

EASA

Making Excel behave like an Enterprise Application

EASA is a “topware” platform that enables authors (a.k.a. “citizen developers”) to create and publish custom, fit-for-purpose applications within a low code environment. The key difference between EASA and other low-code platforms, however, is that EASA enables authors to create apps that leverage existing assets such as spreadsheets—thus the “topware” moniker. In a recent interview with Sebastian Dewhurst, Director of Business Development at EASA, we discuss how the platform enables users to leverage their existing knowledge, creating sophisticated logic models within the well-known framework of Excel, and then publishing them as enterprise web apps.

What spurred the conception of EASA?

We refer to EASA as a low code platform, but we are very much about web-enabling existing tools and assets. We started out in the engineering sector, web enabling engineering and scientific models created not only in Excel, but also Matlab, R code, Python (including machine learning models). The initial business case for using EASA is usually an existing process which has been identified as bottlenecked and inefficient. Very often, there is an Excel spreadsheet that underpins the process. It’s telling that this holds true across many industries—in engineering and manufacturing, Excel is widely used as an engineering calculator; in insurance and financial services, spreadsheets are frequently used as pricing tools; in almost every industry, Excel is ubiquitous and the processes it underpins are largely uncontrolled. EASA eliminates this “Excel chaos”.

Can you tell us a little about EASA?

I think of Excel as the original low code development platform. It is not web-based or enterprise-friendly, but it is quite easy to get logic into it and has been the go-to tool for many years. We see some sophisticated spreadsheets which almost behave like apps—with the buttons, pull-downs, and so on. You can actually build a decent app in Excel—especially if you’re prepared to learn Visual Basic. However, spreadsheets are desktop apps at best—they are not web-based, lack authentication, version control, intellectual property security, and users’ data all too often ends up in local spreadsheets instead of in a central



SEBASTIAN DEWHURST
DIRECTOR OF BUSINESS
DEVELOPMENT

database—in other words, spreadsheets are not enterprise-grade. Yet the use of Excel is growing—possibly because most low code platforms are not as powerful and flexible as Excel when it comes to defining complicated logic. Furthermore, of course, everyone already knows how to use Excel. So, to a certain extent, there’s a reluctance to learn a whole new platform, when users already know Excel.

I think EASA can be regarded as a sort of bridge between low code and Excel. It brings the benefits of a low code platform (enterprise friendly, database-backed, authentication, version control, etc.), but it does allow you to leverage the power and flexibility of Excel.

How does EASA work?

With EASA, you can deploy an app by simply pointing our Excel Wizard to your template spreadsheet, and defining which parts of the spreadsheet you wish to expose to



users in a web app. EASA then auto-creates a web app. By default, the app will look quite similar to the original spreadsheet, or there is the option of using CSS to make the app look and feel completely custom. The web app performs exactly as the spreadsheet, because the master spreadsheet is copied to the server and acts as a logic engine for the app; the logic does not have to be redefined. So, all the functionality of Excel can be reproduced in the web app—including VB and macros. For many use-cases, this is far superior to other approaches for sharing spreadsheets such as network drives or SharePoint, and allows multiple users to use the app at any time. Every time a user saves a case, the information is saved back to a database. So if you have a 100 people using a pricing spreadsheet for example, the data is automatically saved in a database. Therefore, generating a report—say, the total

amount of business quoted based on region, or line of business—becomes trivial.

Can you provide a case study that highlights the usability of EASA?

Like many organizations, LeasePlan NZ has core systems supplemented by many desktop solutions such as spreadsheets. Many of these “end-user developed” solutions did not measure up in this age of increasing focus on compliance and both regulatory and business control.

LeasePlan has used EASA to build around 20 applications to date. Where requirements are well defined, it has been found that moderately complex applications can usually be built with about 20 hours of work effort. This validates the tremendous benefit of re-using knowledge already in the business in the form of Excel spreadsheets.

Thus far, LeasePlan has developed apps for: ROI and Deal Analytics; Pricing; Lease Settlement; Trial Balance reporting; Order Review, Order Confirmation and Registration Confirmation, among many others.

An additional unexpected benefit is EASA's ability to rapidly generate working prototypes for proposed applications.

EASA makes Excel behave like an enterprise application

What does the future look like for EASA?

We have committed to a new architecture for EASA 7.0 which will be released next year; it will be state-of-the-art and will feel more modern. While the EASA applications that we see today are competitive, useful, and functional, we want to improve the user-experience still further. Given the relationship we have with Microsoft Excel, we are also releasing a capability called Excel Desktop Client. With EDC, the customer does not need to build a web app to interact with the spreadsheet – rather, the EASA server provides a spreadsheet, but in a secured version which cannot be saved locally. Instead, when users save an instance, the data is saved back to a central database. This provides some of the benefits of EASA but without the overhead of actually building a web app. **CA**