

Bug Fixes

ID : 1232**Fixed in version :** 3.2.1**Short Description:** Loading table via RTF loses formatting**Full Description:** When converting an OWrite table to RTF and later load it again, any empty cells lose their formatting, i.e. the 9 point font size changes to 10 point. Any cells that contain text stay at 9 point.**Comments :** This problem is related to the last paragraph in a document or last paragraph in a table cell or text box. The last paragraph in any of these circumstances does not contain a final CR character and therefore has a zero character count if it does not contain any other text. Due to an oversight, paragraph specific style information that normally overrides the paragraph's document style was not applied in such circumstances.

For example:

- the last paragraph in a table inherits the 'Normal' style which specifies Arial 10pt
- if one selects the cell and changes the font size to 9pt, all paragraphs will override the 'Normal' style's 10pt
- when loading such a document via RTF, the final paragraph in the cell will fail to override the point size during the import and revert back to the 'Normal' style which is 10pt.

This issue was partially fixed in 3.2.0 but required some additional work to resolve all formatting issues related to final paragraphs.

ID : 1235**Fixed in version :** 3.2.2**Short Description:** Slow rendering Owrite**Full Description:** A user try to open the file attached on web client. When he try to open this file the web client freeze and after that crash. The document is 3 MB and the network is fast. In fact, to send the document to the client takes little time. The problem is when owrite render the document. There is any solution?**Comments :** We were unable to reproduce any crashes. The file took a long time to load due to its large size (109 pages of table fields). On our test machine it took approximately 16 seconds to load. We have discovered a optimisation issue and once we have resolved it, it has reduced the loading time to about 7 seconds. This is the best that can be achieved.

The problem was not slow rendering, but a slow loading time due to the fact that the document was saved in an evaluated state and that the data in the table cells were resizing each cell during loading. This may be avoidable if such large documents are saved (\$savedata) with the third parameter set to kTrue so that the evaluated data is made permanent.

During final testing it came to light that the bMakeDataPermanent option during \$savedata calls did not work correctly for table rows that were added during evaluation. This has now been resolved as part of this case.

PLEASE NOTE: it may not be possible for existing documents to benefit from re-saving them with this flag enabled. However, evaluated documents that are saved with bMakeDataPermanent set to kTrue while \$evalcales is also kTrue, should subsequently load much faster.

ID : 1237

Fixed in version : 3.2.5

Short Description: owrite export to html replacing '&' with '&'

Full Description: I think this is a change to behaviour as things that work are now not. My call to get the html out of owrite is

Do

```
iOWrite.$savedata(binary,kWriFmtHTML,kTrue,kWriHtmlNoMargins,kTrue,kWriHtmlSizeAdjust,kF  
alse,kWriHtmlRawSupport,kTrue,kWriHtmlNoAutoSize,kTrue,kWriHtmlBgColor,iOWrite.$papercolo  
r)
```

all the '&' in the exported 'html' seems to be replaced by &

for example, the .Header_&_Footer style is .Header_&_Footer and that may or may not affect the page rendering... I don't know

```
.Header_&amp;_Footer\t{text-align:left;margin-top:0;padding-top:0pt;margin-bottom:0;padding-botto  
m:0pt;font-family:Times New
```

```
Roman;font-size:10pt;font-style:normal;font-weight:normal;text-decoration:none;vertical-align:top;ve  
rtical-align:top;color:#000000;text-decoration:none;}r\n\t.Header_&amp;_Footer_table
```

the more important part is my rendering of pictures. I am creating a URL that is:

```
https://127.0.0.1/TheatreManager/1/picture?id=2&pmb_seq=217&tick=2278715
```

and it is coming out of the owrite exporter with the the '&' replaced by '&' as follows:

```
<img src="https://127.0.0.1/TheatreManager/1/picture?id=2&amp;pmb_seq=217&amp;tick=2278715  
etc..
```

the problem with the & in this case is that it makes the parameters that follow in the URL ineffective the browsers and our servers won't process the parameters where the & is. I can fix it temporarily by replacing the & after the export...

but I'm not sure that a global change that the html export function is doing is wise in this case.

Comments : OWrite has always converted the '&' to the html "&" escape sequence within an image's SRC attribute (we have tested this as far back as version 2.7) . This was probably an oversight but it appears that all browsers and servers handle this correctly so it was never noticed. As it is not necessary to convert '&' characters when correctly used to append parameters to an image's URL we have changed this behaviour.

IMPORTANT: The URL of an image that is provided during export to HTML, must be valid URL and the '?' and '&' characters should only be used to append additional parameters to an URL.

ID : 1238

Fixed in version : 3.2.5

Short Description: OWrite Table Performance

Full Description: it seems that the time to fill an OWrite Table rises exponentially with the line count of the bound Omnis List. To demonstrate this behaviour we attached an example lbs.

The lbs contains an Omnis Windows Class which generates an Omnis Report with an OWrite Table. The first entry field defines the number of lines in the bound Omnis List and the second outputs the amount of seconds it took to generate the report.

In our testing environment using OWrite 2.7 we came up with the following Result (Windows 7 64-bit):

Line count	Seconds to generate the report
50	< 1s
100	< 1s
250	1s
500	3s
750	7s
1000	19s
2000	173s
4000	1639s

This makes it nearly impossible to print more than 1000 Records.

Comments : Please note that although OWrite is ideally suited for producing end user reports due to its customizable nature, it has different mechanics to content with. OWrite allows the insertion, removal or editing of content at any point within a document which will impact existing content and therefore has overheads that the Omnis report generator does not. When evaluating documents against large lists, there will always be a certain amount of exponential reduction of performance. However, we isolated some unnecessary pagination relating to table fields and discovered a redraw issue that caused these rather large exponential decreases in performance that you experienced. After rectifying these issues, our own test show a very substantial increase in performance when loading or evaluating documents with large lists/tables. Using the provided example, OWrite evaluated 8000 lines/rows in less than 20 seconds.

WARNING: Some of the changes were radical and this new release requires thorough testing before distributing it.

ID : 1240

Fixed in version : 3.2.5

Short Description: OWrite does not redraw when changing \$papercolor

Full Description: When changing the \$papercolor within evSetFocus and evKillFocus oWrite doesn't redraw right away and I can't seem to force it to.
Once I start typing or when I look with the notation inspector I can see the \$papercolor has changed.

Is there a command to force oWrite to redraw itself?

Comments : OWrite was not forcing a redraw when the paper color was changed during event handling.