



# InDetail

InDetail Paper by Bloor  
Author **Philip Howard**  
Publish date **November 2016**

---

**EASA**



**The problem for EASA is that it is in a class of one. With no other comparable products in the market it is difficult to get the word out as to how valuable the technology is. It is our hope that this paper helps to redress that balance.**



Author **Philip Howard**

# Executive summary

It is slightly difficult to characterise EASA – mostly because it’s rather unique, and there are no directly comparable products with which one can make an obvious comparison. The company (also called EASA) describes the product as a “codeless rapid application development platform”, emphasising “codeless” by referring to those who create apps as “authors” rather than “developers”. In fact, authors are as likely to be business users as IT people. Moreover, when you “author” an app, you are often not so much authoring an application from scratch; rather, you are usually creating a user interface that drives an existing tool, for example an Excel spreadsheet or a home-grown script. The result is an application with an interface that is specific to the users’ needs, that is accessible to any authorised user on any device (including mobile). Access management, version control, and audit trails are all provided, while securing the intellectual property in the underlying spreadsheet or script is a fundamental part of EASA’s proposition.

Why would you want to do this?

There are several reasons: you might need to deploy an application (R code or Python script, say) which doesn’t yet have a graphical user interface to a wider user audience; you might simply want a browser-based interface to an application that is only supported on a desktop; or you might have a scenario where you need to ensure that sales staff are using only the current and most up-to-date version of a spreadsheet used for configuration, pricing and quoting (CPQ), while imposing security, governance and compliance around the intellectual property embedded in the spreadsheet, and while capturing the data from every quote issued by every sales person. Or you might have a mathematical, scientific or technical model with similar characteristics. Or, of course, you might have some combination of these requirements.

## Fast facts

Using EASA you can create custom web applications, known as EASAPs, which connect to underlying spreadsheets, scripts, and/or databases. A major point to note about an EASAP is that the underlying tool (Excel, MATLAB and so on) or code that defines your application no longer needs to be shared. What is shared is the (web) application that interfaces to that application. This means, firstly, that the end-users do not have to have Excel or MATLAB running on whatever device they are using. This in turn means that EASAPs are easily deployable on mobile devices as well as desktops. It also means that the application interface does not need to be based on the technology underpinning the application: you may be interacting, ultimately, with a spreadsheet, but your interface doesn’t have to look like a spreadsheet, unless you want it to. Finally, and perhaps most importantly, this approach means that you can change the logic in your underlying application without having to change the user interface. In the case of a spreadsheet application, for example, this means that you can ensure that users are always employing the latest, most current, version of that spreadsheet.

## Key findings

In the opinion of Bloor Research, the following represent the key facts of which prospective users should be aware:

- The process of building an EASAP, whether from scratch or for encapsulating an existing model, is entirely codeless. This uses patented technology.
- Once an EASAP is published, authorised users access it through a web browser.
- EASA applications can front-end a variety of applications, whether these are based on Microsoft Excel, MATLAB, Mathcad, SAS and others, or are simply conventionally coded applications.
- You can integrate data from outside your application – via web services or JDBC (SQL and NoSQL databases



**...perhaps most importantly, this approach means that you can change the logic in your underlying application without having to change the user interface.**





**We are not aware of any other company with a solution like EASA. From a legacy modernisation perspective, the logical separation of the interface from the core application is distinctive.**



and application environments such as CRM) – into your EASA application, thereby enriching its use.

- Cloud-based options, both private and public, are available.
- Version control is implicit both with respect to the models themselves and in terms of different versions of Excel that may be in use.
- From a security perspective the product supports SSL (secure sockets layer) as well as authentication that leverages products such as Active Directory and LDAP.
- EASA is complementary to spreadsheet governance solutions and it partners with, for example, Cluster Seven.

#### **The bottom line**

We are not aware of any other company with a solution like EASA. From a legacy modernisation perspective, the logical separation of the interface from the core application is distinctive.

