



You are receiving this Newsletter because you expressed an interest in our software or are currently licensing one or more of our software components. This newsletter provides up-to-date information on the progress of our current developments, new software releases, and details about future developments.

Please also visit our news page for a summary of developments over the past twelve months.

<https://home.brainydata.com/news.htm>

NOTE: NEW SUBSCRIBERS will automatically receive a copy of the last newsletter that we circulated and consequently the date shown at the top of the newsletter may not be current.

Introduction

It has been a while ...

... since our last Newsletter. We had some unexpected important projects to contend with that came our way in the new year, that included Acroforms, the partial rewrite of our export code to HTML and various other important changes as well as additional research for various other projects. More about this in the coming sections. Here is a summary of what is in this Newsletter...

- Our long-term plans regarding support for Acroforms and PDF file merging and signing
- Our short-term plans for oWrite/jsoWrite forms in lieu of Acroforms
- Details about our jsoWrite/oWrite version 5 beta features and changes
- Studio 10.2 ports (all platforms) and Universal binaries (macOS)
- Fully notarized builds in support of macOS Catalina and Big Sur are now available as patch releases
- PDFWriter 6.5 Feature & Maintenance Release
- Updated news about our jsoGantt port
- Technical hint and new examples about using oWrite calculated fields

PDFDevice News

Acroforms (development temporarily halted)

As mentioned in the introduction, the delay in our planned development cycle was due to some unexpected projects, one of which was the Acroform support for PDFDevice. Acroforms are made up of interactive form fields in a PDF document that are typically linked to an action such as an URL that sends the entered data to an HTTP server. We had a number of requests for this feature and thus believing it to be important, embarked on its immediate development. Sadly, despite investing a larger amount of time than we had intended, we have not been able to make it work at this time.

Consequently, with great disappointment, we have put the development of this feature on hold. However, we will continue to investigate its possibility once we have sufficiently progressed with some of our other current development commitments. We are sorry if we have raised some false hopes, but this is one of the uncertainties of software development we face due to the complexity of the technology. On a more positive note, we will add interactive forms to oWrite which may provide a suitable alternative. Please see the next section.

PDF File merging and signing

We are currently investigating options for adding the ability to merge and sign PDF files. Our focus is primarily on merging with signing being an option if possible. This feature is already available in our PDFWriter MS Windows printer driver solution. It would be nice to have a more integrated cross platform solution.

 **oWrite/jsWrite forms (new alternative Acroform feature)**

Although we have temporarily suspended the development of Acroform support for PDFDevice, we nevertheless have been working on adding something along the lines of Acroforms to oWrite and jsWrite. These new oWrite forms will work like Acroforms, but with the added benefit that end users will be able to design forms as well as fill in and submit form data in both oWrite desktop and web-client implementations. Together with oWrite's ability to export these forms to standard HTML, it should provide a useful feature for developers and clients alike to easily design their own HTML forms without having to write any HTML. However, this new feature will spawn a new major release cycle (version 6.0 alpha 1) as we have closed the current version 5 alpha cycle. See the next section for further details about the version 5 beta release.

Version 6 alpha 1 will focus on the desktop version and will be released as soon as the work is completed. The jsWrite version to implement forms will be released soon thereafter as alpha 2.

 **oWrite/jsWrite version 5 beta release**

As well as working on other enhancements, most of our work since alpha 4 has been centred around issues that had been reported, which were considered high risk and therefore required a major release. Below are descriptions of all the more major changes we have worked on since alpha 4. **As these are considered major changes, the affected areas of oWrite will require careful testing within your implementations.**

HTML Export (Case 1887)

There were problems with some very old code that was responsible for interpreting and exporting HTML for floating objects such as pictures and text boxes. This code had seen many tweaks over the years and the decision was taken that the current approach was no longer sustainable. Consequently this area of the code has been entirely re-written and thus **requires careful testing**. All the supported text wrapping modes, wrap, split, behind and in front, should now produce comparable documents in HTML. There were some major issues when a single paragraph contained multiple floating objects that affected all of these modes in different ways. These have been resolved as far as we can tell. Of course, HTML manages documents somewhat differently to a word-processor, so some visual differences will remain between documents viewed in oWrite and documents viewed within browsers.

