

# Brainy Data Support Services & Policies

## Contents

- 1. Introduction .....2
- 2. Design of our support subscriptions.....2
- 3. Benefits of our support subscriptions.....2
- 4. Why do we charge for maintenance? .....3
- 5. How do we charge for DMS and DSS?.....4
- 6. Why do we charge a support fee for each technical contact? .....4
- 7. How do we cater for developers without a DMS?.....5
- 8. Why use a support request form?.....5
- 9. What information should be provided in the description field?.....5
- 10. Why do you have to provide example code? .....6

## 1. Introduction

This document details our policies regarding our *Deployment Maintenance Subscription* (DMS) and *Developer Support Subscription* (DSS). It will answer questions such as: what benefits are included in our support subscriptions, why and how we charge for maintenance and support; why we insist on the use of our [support request form](#); how the request form should be used; and why we insist that bug reports are accompanied by working examples.

We hope that this document clarifies our policy positions and addresses your questions. If you have further queries or comments regarding our policies don't hesitate to [contact](#) us.

## 2. Design of our support subscriptions

Our DMS and DSS products are designed around a simplified licensing model that does not distinguish between runtime and developer licenses for the purpose of distribution and client installations, but instead distinguishes between *deployment-maintenance* and *developer-support*. We chose this design because we wanted our Development Life Cycle to be more responsive. This responsiveness requires us to release many new enhancements within minor version releases for which the traditional *developer-deployment-licensing*<sup>1</sup> and *maintenance-support-subscription* models do not work. Please see our Development Life Cycle document at <https://supportpublic.brainydata.com/pd/pd0002.pdf> for more information regarding the detailed implementation of our responsive licensing.

We feel the *deployment-maintenance* and *developer-support* corollary model provides a fairer system of paying for the services that you require. The DMS cost is typically<sup>2</sup> based on the size of your deployment base and ensures your deployment base remains up to date with the latest improvements and corrections. The DSS is based on the number of active developers who will require access to our developer support services, not the number of actual developers in your company who may or may not develop our plugin software. By not enforcing individual costly developer licenses that are traditionally required for every developer machine, we believe we are treating our customers more fairly while at the same time insuring our continued survival in this small developer market through a fairer employment based pricing model.

## 3. Benefits of our support subscriptions

Please also read section 2 above to find out why we designed our subscription products in this way.

The DMS is designed to provide peace of mind for your entire deployment base. It entitles you to download and distribute all new versions of the insured software according to the limitations of your deployment license that was initially purchased. This is the limit of the benefits of the DMS.

In the absence of developer licenses, the DSS is designed to provide developer support benefits for developers who require access to our support team for the purpose of reporting software issues,

---

<sup>1</sup> Sometime also referred to as developer and runtime licensing.

<sup>2</sup> Not all products offer banded deployment licenses. A few of our products offer only unlimited or product based deployment licenses.

access to patch and Studio Now releases, request enhancements and technical support related to development and use. Please note that support is only provided for the products for which you also have an active DMS.

**Studio Now patch releases:** Access to Studio Now releases requires a Developer Support Subscription (DSS), just as Studio now itself requires you to subscribe to the Omnis Developer Partner (ODP) program with Omnis Software. At the time of writing this, the public Studio release for everyone is version 11.1 and we have also made available Studio 11.1 releases for everyone with an active DMS. The current Studio Now releases are 11.2 builds which are equivalent to 11.2 alpha or beta releases. These releases are much more frequent (typically one a month) and include both new features and maintenance changes for which we are obliged to provide builds for our components if there are potential compatibility issues. Supporting this level of compatibility requires more effort on our behalf. Once Omnis releases 11.2 for everyone we will also make available our final 11.2 builds for everyone with an active DMS. In the meantime, a DSS is required.

#### **4. Why do we charge for maintenance?**

Our external components provide highly flexible programable interfaces for adding important functionality to Omnis applications. The flexibility of our external components allows developers to use them in diverse software markets and consequently our products are implemented in diverse ways. All our external components are provided with working examples that demonstrate their features and their intended use. However, we cannot anticipate all possible uses and developers will often find themselves operating outside the scope of our example applications and sometimes our external components may not perform as expected.

Some may feel these are software errors that should be corrected free of charge. However, the reality is that they are the result of technical uncertainties through complexity that is exacerbated through the programable nature of our software and for this we cannot provide a guaranty. Often, the circumstances surrounding such issues are unique to the application. In many cases, when issues arise for which we alter our components we do it to enable developer's to use them in a preferred way, not necessarily in a way that was part of the original design. These points argue that there is a thin line between, what is traditionally perceived as, error corrections and enhancements, thus rendering this distinction invalid.

Furthermore, our less visible work, such as compatibility with multiple major versions of Omnis Studio as well as the support of different platforms and different versions of operating systems which we cannot limit to major version releases, incurs substantial costs.

All developers who purchase a DMS benefit from this ongoing work, which is partially financed by the DMS fee according to the size of your deployment base (i.e. based on how many users benefit from our product). We feel this model is fairer than charging everyone the same regardless of the size of their distribution base and is also less prohibitive, in terms of cost, for smaller developers.

## **5. How do we charge for DMS and DSS?**

When you purchase a deployment license for one of our products, most of our products include a twelve months DMS free of charge. After this initial period we charge an annual fee based on a percentage value of the deployment license fee (typically 30% based on current prices). The DMS is an optional subscription for many of our products and alternative upgrade paths are available within 24 months of your last DMS expiring (50% discount within 12 months, 25% discount within 24 month). If your DMS expired more than 2 years earlier, a new deployment license must be purchased at full list price. Please refer to our [licensing document](#) for further details regarding the full range of deployment options.

