Release Notes OWrite/JSOWrite version 4.2.0.0

18 Feb 2019 12:32

JSOWrite Bug Fixes

ID :	1719	**Fixed in version :**	4.1.4.0
Short Description:	Problem with Justified text with bullets list		

Full Description: I find a problem with bulleted list and text alignment.
If I insert a bulleted list in a new document and then I try to justify the text in the list, as a result the text isn't justified.
In the previous version of the component, the text embedded in the bulleted list, was justified correctly.

Comments : We confirmed that the text justification was not working for the kWriStyParaJstFull option for text in bullet lists nor was the correct justification displayed in the JS interface for kWriStyParaJstCenter and kWriStyParaJstRight even when the text was correctly justified.

It is possible that this was a regression, but we were unable to confirm when this may have stopped working.

JSOWrite Enhancements

ID :	1720	**Implemented in version :**	4.1.4.0
Short Description:	Improvements to copy functionality		

Full Description: We have added some special code to better handle copy content requests by the browser, thus improving the appearance of that content when pasted to other word processors.

Comments :

ID :	1721	**Implemented in version :**	4.1.4.0
Short Description:	Support of field view display.		

Full Description: We have added support for the kWriViewField display mode. Please note: The value of the $pageview property is stored with document data, so changing this mode and saving the document will cause the document to be displayed in the new display mode the next time it is loaded by OWrite Desktop or JS-OWrite.

Comments :

ID :	1714	**Fixed in version :**	4.1.3.2
Short Description:	Impossible resize columns on table		

Full Description: When I set the parameter '$pageview' to 'kWriViewNormal', I can't resize entire columns on tables (it's resized the cell only).

Comments : This issue was caused by the floating point changes in OWrite version 4.1. Further more, because of the different behaviour required by Cocoa when drawing sizing rects, we had to do some changes in the way these were drawn in general.

ID :	1715	**Fixed in version :**	4.1.3.2
Short Description:	Strange behaviour resizing columns		

Full Description: When you resize columns on a table using Win 10, a line is displayed all over the document.

Comments : This issue was caused by the Studio Cocoa changes on Macintosh. Because of the different behaviour required by Cocoa when drawing sizing rects, we had to do some changes in the way these were drawn in general.

ID :	1716	**Fixed in version :**	4.1.3.3
Short Description:	Random crashes and other problems		

Full Description: A customer use oWrite component under macOS to evaluate documents starting from own templates.
Crashes occurs quite often simply modifying oWrite documents, without a particular actions sequence.
Sometimes happens, sometimes no.
Sincerly I don't know how face this problem.
Tell me if there are any information to collect and send you to investigate the problem.
Furthmore, with the customer's DB when some documents are opened, Omnis freeze and it's necessary force the quit.

For tests we used oWrite 4.1.3.2 (sended last week).

Thanks in advance

Comments : There was a problem with c++ object destruction which could cause intermittent crashes or hangs.
This was a regression introduced during our major font-handling rewrite.

ID : 1722 **Implemented in version :** 4.1.4.0

Short Description: Method $picturefrompage enhancements

Full Description: We have made two enhancements to the $picturefrompage method

1) High-res snap shot support:
Passing zero for the width and height parameters (parameters 2 & 3) will cause the function to return an image at full screen resolution. This mode will consider $docscale, making it possible to return high resolution images of the specified document page.

2) Taking snap-shot of specified object:
We have added a new parameter. When five parameters are passed to the method, the fourth parameter will specify an object ID for taking a snap-shot of the specified object only (instead of the entire page) such as a table field. What originally was the fourth parameter (the variable that receives the image) is passed in the fifth parameter. The function will automatically detect if four or five parameters have been passed and behave accordingly.

Comments :

Release Notes JSOWrite version 4.1.3.0

29 Nov 2018 22:36

Enhancements

ID :	1700	**Implemented in version :**	4.1.3.0
Short Description:	Problem with jsOwrite in subforms		

Full Description: If I use jsowrite in a 2 subforms in different pagepane within
the same form and I switch between panes, in one pane owrite disappears.
See form rfMain in the OWrite Javascript folder in the example.