With respect to spreadsheet deployment – which represents the majority of EASA's use-cases – of course there are spreadsheet governance tools which, at least in some cases, have overlapping capabilities. However, there is a significant difference. As mentioned in the **Fast Facts** above, with EASA you are not sharing the spreadsheets themselves: you are providing an interface to a spreadsheet instance running on a secure server. Spreadsheet governance tools may centrally control spreadsheets but, essentially, still share the spreadsheet. This can have a number of consequences. For instance, it means that the interface is limited to what the spreadsheet itself looks like; it can mean that the intellectual property embedded in the spreadsheet is not secured; and it usually means that end-users have to have Excel installed on their devices.

Alternative spreadsheet deployment tools such as SharePoint and Google Docs do in fact remove the need for a local version of Excel, and can help with version control. However, because the original spreadsheet is translated into a web app, rather than connected to a “real” Excel instance, then a great deal of Excel functionality, such as VBA, data validation, buttons, and so forth, are lost. This is not the case with EASA, which retains macro- and VBA-enabled capabilities.

Of course, EASA is not limited to spreadsheet environments. In particular, there are many modelling tools (HPE's Product Express would be a good example), or statistical environments (perhaps based on R) that do not have a graphical user interface of their own. In these cases, putting an EASAP on the front-end is a fast and effective solution. It is for this reason that EASA sometimes talks about its product as enabling the “democratisation of models” in the sense that it enables widespread deployment and use of both financial and technical models throughout the business. This significantly adds to the value of such models if it means that their (correct) usage becomes more widespread.

# The product

**E**ASA runs on Windows, Linux and VM platforms with each EASA Server capable of handling around 1,000 concurrent users (regardless of the number of applications). In theory, the environment can be deployed either on premises or in the cloud (EASA is partnered with AWS). However, in practice EASA has found little appetite for public cloud deployments, presumably because most customers consider the content of their models and applications too sensitive.

**Figure 1** illustrates the components within an EASA environment. If we discuss this starting on the right-hand side, options for databases include Microsoft SQL Server, Oracle and Hadoop (connected via JDBC), as well as applications such as CRM (customer relationship management) or PLM (product lifecycle management). Data from any of these environments can be included within an EASA application. Excel Servers (EASA terminology for servers which run Excel.exe) are also optional and refer to Windows or VM systems running Excel XP, 2003, 2007, 2010 or 2013. There is an EASA component that enables efficient bi-directional communication between Excel instances and the EASA Server itself.

On the left-hand side of this diagram the distinction that needs to be made is between authors and users (mobile or otherwise). The former are the people who create the EASA applications. They may or may not be the creators of underlying models themselves (if the application is model-based) and they certainly don't need to be developers – because there is no coding involved – which is why EASA refers to them as authors. The latter – users – are the consumers of the apps published by the authors.

There are, in fact, two authoring tools. One is the Excel Range Wizard, which creates EASAPs directly from spreadsheets that you have developed. The second is the EASAP builder, which is used either for non-Excel based apps or for Excel based apps where you want user interface components that don't exist in the underlying spreadsheet. We shall discuss each of these in turn.

## The Excel Range Wizard

The very simplest way to use this wizard is to open the spreadsheet you want to build an EASAP for, highlight the cells of the spreadsheet that you want to expose within the EASAP, and hit the equivalent of the “go” button. Apart from giving your new EASAP a name and a few other housekeeping details, that's it. You can literally deploy spreadsheets within a few minutes in this way. The EASA software captures all the relevant logic (for example, validation logic), identifies inputs and outputs, generates the corresponding user interface, and connects it to the master version of the spreadsheet, which is also uploaded to the EASA Server. There is also a test facility so that you can check that things work appropriately, but other than that you are now ready to publish your EASAP.