RTF import & code pages (Case 1879)

We have added support for font based code page translations. Traditionally, oWrite only allowed a single document wide code page when importing RTF. We have changed the oWrite RTF import code so that it now respects font based code pages for RTF documents that contain multiple languages. This is a **major functional change** when importing RTF and **requires complete testing of** your RTF imports. Please note that oWrite uses RTF unicode during export instead of code pages in the support of multi-language documents. Consequently, these changes should, theoretically, not affect documents that were exported to RTF from oWrite. However, we recommend that you do test the importing of such documents, just in case.

RTF Import & type faces (Case 1870)

When oWrite desktop is loaded, it caches information about all installed fonts which includes the various naming conventions. However, how this list was built was not always conducive for the RTF import code to match full font names that combined family and typeface names using the Windows standard of combining these. Traditionally, oWrite focused more on the postscript standard for font mapping during import and export, which appears to work better on Macintosh.

To resolve this issue we had to partially re-write our internal font caching code and the way font names are cached. These changes may impact all font mapping code, including the font map feature \$loadfontmap().

jsoWrite: Spell check as you type

We have added php based spell checking to jsoWrite. There is a new property called \$spellcheckerurl which is a property of the control and specifies the url of the PHP script on the server. The url may provide additional parameters for the PHP script, such as spell checker initialisation options, i.e. language. We provide a basic PHP script that you can modify for your own requirements. The script's url must provide a number of parameters for additional functionality such as adding words to a server dictionary, or spell checking in a language other than US/UK English. Below is a list of all the parameters supported by the provided script and jsoWrite kernel:

lang	used by server script - the two/four letter language code (default is 'en')
spelling	used by server script - the language variation
action	used by server script - one of 'suggest', 'add_personal', 'add_replace' (default is 'suggest')
openex	used by server script - if specified script will open additional personal and replace dictionaries in addition to the language dictionaries
IgnoreWordsWithNumbers	used by kernel - ignore words that contain numbers if true (defaults to true)
IgnoreDomainNames	used by kernel - ignore domain names if true (defaults to true)
IgnoreAllCapsWords	used by kernel - ignore words in all-caps if true (defaults to true)
ReportDoubledWords	used by kernel - reports duplicate words if true (defaults to true)

Our demo \$spellcheckerurl is set to:

```
https:\\demos.brainydata.com\\software\\spell.php?lang="en"&openex="pr"
```

There are many ways to implement the server side. The only actions jsoWrite will perform are the 'suggest' action and the 'add_personal' action if a string is specified for the new string resource 7115. This new feature introduces three new translatable strings that are loaded by jsoWrite. These are:

```
"7113":["Ignore"],  
"7114":["Ignore All"],  
"7115":["Add"]
```

The translatable strings are found in the file JSOWriteStrings.tsv which is located alongside the Document Manager examples library. If you set string 7115 to empty, jsoWrite will not offer this option in the spell checker menu.

The new server folder in the examples tree contains the example php script *spell.php*.

oWrite: \$evalclickcalc (case 1886)

In old versions of oWrite (pre version 3), calculated fields only allowed for a single calculation which could either be used for calculating the display or for a calculation that is executed when the field is clicked (\$curobjclicks is true). In version 3, we introduced \$curobjclickcalc which, alongside \$curobjcalc, allows a calculated field to have a calculation for the display data as well as a separate calculation to be executed when the field is clicked. However, pre version 3 documents may not sit well alongside post version 3 documents that utilised both types of calculations, requiring the separate handling of the two types of documents in some circumstances. To help with this inconsistency, we have introduced this new option. Although this behaviour only kicks in when enabled and thus it does not consist of a major behavioural change for everyone, we nevertheless recognised the logistical complexity of calculated fields and want to highlight some useful standards and possibilities when using calculated fields and pictures. To this end we have dedicated the "Technical Hint" section in this newsletter to the use of the interrelated properties \$curobjcalc, \$curobjclickcalc, \$curobjdisplay, \$curobjclicks, \$evalclickcalc and \$evalcalcs.

For further details about the use of \$evalclickcalc, please also refer to the beta release notes.

Other enhancements

The following are the more noteworthy minor enhancements and changes since alpha 4, full details of which are available in the beta release notes.

Simple Find & Replace: It is now possible to use the find and replace functions specifying plain text instead of having to use the find & replace objects.

Transparent text boxes: The background of text boxes can now be transparent.

Release Notes:

This completes the list of major changes since the oWrite alpha 4 release. For a full list of changes see the beta release notes at

https://supportpublic.brainydata.com/rn/owrite_jsowrite_4910.pdf.

