

## JSOWrite Expelled Gremlins

**ID :** 2016**Fixed in version :** 5.6.0.0**Short Description:** Mergedoc unexpected behaviour

**Full Description:** i'm experiencing a strange behaviour with the document merging function. With one particular window, when i change from "merged" to "non merged", JSOWrite goes crazy and it adds several variables.

In the attached zip there is:

- a sample library with a working example and a non working example
- a video with the working example
- a video with the bugged example

AttachedFile: mergedoc.zip

**Comments :** {none}

**ID :** 2017**Fixed in version :** 5.5.1.0**Short Description:** Document horzscroll goes over the document

**Full Description:** i've noticed that, when scrolling horizontally the document, the "page" goes over the vertical ruler.

AttachedFile: jsowrite\_document\_over\_ruler.gif

**Comments :**

**ID :** 2018**Fixed in version :** 5.6.0.0**Short Description:** JSOWrite scrollbar - problem with resizing

**Full Description:** when the JSOWrite is resized, the horizontal scrollbar get misplaced and part of the component is grayed out.

I've attached a sample library and a GIF showing the issues.

AttachedFile: Resize Example.zip

**Comments :** {none}

**ID :** 2022**Fixed in version :** 5.6.0.0**Short Description:** Reset text properties to default style

**Full Description:** is there a way to reset the style of a text run to the default values for a given style in JSOWrite?

For instance, let's say that i have a style "Heading 1" with black text, and the user by mistake perform some changes (i.e. changes the text color, makes it bold, etc.).

There is a way to let the user "go back" to the default "Heading 1" appearance?

In the desktop version, setting again the style seems to do the job, but on JSClient it doesn't seems to work.

Any suggestions?

**Comments :** {none}

**ID :** 2023

**Fixed in version :** 5.6.0.0

**Short Description:** Calculated fields styles fall over

**Full Description:** i've noticed that, when typing text after inserting a calculated field or an object info, there are certain key combinations which alter the style.

It's a bit hard to explain by words, so i've attached two gifs that demonstrate the issues.

**Comments :** {none}

**ID :** 2024

**Fixed in version :** 5.6.0.0

**Short Description:** Strange header style with ObjInfo

**Full Description:** i've noticed that if i put some object info (like the pagenumber / pagecount) in the document header and i make them right aligned, there full header line has a light green back color when printed in PDF.

**Comments :** {none}

**ID :** 2026

**Fixed in version :** 5.6.0.0

**Short Description:** Table style changes aren't save

**Full Description:** i've noticed that if i select a table and change the text style from BOLD to regular, the change isn't properly saved, and when the document is merged, the text is still bold.

AttachedFile: tablestyle.gif

**Comments :** {none}

**ID :** 2029

**Fixed in version :** 5.6.0.0

**Short Description:** Merged document doesn't reset styles for InfoObjs

**Full Description:** I've noticed that when I print a merged document which has both an ObjInfo (page count) and a calculated field, the calculated field is displayed with the correct text style, but the ObjInfo keeps the altered style.

It is possible to merge the style of the ObjInfo like with the other calculated fields?

I've attached an example gif.

**Comments :** {none}

**ID :** 2030

**Fixed in version :** 5.6.0.0

**Short Description:** jsWrite controls do not display in design mode

**Full Description:** Neither the jsWrite nor the property controls display in design mode

**Comments :** It appears the Studio 'html' folder includes a 'design' folder which appears to have been introduced in Studio 10.2. For the design mode, Studio creates copies of the remote forms within this folder instead of the root of the 'html' folder, which means the 'design' folder requires a copy of the additional 'owrite' folder that ships with jsWrite.

Further changes were also required within the ctl\_owrite.js script file for design mode to function correctly.

Consequently, for design mode to work correctly, copy the folder 'owrite' to both the 'html' folder and 'html/design' folder.

**ID :** 2034

**Fixed in version :** 5.6.0.0

**Short Description:** image keepaspectratio doesn't always work

**Full Description:** i've notice a small issue with the behaviour of \$curobjlockaspect.  
If the flag is set to true and the image is resized using the handles, it works fine.  
But, if the image is resized by changing \$curobjheight or \$curobjwidth, the flag is ignored and the aspect ratio gets changed

**Comments :**

## OWrite Expelled Gremlins

**ID :** 2007

**Fixed in version :** 5.5.0.1

**Short Description:** Click calculation results corrupt when exporting to HTML

**Full Description:** Exporting click calculations was broken by a change for case 1987. When \$owrite\_export returns RAW html for either HREF or SRC links, these links became corrupt in the final HTML output.

**Comments :**

**ID :** 2008

**Fixed in version :** 5.5.0.1

**Short Description:** Click calculation result data not exported to plain text

**Full Description:** Calculated fields with evaluated click calculations showed up as empty when exporting to text documents.

**Comments :**

**ID :** 2019

**Fixed in version :** 5.5.0.1

**Short Description:** OWrite ruler doesn't display number

**Full Description:** On macOS 13 and 14, with the latest version on Omnis (11 build 35659), the OWrite ruler doesn't display the numbers on the horizontal ruler. See attached screenshot.

**Comments :**

**ID :** 2027

**Fixed in version :** 5.6.0.0

**Short Description:** Strange header style with highlighted text

**Full Description:** this is a spin-off from case 2024:

We noticed that the same issue exists with right-aligned paragraph where the last piece of text is highlighted. It applies only to the last line in a header, footer or body text (i.e. a line that is not terminated by a CR character).

**Comments :** {none}

**ID :** 2031

**Fixed in version :** 5.6.0.0

**Short Description:** The horizontal ruler numbers are no longer displayed

**Full Description:** The horizontal ruler numbers are no longer displayed on macOS 12 and higher and Omnis Studio 11 and OWrite 5.5.

With Omnis Studio 11 and OWrite 5.5 and Windows 11, the numbers are displayed.

You can see it in the attached screenshots.

AttachedFile: Captures.zip

**Comments :** {none}

## OWrite Enhancements

**ID :** 2006

**Implemented in version :** 5.5.0.1

**Short Description:** Ability to attach non-visual object to oWrite window control

**Full Description:** We have added a new function that allows you to attach the oWrite non-visual object to the oWrite window control so that the document being edited can be directly manipulated or printed and exported via the non-visual object's methods. This will allow one to create non-visual object classes that contain all your evaluation, printing and exporting methods without having to re-create them in different windows or having to save and load the document data for use in your non-visual object. Once a non-visual object is attached to the window control, it has full access to the loaded document data.

Usage:

Do windowRef.\$objs.OWriteControl.\$attachobj(OWriteNVObj.\$owriteobj)

Your OWriteNVObj object variable must be of subtype ".OWrite.OWriteDoc" or your own object class that has ".OWrite.OWriteDoc" as the upper-most super class.

**Comments :** The intended use of this feature is to be able to separate common code from your window classes so that this code can be shared by all window controls as well as your non-visual tasks that involve document editing, merging, printing, importing and exporting. In other words one object that can do it all.

**ID :** 2028

**Implemented in version :** 5.6.0.0

**Short Description:** Calculated fields, text-boxes and \$docinsert now support 7bit & 32bit RTF

**Full Description:** The method \$loaddata(kWriFormatRTF,...) always supported both 7bit rtf in binary format as well as rtf stored as 32 bit unicode data. Other methods of inserting RTF (calculated fields, table cells, text boxes & the \$docinsert method) only supported 32bit unicode which is the standard Omnis character data.

This change means that results for calculations that expect text or the text data passed to \$docinsert(kWriObjTypeText,...) can now also accept 7bit RTF binary data

**Comments :**