AttachedFile: example.zip

Comments : It is now possible to place multiple JS-OWrite controls within the same remote form or within multiple subforms within the same parent form. Multiple JS-OWrite controls may also share one set of interface controls (JS-OWrite property controls). The JS-OWrite control that receives the focus will take over the current set of property controls within the main form. Each JS-OWrite control may also have its own set of property controls when using OWrite within subforms.

The following warning concerns the use of JS-OWrite within page-pane controls.
When using multiple instance of OWrite with the page-pane control, we found that the page-pane control only maintains the html content for the current page. When a page is closed, the entire content is destroyed, including the instance of OWrite. That means, every-time you show a page with an OWrite control, all the document handling is re-loaded and the current input position is lost. You must also save the current document to your instance variable, prior to the page changing or any changes are lost. There is no solution to this, other than lobbying Omnis Software to change the behaviour of the paged pane.

A solution to the paged-pane issue is to use the subform control using the $multipleclasses feature. Different remote form classes can be shown within the same subform control and all are maintained when hidden.

We have added some simple examples that implement both solutions to the Document Manager examples.

rfOWriteSimpleMultiSubMain - demonstrates the use of subforms and $multipleclasses to toggle between two instances of OWrite

rfOWriteSimpleMultiPage - demonstrates the use of the paged pane with subforms to toggle between two instances of OWrite

Release Notes JSOWrite version 4.1.2.0

17 Aug 2018 11:52

Bug Fixes

ID :	1676	**Fixed in version :**	4.1.2
Short Description:	Deleting table cells breaks table evaluation		

Full Description: After deleting some rows in the Write table, the table no longer evaluates against the data list. The list data is not displayed after evaluating the document.

Comments : The problem was with the Java script on the client when deleting rows in a table using JS-OWrite. When creating a document with a table with multiple rows, and then remove some rows from the table using the JS-Client, the document only removes the cells, but does not correctly remove the table rows, which are left empty in the document data. This breaks the evaluation of the document on the server.

Please note: Existing documents that have been effected by this will be automatically repaired when loaded and saved on the client.

Release Notes JSOWrite version 4.1.0

Bug Fixes

ID : 1482 **Fixed in version :** 4.1.0

Short Description: Error in Pict property

Full Description: JSowrite Alpha 3

In your example library, if I insert a sample picture,
some property of kWritypePict don't work:
- wrapping style Inline and Split
- Inline alignment ($curobjalign)
- Wrap margin ($curobjwdtop,$curobjwdleft,$curobjwdright,$curobjwdbottom)

Comments : All three issues have been resolved.

ID : 1578 **Fixed in version :** 4.1.0

Short Description: font size change issue

Full Description: changes to font size does not redraw the selected text, changes appear leaving the size field

take a look to the attached movie

{File: fontsize720.mov}

Comments :

ID : 1598 **Fixed in version :** 4.1

Short Description: Copy&Paste Issue

Full Description: Copy and paste first paragraph of JSClient demo document: copied text is not the same of original.

Link to video: https://drive.google.com/open?id=0ByaFJoHYxPBzLVlSdlZEbGdGZkU

Comments :

ID : 1634 **Fixed in version :** 4.1.0

Short Description: font change unexpectedly

Full Description: font change inexpectly during the text editing.

we have reproduced a case using your example here

http://demo.888sp.com:50000/jschtml/owritedemo.htm

and you can see a demo video attached

AttachedFile: font_change_unexpectly.mp4.zip

Comments : I have now been able to reproduce this issue. I misread the instructions. I read back space not back arrow (which is actually called left-arrow in english).

ID :	1636	**Fixed in version :**	4.1.0
Short Description:	issue resizing table		

Full Description: resizing the table columns work as we expect but if you try to resize the right border you can see a different behavior. Seem not possible to resize the table border, it only apply the change to the cell but if you try there is something wrong, ghost lines appear and is not simple to return to the original position.