“  
You can literally  
deploy spreadsheets  
within a few minutes  
in this way.  
”

**Figure 1 – The components within an EASA environment**



Of course, you will often want to do more than simply emulate the existing spreadsheet interface. While you would use the EASAP Builder if you wanted to create something not based on a spreadsheet, you may also want to filter out particular aspects of the spreadsheet which should not be viewable by end users. Thus there are options (you simply tick relevant boxes in the wizard) if you want to do things such as removing columns and rows, or ignoring cells if they are empty. These options are parameterised so that, for example, ignoring cells may be dependent on whether the cells are formatted or not. There are also options to extract charts as well as the ability make multiple selections from different worksheets.



EASAP Builder is typically used where there is no existing graphical interface at all in your underlying model.



## EASAP Builder

An example of EASAP Builder's interface is shown in **Figure 2**, which is pretty typical of the entire environment. In this case it is showing optional parameters that you can enter to define the layout of a sub-pane, but there are also options for adding items such as customised buttons, database extractions or report generation (which can be to pdf). You can also use EASAP Builder to brand your published EASAPs, incorporating your own company logo and colours, for example.

While the Excel Range Wizard enables you to simply highlight and grab an existing Excel interface and automatically reuse it, with EASAP Builder you define the user interface more explicitly. Thus, EASAP Builder is typically used where there is no existing graphical interface at all in your underlying model: for example, there is only a command line interface.

The main elements that you can define when using the EASAP Builder are:

- The user interface style or “skin”
- Form items – things like buttons, input boxes, images and so on
- Tabs – similarly
- Diagrams – 2D graphs, diagrams per se and the styles applied to these
- Units – what measures are used for currency, weights, temperature and so forth
- Processes – database read or write, send email, run macro and so on
- Output options – reporting, HTML style
- Scalar data processing
- List data processing
- Event processing
- Actions to be taken.

Figure 2 – EASAP Builder's interface

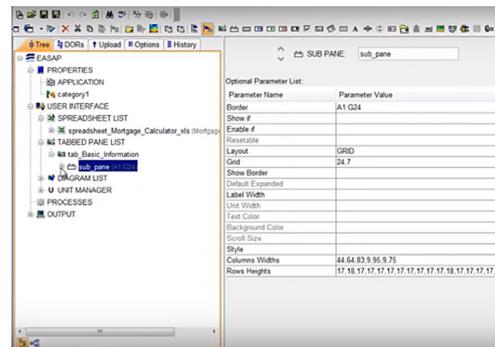
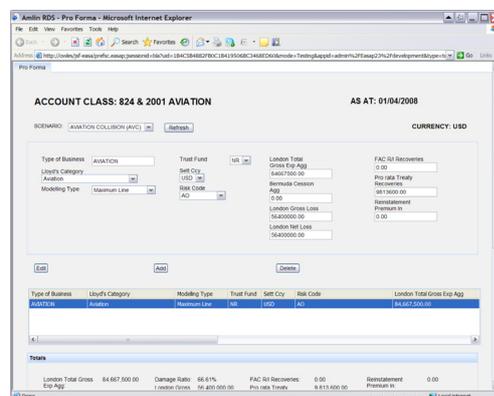


Figure 3 – An example interface authored using EASA



To reiterate: one of the notable things that you can do with EASA is to create user interfaces that do not look like the underlying model. This is especially important if the user is to be able to access the application via a smartphone where the amount of screen real estate is limited, but it can also be important more generally. **Figure 3** shows an example of the sort of interface you can develop using EASA.

Finally, with respect to the technical capabilities of the product, EASA also provides its own Expression Language where an expression is a mathematical or logical statement consisting of some combination of numbers, mathematical operators (including trigonometric, hyperbolic, logarithmic and factorial functions), logical (Boolean) operators, constants and reference to existing object values. A common example of where you might want to use the Expression Language would be where the same EASAP is going to be deployed in multiple regions. While EASA supports multiple languages out of the box, you might also want to perform if/then type of functions and display the same EASAP differently to different user-groups.

