26/2017 03:50 PM - Eugenie Lyzenko

The task branch 1834b has been updated to revision is 11197.

Ironically I have missed the round rectangle fix. Now it is in branch.

#268 - 09/27/2017 06:44 AM - Eugenie Lyzenko

The main regression cycle has passed. Waiting for CTRL-C results.

#269 - 09/27/2017 07:51 AM - Eugenie Lyzenko

Testing completed. No regressions found. Had to run 3-way tests in standalone session. The results: 1834b_11197_ee1fc80_20170927_evl.zip.

So the branch is now ready to be merged into trunk I think.

#270 - 09/27/2017 08:35 AM - Greg Shah

Please merge to trunk.

#271 - 09/27/2017 08:48 AM - Eugenie Lyzenko

Greg Shah wrote:

Please merge to trunk.

Done.

Branch 1834b was merged to trunk as revno 11170 then it was archived.

#272 - 09/27/2017 10:28 AM - Greg Shah

- Status changed from WIP to Closed

- % Done changed from 0 to 100

Files

dnd_test0_1_fwd_20170703a.jpg	42.6 KB	07/04/2017	Eugenie Lyzenko
dnd_test4_fwd_20170705a.jpg	53.4 KB	07/06/2017	Eugenie Lyzenko
dnd_test4_2_4gl.jpg	66.4 KB	07/06/2017	Eugenie Lyzenko
rectangle-select-and-drag.png	2.21 KB	07/06/2017	Ovidiu Maxiniuc
web_client_20170707.jpg	84.5 KB	07/09/2017	Eugenie Lyzenko
dnd_test0_3_0_4gl.jpg	26.1 KB	07/22/2017	Eugenie Lyzenko

dnd_test0_1_fwd_web_20170731a.jpg	108 KB	08/01/2017	Eugenie Lyzenko
sel_box_web_20170811b.jpg	112 KB	08/12/2017	Eugenie Lyzenko
sel_box_web_20170811a.jpg	98.4 KB	08/12/2017	Eugenie Lyzenko
dnd_test0_1_4gl_win10.jpg	108 KB	08/15/2017	Eugenie Lyzenko
test_issue0.jpg	93 KB	08/17/2017	Eugenie Lyzenko
set_box_on_darkgray_4gl_classic.jpg	57.8 KB	08/18/2017	Eugenie Lyzenko
set_box_on_darkgray_4gl_win10.jpg	55.2 KB	08/18/2017	Eugenie Lyzenko
set_box_on_darkgray_fwd_classic.jpg	49.8 KB	08/18/2017	Eugenie Lyzenko
dnd_test3_fwd_20170824a.jpg	45.6 KB	08/24/2017	Eugenie Lyzenko
dnd_test3_fwd_web_20170825a.jpg	58.6 KB	08/25/2017	Eugenie Lyzenko
dnd_test3_fwd_web_20170829a.jpg	62.8 KB	08/30/2017	Eugenie Lyzenko
Хатха-Йога для начинающих.djvu	1.57 MB	09/01/2017	Sergey Ivanovskiy
dnd_test3_fwd_web_unicode_20170901a.jpg	42.5 KB	09/01/2017	Eugenie Lyzenko
MsgsInterleaving.png	28.7 KB	09/03/2017	Sergey Ivanovskiy
gui_button_test4_issue_fwd_20170925.jpg	32.1 KB	09/26/2017	Eugenie Lyzenko