



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.

## Studio 10 ports

With the exception of OLDAP, we have updated all our projects and rebuilt all our current release software with the latest Studio 10 SDK. These builds are available for download at the locations below

JS-Signature: [https://support.brainydata.com/jssignature\\_su.htm](https://support.brainydata.com/jssignature_su.htm)

JS-OWrite: [https://support.brainydata.com/jsowrite\\_su.htm](https://support.brainydata.com/jsowrite_su.htm)

OWrite: [https://support.brainydata.com/owrite\\_su.htm](https://support.brainydata.com/owrite_su.htm)

PDFDevice: [https://support.brainydata.com/pdfdevice\\_su.htm](https://support.brainydata.com/pdfdevice_su.htm)

OSpell2: [https://support.brainydata.com/ospell2\\_su.htm](https://support.brainydata.com/ospell2_su.htm)

OGantt: [https://support.brainydata.com/ogantt\\_su.htm](https://support.brainydata.com/ogantt_su.htm)

OCal: [https://support.brainydata.com/ocal\\_su.htm](https://support.brainydata.com/ocal_su.htm)

We will release OLDAP for Studio 10 in due course.

## Software updates

OWrite and JS-OWrite version 4.2.0.0 are available for download at the locations below.

OWrite Desktop License: [https://support.brainydata.com/owrite\\_su.htm](https://support.brainydata.com/owrite_su.htm)

OWrite JS-Server License: [https://support.brainydata.com/jsowrite\\_su.htm](https://support.brainydata.com/jsowrite_su.htm)

OWrite Desktop/Server: We have resolved some further regressions that were caused by our move to floating point measurements and resolved some cosmetic issues caused by the Cocoa port. This build also includes some enhancements to the \$picturefrompage method, which is accompanied by a new JS-OWrite example for building special page headers (see TN0026).

JS-OWrite Client: we have added support for the field view display mode, made some improvements to how text is copied, fixed a regression related to bullet/numbered lists and added external source support for images (see TN0026).

OGantt version 4.1.0 is available for download from the page below. This release fixes an intermittent crash and some printing issues

[https://support.brainydata.com/ogantt\\_su.htm](https://support.brainydata.com/ogantt_su.htm)

PDFWriter version 6.0.1.1 is available for download from the location below. As well as full Windows 10 certification and PDF v1.7 support, this version implements new locking mechanisms that are required to run multiple processes of the same Omnis executable (see TN0028).

[https://support.brainydata.com/pdfwriter\\_su.htm](https://support.brainydata.com/pdfwriter_su.htm)

IMPORTANT NOTE: you will require a new activation code for this version. Please contact our support department via the support request form to request your new activation code and allow at least 48 hours for the new code to arrive.

[https://support.brainydata.com/request\\_form.htm](https://support.brainydata.com/request_form.htm)



## New Technical Notes

We have uploaded some new technical notes that accompany the latest software releases.

TN0026: OWrite/JSOWrite - Creating Special Page Headers

TN0027: JSOWrite - Inserting Objects

TN0028: PDFWriter - New Locking Feature

TN0029: PDFDevice - Missing report destination icon

All technical notes can be viewed at

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



## Ongoing development - JS-Client Ports

By now, we hoped we would have progressed further down the long road of porting our components to Javascript. Unfortunately, the JS-OWrite port not only proved unexpectedly more difficult and time-consuming, but also subsequent resources required to support developers implementing JS-OWrite has far exceeded our expectations. In addition, OWrite had to undergo some major changes to accommodate this new platform, creating further uncertainty for existing functionality which had inadvertently been affected by these substantial changes. Considering these uncertainties was only possible in hindsight. Consequently, our ability to predict completion of future projects, such as JS-OCal, had been compromised in the extreme. We had already touched upon our difficulties with progressing our JS-OCal port in our December Newsletter, but we would like to stress at this point that we will do our best to progress with the port as best as we can. Although we cannot make any firm commitment as to a definitive time-frame at this moment, we will endeavour to keep you up to date on our progress.



## Technical Hint - JS-OWrite Asynchronicity

Many operations carried out by JS-OWrite on the client, such as saving data, are asynchronous operations. This means that when one calls `$savedata()`, for example, any code that follows will be executed prior to the data having been saved.

To help with the asynchronous nature of these operations, many of the client JS-OWrite methods can generate an `evAsyncDone` event, when the operation has been completed. The methods `$savedata()`, `$loaddata()`, `$loadautosave()`, `$search()`, `$insert()`, `$tableaction()` and `$setselection()` can all be called with additional parameters specifying your personal async ID and optional async data. When the action triggered by the JS-OWrite method is completed the `evAsyncDone` event will be generated passing to it the ID and data that you provided.

For example: saving data for a print request has to be carried out in two distinct asynchronous operations. Calling `$savedata("Print","PDF")` will generate the `evAsyncDone` event once the data has been saved. The event parameters `pAsyncId` and `pAsyncData` will contain the parameters you provided. Once you receive the event you can print the saved data to PDF. Additional parameters that are passed by JS-OWrite to the event are `pAsyncFunction`, `pAsyncSuccess`, `pAsyncResultData`. Please refer to the JS-OWrite documentation for a detailed description of the `evAsyncDone` event and its parameters.

---

## 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 document was produced by OWrite and PDFDevice.

---

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) 2019 Brainy Data Limited

---