Paul Filkin on the SDL AppStore

The SDL AppStore is a hub for the latest third-party apps designed to enhance translator productivity. With more than 300,000 downloads and 33,000 users, the SDL AppStore has quickly become a resource for translators, reviewers, and project managers to find and download apps that complement SDL software. The SDL AppStore is a free and open platform.

Paul Filkin has worked with SDL since 2006. As a client services director, his main focus is helping users of SDL technology get the most from their investment and providing advice to anyone who needs it. He started the SDL Community forums for translators, and has continued to build on this by creating communities of like-minded users so they can benefit more from sharing their experiences directly in an environment where SDL can most effectively get involved. For the past few years he has been working with a team focused on supporting developers who wish to work with the application programming interfaces for the SDL Language Platform. He also maintains the *multifarious* blog (https://multifarious.filkin.com), where he addresses many of the practical issues faced by translators and translation companies when using SDL translation technology. I had the opportunity to exchange a few emails with Paul to discuss his work with the SDL AppStore.

> *The AppStore is not only for language solutions.*

Can you give us a short timeline of the SDL AppStore (formerly OpenExchange)? Can you also tell us what SDL hoped to achieve with it and whether it met that goal?

The OpenExchange was launched in 2010 with SDL Trados Studio 2009. It provided five basic application programming interfaces (APIs) that allowed developers to create applications/integrations. (For those unfamiliar with term, APIs enable programs to communicate with each other.)

The old Trados applications were integrated into the workflow of many companies to such an extent that migrating to SDL Trados Studio would have been impossible without the ability to do something similar with the new Studio platform. But we also wanted to try and create a better platform that developers and users could employ to extend their use of our products. APIs are key to this.

It's been an interesting journey and we've learned a lot about how to support a platform like this. Not just in terms of making apps available, but in being able to properly support the developers who use the technology to create solutions for themselves, their companies, and for the SDL AppStore.

It's a journey we've not completed and it evolves constantly, but I think we're seeing really positive results, both for us and our customers. So, yes, I think we've met our original goal and I'm really proud of how SDL has allowed it to evolve in such an open way. I think the work we've done in this area has played a large part in the transformation of SDL. I believe that today we're seen as an open company with a mission to support the needs of our customers. This platform is key to delivering on that mission.

Now, here's a brief timeline of the implementation of the SDL AppStore:

- The SDL OpenExchange was launched in 2010 on translationzone.com with five APIs.
- The first SDL app to be published was SDL TTX It! The first third-party app was Excelling MultiTerm by Kaleidoscope.
- We hit 10,000 unique AppStore users around March 2013.
- In 2014, we started the Developer Experience Team outside of SDL core development so we could properly support third-party developers, maintain apps developed by SDL, and continue to grow what we started.
- We reached 100 apps in May 2014.
- With Studio 2014 we introduced the Integration API, which allowed developers to customize the user interface and create completely new features that could run within Studio.

- We ran a developer competition in the summer of 2014. By June 2014, we reached the milestone of 100,000 unique downloads. By the end of 2014, we had 20,000 unique users.
- In 2015, the Developer Experience Team created a custom library to simplify development in SDL Trados Studio. This helps novice developers get started and reduces the coding effort for more experienced developers.
- We provided two more APIs in 2016: the Terminology API and Batch Task API.
- We then ran an app idea competition in the summer of 2016 (the winner was SDL Analyse) before reaching more than 30,000 unique AppStore users later that year.
- OpenExchange was renamed SDL AppStore as we extended its use to support more of SDL's product offerings.
- We had 214 apps and exceeded 300,000 downloads at the end of May 2017.
- The most popular app to date is the Glossary Converter, with almost 19,000 downloads.
- In 2017, the Developer Experience Team created and released a custom library to simplify development with SDL Trados GroupShare.

Can you give us an idea of how many apps are presently offered on the SDL AppStore versus how many apps have been developed for private use?

