10 things you need to know about Uniface 10

(Original creator: a4dvo)

Working on a product like Uniface 10 feels a bit like the movie Oceans Eleven. A team of highly skilled professionals get together to pull off a feat that is considered undoable, or at least quite difficult to achieve. In fact there are quite a few similarities between the teams working in Amsterdam and the guys in the movie (Although me being Brad Pitt isn't one of them, apparently). For instance the team working on Uniface 10 consists of people, each with his or her own unique skill set, working towards a common goal. There are lead-characters and supporting cast and everything.

But enough about us, what about Uniface 10? The first thing you will probably notice, besides the fresh white paint instead of the dull gray, is the **UBAR**, an explorer-type address bar that lets you open your editors. The other thing that jumps from the screen is the **tabstrip.** A horizontal ribbon bar showing all the open editors, allowing you to switch easily between entity, component, and other **UDO**'s. A UDO, short for Uniface Development Object, is something we use in the lab to indicate a Uniface element (entity, field, component, library, and so on). These UDO's come in two distinct flavors – UDO's and main UDO's. The difference? Main UDO's have their own editor. Before you can start creating these UDO objects you need to create a **project** (also a UDO). In my mind one of the better features of Uniface 10 is the ability to organize your work in… well projects. Something I really hated doing was having to verify that all my changes were compiled. Sometimes I would just compile all from the command line (not an option if the application contains over a 1000 components). Just put everything in a project and hit compile. It is even possible to export the entire project and share it.

The **Integrated Development Environment** or IDE for friends (we dropped the term IDF) also allows you to use **snippets** in your code. Put your favorite piece of code (HTML, CSS, JavaScript or Uniface code) in a snippet and copy and paste into the wee hours of the night. We will provide you with some of our snippets but feel free to create your own. Just as in any other film there are some characters that will not make it to the end of the movie. For instance the (pirate voice) **URR** file is no longer part of Uniface. I have always had great difficulty understanding the Uniface Runtime Repository and am glad to see it go. Seeing it off with a bonfire made from my URR notes and the documentation. The UAR deployment option is the way to go from now on. The removal of the application model entity from the meta dictionary hit me a little harder.

In Uniface 10 the **application model is just a namespace**. Need to move an entity from one model to another? Just rename the model part of the entity name. So be careful, especially with typos. We pulled all the triggers. And replaced them with **script containers**. From now we will supply you have with a Declaration container and a Script container to write your **ProcScript** in. Just for good measure we will give you three containers in the entity editor (Declarations, Collection and Occurrence). But that's it. Want to create a trigger? Just use the keyword **trigger** in the same fashion you define an operation or an entry. Just to raise the stakes a little we have renamed the trigger names to make them more uniform.

What else do you need to know? Well the first version is an early adopter version. We are still working on a lot of stuff, that will be made available in future release. For instance there are no library elements yet. So you have to do without include procs, messages and such. But we just couldn't wait to show you what we have done so far.