## Use cases

**E**ASA is suitable for use in a wide-range of applications. It has, for example, been used to web-enable legacy text-based applications driven from the command line. More generally, however, it addresses a specific business problem. Rather than try to put a label on that issue we shall describe it.

Suppose you have a business model. For example, you are a motor manufacturer and you have a pricing model that reflects all of the different options that are available for each of the different cars that you make. Or the pricing (and profitability) of each sale of your products or services are dependent on many inputs, some of which (for instance, raw material prices) may fluctuate. Or you have an actuarial model. Or perhaps your model is based more on engineering: for example, how best to configure wind turbines for power generation within a wind farm. Or your focus is more scientific: for instance, modelling drag across an aerofoil. Or you are modelling realistic disaster scenarios for insurance purposes. Or, in the same industry, you are modelling your capital reserves in order to ensure Solvency II compliance. Or you are using models to support project management. Or ... well, we could go on, but you get the idea.

There are several key points about scenarios such as these. Firstly, the models themselves are subject to change and evolve over time. Secondly, they are typically reusable across different use cases. Thirdly, they are created by specialists but are used more widely across different areas of the business. Next, they tend to be highly configurable (by the users). Finally, they often have little or inadequate security, and intellectual property is in danger of being compromised because of the lack of appropriate controls. Now we come to the issues involved: how do you ensure that users (as opposed to developers) of these models only use the latest version of the relevant model? How do you

ensure that users only see those parts of the model that are relevant to their business role and how do you ensure that they can only change what they are allowed to change? How do you capture every instance saved by every user, and perhaps enable collaboration between users? These are the points that EASA addresses, so that one way to categorise EASA is as a model deployment tool.

However, there is a further point: the majority of the models for which EASA is used are based on Excel spreadsheets. This leads on to a discussion of spreadsheet governance. While EASA certainly provides security within spreadsheet environments – preventing unauthorised users from seeing, let alone changing, formulae, for example – it is only otherwise a governance tool in the sense that it prevents spreadsheet proliferation by preserving a single version of the truth and ensuring that users can only work with that version. However, it assumes that the models you have built using your spreadsheet are valid and correct, that links remain in place and are not broken, that nobody is doing fraudulent things with your spreadsheets, and it ignores who does what to the underlying model. It is best therefore to view EASA as complementary to spreadsheet governance tools that do these things and, indeed, the company formally partners with vendors in this space.



**How do you capture every instance saved by every user, and perhaps enable collaboration between users? These are the points that EASA addresses, so that one way to categorise EASA is as a model deployment tool.**



## The vendor

Initially, the capabilities that are now offered by EASA were developed by the UK's Atomic Energy Authority (AEA). It had developed various scientific, technical and engineering programmes as part of its business. The issue it faced was that these contained a significant degree of intellectual property that was exposed to users of those applications. Further, as these models and simulations evolved over time, or were applied to new use cases, there was no way of ensuring that users were deploying the latest version of the model. To resolve these issues, the company developed the software that is the subject of this paper.

Recognising the value of what it had developed, AEA Technology spun out EASA Software in the early part of this century as an independent company. Initially, it focused on the technical and engineering models that it had been originally developed for. However, one of its customers – Proctor & Gamble – applied EASA's technology to spreadsheet models that the company used, shared its experience of doing so with EASA and this ultimately led to the development of the Excel Range Wizard.

EASA nowadays has its headquarters in Oxford, with a wholly owned subsidiary in St. Petersburg, Florida. Development is in Oxford, and there is an additional US office providing technical services in Pittsburgh, Pennsylvania. Direct sales are based in both Oxford and St. Petersburg. The company has partners in Japan, China, South Africa, Australia and New Zealand, South Korea and India.