It's worth noting that the SDL AppStore is not only for language solutions. We probably have almost as many apps available for our content management solutions as we do for language. SDL is committed to supporting the API ecosystem, and this means the AppStore, open source, and API lead development in general. In terms of language solutions, as of April 2017, there were 210 apps available on the AppStore. We're also approaching 300k downloads.

The apps consist of various kinds of solutions: some are fully integrated into the products, some are standalone tools, and some are integrated with other types of products. The reason I make this distinction is that the only apps we know about for sure are the plugins developed for private use, because we sign them electronically for our customers to remove the warning message when Studio starts.

There are over 1,000 developers who have signed up to be able to work on apps that could eventually be offered on the AppStore. This doesn't account for the fact that access to the APIs and the software development kit (SDK) are free. In addition, there's actually no need to sign up unless you intend to see your apps published on the AppStore. This is why knowing exactly how many apps have been developed for private use is pretty hard to determine. We used to require all developers to sign up to get access to the APIs, but we decided this was a barrier to growth. We wanted to make this process as simple as possible for developers.

So, coming back to your original question, I can tell you that I'm aware of almost 200 plugins that we've been asked to sign electronically for private use.

> *The SDL AppStore is something I see as a natural progression for any API strategy once you have a sufficiently large user base sophisticated enough to take advantage of it.*

Are there any success stories of developers who've been able to make developing and selling apps into a profitable offering?

This is another tricky one for us to answer. It's important to note that this is not a profit-making scheme for SDL. We provide the APIs as a value-added service and our website is primarily set up to serve free applications. There are around 44 paid apps available on the AppStore. In most cases, the developers actually redirect traffic to their own websites, where the sales are completed. So, we don't really know how successful they are. We also see some developers creating solutions that they sell only on their own websites, such as Kaleidoscope. In these cases, I think there are some developers who have been more successful than others at making this a more profitable exercise. The biggest difficulty is that the majority of our users won't pay additional money for apps, so it's not really a high-volume business.

Another important point to make is related to the motivation for creating apps in the first place. In regards to the apps on the AppStore, the prime motivation isn't money. Many people do this for exposure and to showcase their skills, and some just like to develop and solve problems. In particular, I believe there's a high degree of satisfaction to be gained by solving a problem that other users have mentioned from time to time. Jesse Good (a "translator by day and coder by night" in Japan) is a very good example of this, and many of his apps have come from discussions on the SDL Community forums (e.g., CleanUp Tasks, Copy Tags, and Target Word Count). In fact, the most downloaded app of all time, the Glossary Converter, is an app we would have loved to integrate into our products at some point, but the developer enjoys giving back to the community and has no intention of selling it or handing the source code over to us. He sees this as a way of being a part of the wider freeware community that he has been able to benefit from over the years.

Coming back to motivation, we also regularly provide contact details for third-party developers when we're asked by our own customers how they can create solutions for things. The most common requests are custom file types and integrations to content solutions. For example, we see requests for bilingual XML, where the customer wants a content management system (CMS) that exports/imports bilingual XML. We also have some particularly large customers who regularly employ third-party developers to work on integrations for an intriguing array of database systems, ranging from pure document management to alternative translation memory systems and terminology solutions.

I don't know if I would say these are success stories, but in addition to the work most of these developers already do in project management, translating, and general consultancy, I think the SDL AppStore has provided more strings in their bows! Perhaps one more interesting example is where a freelance translator who did a little programming in college took advantage of the APIs in Studio. In just a short period he became quite an expert on machine translation and how it could be used in conjunction with translation tools. So much so that after a couple of years he transitioned from freelance translating to a full-time programming role with Amazon!

Are there apps that have made it into the next version of Trados Studio? Has the success of apps influenced the course of development of the next version of Trados Studio (or MultiTerm)?

I like the idea of the SDL AppStore being an extension of the product and providing the user with some choice, rather than bloating it with more and more solutions that we thought were good ideas. We generally don't see apps making it into the core product, but there are a few exceptions.