The DSS which provides developer support services is charged according to the number of registered developers (only registered developers can contact our support department). There are three tears of subscription:

SOLO (1 registered developer)

TEAM (up to 3 registered developers)

PREMIER (up to 7 registered developers)

The TEAM and PREMIER subscriptions include the additional benefit of 25% and 50% discounts on consultancy and new deployment licenses (excludes upgrades of existing deployment licenses).

Please also read sections 2 and 3 above to find out why we designed our subscription products in this way and the limitations and benefits of each subscription.

Our maintenance and support structure is not much different to that of other software companies. It is common practice for companies to release regular major software upgrades for which upgrade fees apply. If a developer requires software corrections that have been applied in a new major version, the developer is subject to upgrade fees unless a paid subscription was entered into that includes free upgrades. As already detailed in sections 2 and 3, at Brainy Data we merely differ in that we try to be more responsive to developer's needs and include, besides software corrections, new feature enhancements in minor as well as major version releases. Thus the typical upgrade model of only charging for major versions does not fit our model. This reality also makes it difficult for us to charge for developer licenses to cover the cost of additional developer benefits through traditional major version paid upgrades. Instead we opted for the DMS-DSS model, which cleanly separates deployment updates and upgrades from developer services.

## **6. Why do we charge a support fee for each technical contact?**

We find that companies with multiple developers working on applications involving our plugin software, often generate a greater number of support requests than is normally the case.

Furthermore, communicating with multiple individuals results in additional work such as duplicate inauguration and technical queries, which places additional strain on our resources. We therefore charge a support fee per registered contact.

Finally, charging a fee per registered contact partially compensates us for not insisting on developer licenses per developer who installs our software (see section 2).

## **7. How do we cater for developers without a DMS?**

Our demo download pages and public support pages provide free of charge the latest demo software, product documentation, technical notes, release notes and examples. Our free material creates an opportunity for developers who chose not to continue their DMS to try out and read up about the latest changes and enhancements prior to paying for an upgrade.

Though we are happy to answer some pre-sales questions such as “can the software do this or that?” or “what is included in the license fee?”, we are unable to answer technical questions such as “how does this work?” or “how do we do that?” or “I have this problem, can this be fixed in the next version?”. Such queries and requests come under the remit of developer support and require a DSS without which we cannot enter into technical dialogue as it takes away valuable resources from developers who have paid for this level of support.

## **8. Why use a support request form?**

In order to deal effectively and efficiently with all support requests we require accurate detailed information for each case. As well as ensuring that all vital information is provided, the support request form is linked to our automated case tracking system which increases our efficiency further. However, our system will only work if all developers use our form. Consequently we will insist on the form being used as we are no longer able to effectively track support cases without using the online form. Queries initiated via email are not compatible with our system. However, after a new case is entered in our system, subsequent dialogue will take place via direct email.

## **9. What information should be provided in the description field?**

For us to understand and resolve an issue efficiently, we require appropriate detailed descriptions. If the description is not written correctly we may miss-interpret or fail to reproduce the problem. One successful approach is to break down the description as follows:

- Background info: this can be anything that may be relevant, i.e. what you are trying to do and why you are doing it this way. The latter may help us interpret the former correctly.
- Steps: the detailed steps to reproduce the problem using our example library or a modified example library which should be attached to the form using the “Upload File” box.
- Expected result: a description of what you expected to happen.
- Actual result: a description of what actually happened.

Please always provide all relevant details in this field. Do not refer to emails or attached documents as an alternative to a description. Descriptions such as “details will follow in an email” or “details are attached” are not compatible with our system.

## 10. Why do you have to provide example code?

When it comes to supporting plug-ins for development tools such as Omnis, there is a very thin line between what should be considered consultancy work and what can be covered by the DSS.

When developers encounter a problem, the first action should be to see if it can be replicated in the examples which will have two outcomes:

- a. The problem can be replicated in which case it is a simple task to provide us with step-by-step instructions when reporting the problem.
- b. The problem cannot be easily replicated so it is probably something intricate in the developer's library that is causing the problem to surface.

In such a case, we cannot accept a video or screen shoot instead of a working example, as such media merely displays the symptoms of the problem but does not demonstrate the cause.

Furthermore, we cannot accept a developer's applications as this would require us to install, learn and debug an unfamiliar application. This may not be an issue if the library is very basic, but in our experience developer's libraries are rarely that. Thus, installing and debugging a developer's own library is not sustainable within the remit of our support program.

We recommend that the circumstances are identified and re-created within our own examples. This approach has the benefit that it filters out cases of incorrect coding. If this approach proves too difficult, we may consider installing and debugging a developer's application, but such work will be subject to consultancy fees.

## Document History

- 16 Jan 2025: added Studio Now
- 21 Oct 2022b: general corrections and improved phrasing
- 21 Oct 2022a: clarified deployment-maintenance and developer-support subscriptions
- 01 Jan 2017: changes to maintenance and support subscription benefits
- 13 May 2015: grammatical corrections and numbered sections
- 12 Dec 2014: further changes to various descriptions
- 04 Dec 2014: divergence of maintenance and technical support programs
- 09 May 2013: first publication