AttachedFile: resizeTable.mp4.zip

Comments : We were able to reproduce this issue and are investigating.

ID :	1637	**Fixed in version :**	4.1.0
Short Description:	Tabs on table issue		

Full Description: using tabs keys in a table cell and backspace key, it destroy the cell and the table become not deletable.
Take a look to the attached video

AttachedFile: tab_table_issue.mp4.zip

Comments : Although we have reproduced the issue there is some miss-understanding on what counts as selected cells, but first-

The bug I identified is that pressing backspace deletes a cell. I am not convinced this should happen and I need to speak to Perry about this.

Now the miss-understanding. Cells are selected by dragging from within a cell across more than one cell. Once the cells are selected one can delete the cells. See attached video. I think we prevented deleting of table cells in any other way because of some technical issues such as undo, but I need to speak to Perry about this to re-establish what these were exactly.

However, this leaves us with the issue when there is only one cell as it appears impossible to select a table consisting of a single cell.

Anyway, hope that the drag-selecting of cells helps right now. We will work at this urgently to resolve the issues as we see them.

ID :	1638	**Fixed in version :**	4.1.0
Short Description:	table over the paper area		

Full Description: if you insert a column in a table and it goes over the paper area is not more possible to resize it

also using cell properties size does not work

look at the attached video

AttachedFile: table_over_.mp4.zip

Comments :

ID :	1639	**Fixed in version :**	4.1.0
Short Description:	table not erasable		

Full Description: a table seem not be erasable

look the attached video with your example

AttachedFile: table_not_erasable.mp4.zip

Comments :

ID :	1641	**Fixed in version :**	4.1.0
Short Description:	different display of the document between screen and PDF		

Full Description: The document that I see on JSClient is different from the same document printed on pdf.
As you can see in the example on http://demo.888sp.com:50000/jschtml/owritedemo.htm,
the document OWRITE\Example1 is differnet from screen to PDF (if you print the document on pdf you can see the difference).
You can see the difference with the document owrite\welcome.

Comments : IMPORTANT!!! READ THESE NOTES PRIOR TO USING THE 4.1GM RELEASE!!! IF YOU HAVE FURTHER QUESTIONS RELATING TO THESE CHANGES, PLEASE CONTACT BRAINY DATA TECHNICAL SUPPORT.

WHAT WE HAVE DONE

A detailed investigation revealed that none of the six major browsers (Chrome, Opera, Firefox, Safari, Edge and IE) measured and formatted text identically within a single platform or across platforms. This applied both to measuring text widths and heights. Consequently, it was technically unviable for the OWrite server object to adopt a strategy of formatting pages to match the screen output of the browsers as this would lead to inconsistent page output depending on which browser a client would use. Thus it was decided that we needed to find a standard of measuring text and find a way to make browsers conform to this standard. It was decided that this standard was to be MSWord. Unfortunately, we discovered that OWrite's output, as well as the browsers', did not always agree with MSWord. Just as with the browsers, it very much depended on the fonts and sizes used. By reverse engineering the issue, we found that the inconsistencies were caused by the Omnis external SDK text functions being integer based. Integer based units work for some text sizes, but not for all. The painful decision was taken to abandon the external SDK functions in favour of direct system calls for measuring text which supports floating point units. This required many changes throughout OWrite. From hereon, OWrite measures and formats text using floating point units which allows the accurate scaling of text to any screen resolution and which achieved near 100% compatibility with MSWord (there are some rare cases where minor difference occur involving more complex pages and table rows). Next we set out to fix the browsers which took two forms. Firstly, to overcome the different font heights reported for the same font and size by different browsers, we added a feature so that OWrite provides the standard font heights which can be downloaded to the client and which our scripts will use to position text vertically on a page (see next section). Secondly, to overcome the different ways browsers measure text widths, we applied browser specific javascript code that remedies these inaccuracies. As a final measure, to combat boundary conditions involving table rows and complex pages, JS-OWrite client now stores page boundary information within the document to ensure that server output (i.e. PDF) match the output within JS-OWrite on the client.