A good example would be AnyTM, which solved a problem our user base had been forced to work around since the early Trados days. You probably recall the days when you tried to add your English (U.S.) translation memory (TM) to an English (Great Britain) project, but received a message that the languages didn't match. Now, any TM overrides this and allows you to use "any TM" you like! This app became part of the product with SDL Trados Studio 2015.

Another example would be InsertSymbols, which makes it much easier for users to enter non-standard symbols and accents without having to use ALT codes or have a custom keyboard installed. It's probably worth noting that the developer in both cases was actually an SDL developer. He was motivated to look for solutions to things being discussed on the SDL Community forums that would not fit into the wider strategic work he did in his day job.

In terms of apps that influence the course of development, that's harder to say. The core development teams tend to work on larger and more strategic projects. Most apps don't deliver the sort of functionality we would take and put into the product. But if I were to go out on a limb, perhaps the Glossary plugin was influential in guiding the teams toward providing a solution for termbase creation in Studio. There are also solutions like SDL MultiTerm Workflow, which is a rebranded QuickTerm provided by Kaleidoscope and built on top of MultiTerm Server. Here the work delivered was so comprehensive that when we looked at creating our own solution it really made more sense to partner with Kaleidoscope, which is what we did.

Why do you think other translation tool vendors have not followed suit? While many have APIs, there are no developers who have a comparable AppStore or who are proactively selling the idea of developing third-party tools.

I think the simple answers are either that they don't see the value in this—and I can recall listening to the negative views of some of our competitors on this several years ago—or that the effort required to support an infrastructure like this is too great. Many of our competitors do have APIs, but there is a world of difference between the capabilities and variety of features available through the APIs for the SDL Language Platform and those in our competitors' products. I think some of the cloud-based tools have pretty good representational state transfer (REST) APIs, but when it comes to the desktop tools, SDL is significantly more capable. With all the API documentation, SDKs, and sample applications freely available on the web, we're also able to allow a developer to create their integrations without talking to us at all.

You have to be prepared to support the developers who wish to use this platform. Once you open it up, the resource requirements to support it are not to be taken lightly. We've introduced an open-source policy for the apps we develop in-house and made this available to anyone. We also created a developer community that provides a great way of bringing people together to help each other. This community also helps us engage with our users and to continually evolve the platform we offer. I think many translation vendors are just not ready for this.

A more complex way to answer your question is to discuss the type of clients other vendors have. SDL has some very sophisticated customers with applications of their own, ranging from project management, terminology solutions, machine translation, accounting, content management systems, and more. These types of customers can have very demanding requirements that involve heavy automation of their systems with their translation tools. To be able to support this requires a robust and accessible API at many levels. Of course, we're certainly not unique in having customers with these kinds of needs. The SDL AppStore is something I see as a natural progression for any API strategy once you have a sufficiently large user base sophisticated enough to take advantage of it. Many translation tool vendors probably don't have this to the extent SDL has. ○

Remember, if you have any ideas and/or suggestions regarding helpful resources or tools you would like to see featured, please e-mail Jost Zetzsche at jzetzsche@internationalwriters.com.

Paul Filkin is a client services director for SDL Language Technologies, where he has worked since the end of 2006. His main focus is helping users of SDL technology get the most from their investment. He can be seen regularly on Twitter, the SDL Community forums he created, and on many of the public forums providing advice to anyone who needs it. More recently, he has been working with a team focused on supporting developers who wish to work with the application programming interfaces for the SDL Language Platform. He also maintains a blog addressing many of the practical issues faced by translators and translation companies in using technology for their work (http://multifarious.filkin.com). Contact: pfilkin@sdl.com.

Jost Zetzsche is the co-author of *Found in Translation: How Language Shapes Our Lives and Transforms the World*, a robust source for replenishing your arsenal of information about how human translation and machine translation each play an important part in the broader world of translation. Contact: jzetzsche@internationalwriters.com.