Downloads:

macOS: https://support.brainydata.com/owrite_sm/patches/owrite_4910_major_release5b1_mac.zip
https://support.brainydata.com/jsowrite/patches/jsowrite_4910_major_release5b1_mac.zip
winOS: https://support.brainydata.com/owrite_sw/patches/owrite_4910_major_release5b1_win.zip
https://support.brainydata.com/jsowrite/patches/jsowrite_4910_major_release5b1_win.zip
additional files: https://demos.brainydata.com/software/owrite_jsowrite_4910_additional_files.zip



Studio 10.2 ports

Outstanding 10.2 Ports

Due to a lack of demand, we have made the decision that we will not be porting oLDAP to Studio 10.2 unless it is specifically requested. If you do intend to use this component with Studio 10.2 in the near future, please let us know.

Universal Binary ports (macOS)

These ports have been completed and are available as beta releases to Developers with a technical support subscription to be used with the current Studio beta release. The downloads for the universal beta builds have been combined with the Notarization downloads. Please see download links in the section 'Notarization on macOS'. Once The Studio universal binary has been released post beta, our universal builds will be made available to all developers with a valid maintenance subscription.



Notarization on macOS

We have now fully notarized all our macOS builds for Studio 8.1.7.2 or later. This means that it will now be possible to simply drop our components into existing Studio installations on Catalina or Big Sur systems. These builds have been made available to developers with a valid technical support subscription and can be downloaded for testing from the following locations:

oWrite Desktop / jsowrite Server: \$version xx.4911

https://support.brainydata.com/owrite_sm/patches/owrite_4911_major_release5b1patch1_mac.zip

https://support.brainydata.com/jsowrite/patches/jsowrite_4911_major_release5b1patch1_mac.zip

pdfDevice Desktop: \$version = xx.4001

https://support.brainydata.com/pdfdevice_sm/patches/pdfdevice_4001_major_patch_mac.zip

oSpell2 Desktop: \$version = xx.3301

https://support.brainydata.com/ospell2_sm/patches/ospell2_3301_major_patch_mac.zip

oCal Desktop: \$version = xx.2121

https://support.brainydata.com/ocal_sm/patches/ocal_2121_major_patch_mac.zip

oGantt: \$version = xx.4205

https://support.brainydata.com/ogantt_sm/patches/ogantt_4205_major_patch_mac.zip

jsSignature: \$version = xx.112

https://support.brainydata.com/jssignature/patches/jssignature_v112_major_patch.zip



Other Feature & Maintenance Release

PDFWriter v6.5

We have updated our PDFWriter product with the latest AMYUNI printer drivers. Please refer to the documentation in the downloads for a list of enhancements.

https://support.brainydata.com/pdfwriter_pw/pdfwriter_6503.zip

You will require new activation codes which we will send to you as soon as we have received them. Some of you have special activation codes that require a manual request with the supplier and consequently may take a little longer to be issued.



jsoGantt development

Having completed our oWrite/jsoWrite alpha cycle, we will resume with our jsoGantt port in due course. We suggested in our last newsletter that a beta may be available by late Q2 or early Q3. This now looks unlikely. We are now hoping to make late Q3 early Q4 for an early alpha or first beta.

If you wish to find out more, please contact us via our contacts page at <https://home.brainydata.com/contact.htm>.



Technical Hints

The 'Technical Hint' is a regular article containing one or more interesting things that may have come up in technical support queries or that we encountered during development.

New oWrite 'not-so-basic' basic examples

Development of the Document Manager began during the initial development of oWrite and has grown over the years to a complex library that endeavours to demonstrate most of the features supported by oWrite. Although the examples are comprehensive, due to their sheer complexity and history, the examples may not demonstrate a coherent best-method approach. It became apparent some time ago that there is a need for a new basic set of example libraries that focus on narrower aspects of oWrite so that the features in focus are not lost amongst the clatter of history and complexity.

We have completed our first in a line of new basic example libraries. This first library will focus on basic editing, data merging and the output of oWrite documents to all supported output formats. Although this library is designed around a

desktop implementation, the main focus of attention is in fact a set of object classes that can equally be used in desktop or JS server implementations. This first library in the series we call *OWriteBasicMergePrintExp.lbs*.