**Website:** [www.easasoftware.com](http://www.easasoftware.com)



**Proctor & Gamble – applied EASA's technology to spreadsheet models that the company used, shared its experience of doing so with EASA and this ultimately led to the development of the Excel Range Wizard.**



## Summary

**E**ASA is a general-purpose tool for building browser-based applications. It is usually employed to support model deployment, of which the most common examples are spreadsheet models. It has a number of benefits, of which the first is that application authoring – which involves no coding – is extremely rapid – usually measured in hours. Even when you are not using a spreadsheet as a starting point, development times are typically measured in weeks rather than months. Secondly, you can make models available via a web browser that enables mobile usage as well as access for users without the underlying modelling tools needing

to be installed locally. But perhaps, most of all, it means that models can be widely deployed within your organisation in a secure and controlled manner. It's ironic that concerns about security and control often mean that deployment is compromised, even when that might be beneficial to the business. With EASA, more people can work with and use your models, thus substantially increasing their value.

The problem for EASA is that it is in a class of one. With no other comparable products in the market it is difficult to get the word out as to how valuable the technology is. It is our hope that this paper helps to redress that balance.



**With EASA,  
more people can  
work with and  
use your models,  
thus substantially  
increasing  
their value.**



### FURTHER INFORMATION

Further information about this subject is available from  
[www.bloorresearch.com/update/2304](http://www.bloorresearch.com/update/2304)



### About the author

**PHILIP HOWARD**

**Research Director / Information Management**

**P**hilip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up his own company in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director, focused on Information Management.

Information management includes anything that refers to the management, movement, governance and storage of data, as well as access to and analysis of that data. It involves diverse technologies that include (but are not limited to) databases and data warehousing, data integration, data quality, master data management, data governance, data migration, metadata management, and data preparation and analytics.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to *IT-Director.com* and *IT-Analysis.com* and was previously editor of both *Application Development News* and *Operating System News* on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and written a number of reports published by companies such as CMI and The Financial Times. Philip speaks regularly at conferences and other events throughout Europe and North America.

Away from work, Philip's primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master), and dining out.

## Bloor overview

Bloor Research is one of Europe's leading IT research, analysis and consultancy organisations, and in 2014 celebrated its 25th anniversary. We explain how to bring greater Agility to corporate IT systems through the effective governance, management and leverage of Information. We have built a reputation for 'telling the right story' with independent, intelligent, well-articulated communications content and publications on all aspects of the ICT industry. We believe the objective of telling the right story is to:

- Describe the technology in context to its business value and the other systems and processes it interacts with.
- Understand how new and innovative technologies fit in with existing ICT investments.

- Look at the whole market and explain all the solutions available and how they can be more effectively evaluated.
- Filter 'noise' and make it easier to find the additional information or news that supports both investment and implementation.
- Ensure all our content is available through the most appropriate channel.

Founded in 1989, we have spent 25 years distributing research and analysis to IT user and vendor organisations throughout the world via online subscriptions, tailored research services, events and consultancy projects. We are committed to turning our knowledge into business value for you.

## Copyright and disclaimer

This document is copyright © 2016 Bloor. No part of this publication may be reproduced by any method whatsoever without the prior consent of Bloor Research. Due to the nature of this material, numerous hardware and software products have been mentioned by name. In the majority, if not all, of the cases, these product names are claimed as trademarks by the companies that manufacture the products. It is not Bloor Research's intent to claim these names or trademarks as our own. Likewise, company logos, graphics or screen shots have been reproduced with the consent of the owner and are subject to that owner's copyright.

Whilst every care has been taken in the preparation of this document to ensure that the information is correct, the publishers cannot accept responsibility for any errors or omissions.



Bloor Research International Ltd  
20-22 Wenlock Road  
LONDON N1 7GU  
United Kingdom

Tel: **+44 (0)20 7043 9750**  
Web: **[www.Bloor.eu](http://www.Bloor.eu)**  
email: **[info@Bloor.eu](mailto:info@Bloor.eu)**