Please note that we DO NOT claim 100% cross browser/cross platform consistency. We recognised that there may still be some small differences in formatting involving more complex pages. However, most standard documents should now appear identical between browsers, MSWord and OWrite.

WHAT YOU MUST DO

In OWrite version 4.1 there are three things you have to do

1) to enter the new cross-platform compatibility mode, set $printdpi to zero and $screendpi to 96 in the desktop visual components and desktop/server non-visual components. When creating a new JS-OWrite client or desktop control or non-visual object, $printdpi should automatically default to zero. WARNING: in previous versions assigning zero to $printdpi would select the platform's native resolution. This feature has changed and $printdpi must now be set to -1 so the native resolution is used. Zero is now reserved for telling OWrite to use a font's EM design units for ultimate accuracy in measuring text.

2) download a fonts height list from the server for JS-Client
The OWrite Document Manager examples show you how to do this. Search for ivFontSizeList in rfOWriteSuper. Ignore matches in other classes (while writing these notes we realised that an identical list exists in a few other places throughout the document manager examples). To optimize JS-OWrite (reduce the number of fonts) we strongly recommend you use the OWrite font map feature, in conjunction with the $getfontlist feature, to limit the fonts available to the user on the client. For an example on how to use this feature, from the "*** OWrite Plus Examples ***" menu select "OWrite Examples" -> "Edit Font Map". Clicking the more info button provides further details.

3) for desktop window controls that are to remain more compatible with screen sizes there are two sets of settings you should consider

3.a) For platform specific display (72dpi on mac and 96 dpi on windows) set $printdpi to -1 and set $screendpi to kWriScrDPIDefault.
3.b) For cross platform display set $printdpi to 96 and set $screendpi to kWriScrDPI96.

These two sets give best results. On the Mac, OWrite always uses fractional point sizes and the Mac is better at scaling fonts for multiple resolutions, resulting in a near perfect screen display even when $printdpi is set to zero. On windows we are still investigating a solution at making text on screen scale across all point sizes and screen resolutions more accurately. For now, we have to stick to non-fractional point sizes and manipulating the spacing between characters at whole pixel boundaries. This may appear less perfect at some resolutions and scaling factors.

WIDER CONSEQUENCES

By abandoning the Omnis external SDK when measuring text, OWrite provides consistent positioning of text on a page, across multiple platforms and resolutions. However, there are two important consequences. Firstly, the changes to OWrite were substantial and require thorough testing within your own implementations before deciding to move permanently to OWrite version 4.1. It may require some changes to your libraries and/or documents. Secondly, OWrite document text send to the Omnis print manager may suffer from extra wide spacing between words or words running into each other. This will be most noticeable on Windows as certain point sizes do not scale accurately to 96dpi screen display. For example, a 8pt font should ideally be scaled to 10.66(recurring)px size, but as Omnis still renders fonts using the standard GDI interface, the px font size has to be rounded to 11px. Consequently the text is displayed using a slightly larger font than OWrite used to measure the position of words, resulting in words running into each other. Equally, when displaying a 10pt font this should calculate to 13.33(recurring)px size, but Omnis will scale this to 13px height and the spaces between words will appear extra large. At larger point sizes, this effect will become less obvious as the inaccuracies decrease percentage wise.

FUTURE WORK

Currently, on Windows, we are unable to use text rendering functions that will allow us to specify floating point font sizes. Being able to use floating point font sizes would allow us to render fonts on screen as accurately as OWrite already renders on Macintosh. The Windows SDK which offers this possibility is GDI+. Unfortunately, using GDI+ would require our software to use STRICT compilations, whereas the Omnis SDK is build with the NOSTRICT option which makes it impossible to build and link both SDKs into the same DLL. We will continue searching for possible solutions to this issue.

