

## Bug Fixes

**ID :** 1300**Fixed in version :** 3.6.1**Short Description:** Titillium Web font issues on Windows**Full Description:** Hate to bother you with another font issue. Attached is a font called Titillium Web.

This font works beautifully on Mac and didn't give us any problems on Windows until I change our app to not use the font list returned by Omnis, but the font list returned by oWrite.

It finds Titillium Web just fine but only on the Mac all the typefaces are there. On Windows only the Light typeface is there.

I have no problems reproducing this with the oWrite sample library, just select the Titillium Web font from the dropdown and you'll see there is only one typeface.

Any ideas why? Is the font wrong? Is oWrite skipping something?

**Comments :** On the Windows platform, as far as we can tell, the Omnis notation `FontOps.$winlistfonts()` returns a list of fonts that does not support any of the typefaces provided by "Titillium Web" other than bold or italic. Omnis Studio 6.0.2 appears to build a list of fonts based on the font family name which does not encode the typefaces thus there is no mechanism for picking some of the more fancy typefaces.

OWrite has always attempted to emulate what we call the Omnis internal font names as we believed we must pass the correct font names to Omnis via the Omnis callback functions, thus OWrite was limited in the way it arrives at the font names.

However, after careful investigation we can confirm that it is possible for OWrite to handle all typefaces provided by this font if we use the TrueType-face-name with the Omnis callback functions as Omnis appears to pass through the TT-face-names to the Windows operating system unaltered and the system appears to recognize these names and selects the correct font. We therefore have changed OWrite to use the TT-face-name on the windows platform. Existing documents should not be affected by this change. Thus what we call the Omnis-font-name has become the TT-face-name on the windows platform.

Unfortunately, this highlighted another cross-platform complication. To explain let's use "Titillium Web Black" as an example.

On Macintosh the Omnis-font-name is similar to the TT-face-name. Both the TT-face-name and the Omnis-font-name on the Macintosh combine the family name with the style name to create a unique font name. However, on the Macintosh Omnis appends the style name with a dot but a TT font appends the style name without any separator so "Titillium Web Black" becomes on Macintosh "Titillium Web.Black" and on Windows "Titillium WebBlack"

We cannot alter what font name we use on the Macintosh as Omnis does not simply pass the fontname through to the system and it is the Omnis font name that OWrite stores in document data which we are reluctant to change at this stage.

We thus have changed OWrite's font matching algorithm to simply ignore the dot in font names so that all fonts and their type faces can be matched more accurately across the two platforms.

Summary:

- OWrite now stores and maps TT-face-names in OWrite documents. Existing documents should continue to work.
- OWrite now ignores '.' separators when mapping font names.

**ID :** 1301

**Fixed in version :** 3.6.2

**Short Description:** Crash when inserting RTF (Regression)

**Full Description:** We just started getting consistent crashing on Windows. I've yet to fully investigate why but it seems to be when inserting RTF into a document using \$insert.

It works fine in oWrite 3.5.0 and it works fine on Mac, it purely goes wrong on Windows.

It is entirely possible this is related to fonts as we are displaying a couple of documents that are authored on a Mac and then displayed on Windows at startup.

I'll try and see if I can isolate it and send you a sample library. For now we'll keep using 3.5.0 on Windows.

**Comments :** This crash was related to a document style that had a paragraph list set, i.e. a bullet, numbered or alpha list. The RTF import was deleting an invalid pointer when a paragraph list style was imported as part of a document style. This was not causing an immediate crash, but we had to open and close the sample window between 10 and 20 times before the corrupted heap would cause an exception somewhere in Omnis.

**ID :** 1304

**Fixed in version :** 3.6.3

**Short Description:** oWrite doesn't see styles used in tables

**Full Description:** We're looking into another weird issue there which is likely our own fault, but this one is easy to reproduce in your sample tools.

Create a blank oWrite document, add a table, put some text in, and set a style on that text.

Then go to the style gallery and remove unused styles.

It will not see that the style used in the table is used and will remove the style (and then crash).

**Comments :**

**ID :** 1305

**Fixed in version :** 3.6.4

**Short Description:** oWrite crashes when inserting RTF using \$::insert

**Full Description:** If you have a header and footer in an RTF document that you try to insert into another document with \$::insert it either simply fails to insert the text or crashes (the crash log mentions something about styles being evaluated in the RTF). This seems to be happening since 360, the header/footer was being ignored in 350 (which is correct in this case I feel).

You can use \$headfootenabed and set it to false and the header/footer will not be included in the RTF but this seems to only work when you do so on the visual component, not on the non-visual component.

For now I filter out the header and footer manually in the resulting RTF before inserting it and that works just fine.

**Comments :** As was correctly surmised in the report, version 3.6.0 added support for importing headers and footers contained within RTF documents. Unfortunately, OWrite version 3.6.x also attempts to parse headers and footers when parsing RTF during a \$::insert, paste or document evaluation when inserting RTF via a calculated field. This caused problems due to the invalid scope.

The work-around is to turn off header and footer support during these operations by assigning kFalse to \$headfootenabed. As far as we can tell this works with both, visual and non-visual components. We cannot confirm that this work-around does not work with non-visual components as the report suggests.

Version 3.6.4 ensures that headers and footers are only parsed when importing entire documents during a call to \$loaddata().

**ID :** 1307

**Fixed in version :** 3.6.4

**Short Description:** Using custom notation with \$dataname fails

**Full Description:** When using custom notation such as MyObj.\$owritedata it does not work with \$dataname unless \$dataname type expects plain text.

**Comments :**