Basic merge, print and export classes

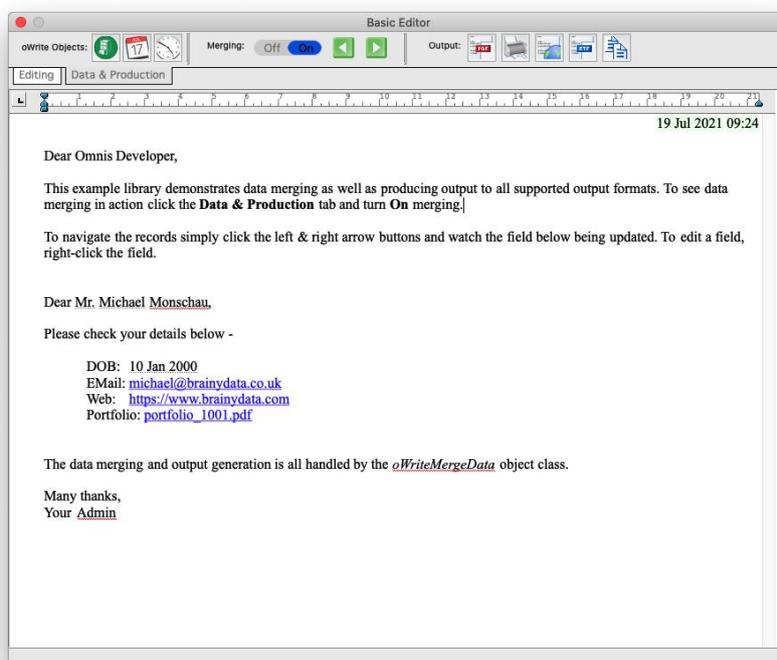
The example library contains two objects, *oWriteMergeData* and *oWriteOutputDoc*. These two objects do all the work in regards to merging, printing and exporting oWrite documents. The object *oWriteMergeData* inherits the non-visual external object *.OWrite.OWriteDoc* and *oWriteOutputDoc* inherits *oWriteMergeData*. This arrangement of inheritance allows *oWriteMergeData* to load and merge documents even when no visual editing is taking place and it allows *oWriteOutputDoc* to merge data prior to printing or exporting documents. A simple interface for editing a document is provided by the classes *wOWriteBasic* and *wOWriteBasicTools*. The class *wEditCalcDBText* implements the main interface for editing an oWrite calculated field and the remaining oSpell and PDFDevice classes are generic classes taken from the Document Manager examples to implement PDF production and spell checking.

Compartmentalising the various functions of oWrite document handling in this way allows the effective separation of interface and function, while allowing the data to drive various interface options. For example, the *oWriteMergeData* object class provides the pick lists for picking database table names, column names and column types when inserting and editing calculated fields. Using this information the window class *wEditCalcDBText* presents the user with appropriate choices and creates the appropriate Omnis calculations for data display and interaction.

We hope that this more simple set of example classes provides an easier entry into utilising the power of our oWrite plus suite of components. We will, in due course, produce a more detailed step-by-step set of instructions as technical notes that will accompany these examples.

The new example library and support files can be downloaded from

https://demos.brainydata.com/software/owrite_basic_merge_print_exp.zip





We want your Feedback!

We would like to add your feedback to our customer page at <https://home.brainydata.com/customers.htm>. Please take a look and if you feel you would like to contribute to this page please send us an email with your feedback about our services and products. You are also very welcome to add details about your own products! **Even if you have already provided feedback previously**, perhaps it is time to update your feedback or add some additional comments?

There is no limit to the size of your feedback, but in addition to your comments we would like to list your company name with a link to your website, your name (although required for us to identify you, you can request for your name not to be shown on our feedback page) and your position within the company (i.e. job title).

Please send all feedback to feedback@brainydata.co.uk (do not forget to include your name, job title and website address). We appreciate your support.



Important Links

News: <https://home.brainydata.com/news.htm>

Products & pricing: <https://products.brainydata.com>

Demo/Examples Downloads: <https://demos.brainydata.com/download.htm>

GitHub: <https://github.com/BrainyData>

Sponsors: <https://home.brainydata.com/sponsors.htm>

Feedback: <https://home.brainydata.com/customers.htm>

Online Documentation: <https://supportpublic.brainydata.com/documentation.htm>

Technical notes: <https://supportpublic.brainydata.com/technotes.htm>

Support Request Form: visit <https://supportpublic.brainydata.com> and click “Software Downloads”

Software Downloads: visit <https://supportpublic.brainydata.com> and click “Contact Support”

This newsletter is for informational purposes only. Brainy Data assumes no responsibility for its accuracy, and the information is subject to change without notice. Any use of, or actions taken based upon, any of the information contained in this newsletter is done entirely at your own risk.

Copyright (c) 2021 Brainy Data Limited

This document was produced by OWrite and PDFDevice.