ID :	1643	**Fixed in version :**	4.1.0
Short Description:	JS-OWrite proprety-control disappear		

Full Description: If I use the form with JSOwrite in a subform, the second time I open the form, the proprety-control disappear. I must to close and reopen the browser to view them again.

AttachedFile: image.JPG

Comments : This was a redraw issue with controls in sub-windows, when the sub-windows were downloaded on a need basis, after the document had been loaded. Clicking in the document content, typically fixed the issue by causing an additional redraw.

ID :	1644	**Fixed in version :**	4.1.0
Short Description:	OWrite report object		

Full Description: I'd like to print the JSOwrite document in a Omnis report
with Owrite Object report (like a old version) but now I haven't a binary
variable but a list. How can I do? I can't print to a PDF directly.

Comments : We have now changed OWrite so it auto-detects the list format when loading OWrite default document formats. Consequently, the OWrite report object can now load documents that are saved as list. As a side-effect, this means that when calling $loaddata it is no longer necessary to specify the kWriSaveAsList parameter. However, kWriSaveAsList and kWriSaveRawPicts must be specified when calling $savedata.

ID :	1658	**Fixed in version :**	4.1.0
Short Description:	image from Owrite WebClient 3.0 to JSOwrite 4.0.2		

Full Description: I have a document with version owrite web client 3.0 with a
picture (see image1.png), all the variables are reported in
the new version jsclient 4.0 except the picture (see image2.png)

AttachedFile: image.zip

Comments : We have identified the problem which is an issue in the JS-Client core editor script. It appears the picture objects in question do not have a width or height specified (both a zero) which of course is a valid state. In such a case OWrite should be using the original width and height settings. It appears the JS-Client OWrite is failing to do that. This has been corrected.

ID :	1659	**Fixed in version :**	4.1.0
Short Description:	Problem with "Canc" in table		

Full Description: If I use the "canc" button on the keyboard when I'm inside a table cell , the result is that is deleted the letter before the curson but It's wrong.
When I press the "canc" button the letter that should be cancelled is the one after the cursor.
If I'm outside the table, it works correctly.

Comments : The description is referring to the 'del' key on the english keyboard. There was a small logic problem handling key exceptions when inside tables. This has now been resolved.

ID :	1660	**Fixed in version :**	4.1.0
Short Description:	Image in Table cell		

Full Description:	You can reproduce the problem with your library: document "Example: Table Picture" My test page is on http://demo.888sp.com:50000/jschtml/owrdemo.htm When I evaluate the document the cell don't expand to fit the image
Comments :	This issue was introduced during 4.1 alpha 2 release and has been resolved for alpha 3.

Release Notes OWrite version 4.0.1

Bug Fixes

ID : 1618 **Fixed in version :** 4.0.1

Short Description: Editclear() do nothing without return value

Full Description: in your example library, please import the document in attached zip file.
The document presents 2 tables. In the first one, select the field "<Pour chaque compogravure" (1st table, line 2, 1st cell, 2nd line)
Now, if I press the del key, nothing happens.
Same result if I use context menu "clear" and method $editclear() returns 0 as if all was ok...

NB: consider our software searches and deletes this kind of fields before calling for $eval as part of our process

Please, could you tell me what is wrong with this ?
Thank you in advance.
Joel

AttachedFile: PbEditClear.zip

Comments : For case 1527 we added some code to prevent the accidental merging of two adjacent tables. However, the code failed to accurately detect when deletion was occurring within the confines of a table cell. This has now been rectified.

ID : 1620 **Fixed in version :** 4.0.1

Short Description: Problems with Export to JSON and Import from JSON Case CA08611

Full Description: The issue is when you export a library that uses you oWrite component to JSON, then some of the properties are not exported as expected.
See the attached example library, JSON export and image.
The properties found so far are $backcolor & $forecolor.

Comments : We have updated OWrite with new flags required by the JSON export.