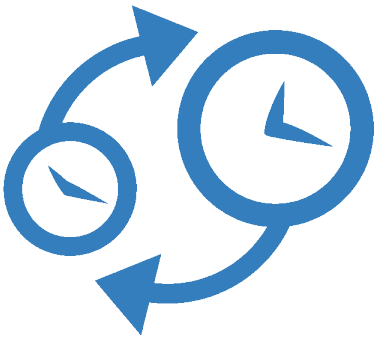
Do I need to compile this?

(Original creator: theo neeskens)



Over the years many Uniface developers have created tools on top of the Uniface Repository. One tool that has been made by many, is one that looks for "dirty" objects: objects that were modified after they were last compiled successfully. In Uniface 9 such a tool would have been based on comparing the fields UTIMESTAMP (modification date of the source) and UCOMPSTAMP (compilation date of the source) of various Uniface Repository tables. In Uniface 10 this has changed, mainly to align the repository with the export format that has been optimized for version management:

- The modification date of a development object source is only found in the main object. And yes, it is also updated when you change a sub-object. So if you change a modeled field, the UTIMESTAMP of the entity is updated.
- The compilation date of a development object is no longer an attribute of the source code. It does not have much value to know when the source was last compiled, if you can't be sure that the compiled object was the result of that compilation. Someone may have copied a different compiled object to the output folder. The only real compilation date is that of the compiled object (file on disk or in a UAR).

Uniface 10.3 is the first release of Uniface 10 that is shipped with meta definitions: the new DICT model is published. So now you can re-engineer the tools that you made for Uniface 9. In order to make it easier to (re)write a tool, the $ude("exist") function has been enhanced to return the attributes of the compiled object (file on disk or in a UAR) such as the modification date.

Compiling objects because their source code has changed

It is not just components that require compilation. There are 14 types of development object that require compilation and generate a file in your resources. I have attached a sample tool that checks whether these objects are "dirty" and therefore require compilation. The tool gathers the source modification date from main development objects, and the compilation date of the compiled objects. In some cases, one main development object (such as a message library) results in many compiled objects (messages). The tool uses $ude("exist") to check the compilation timestamp of the compiled object and $ude("compile") to compile it. The attached export file contains a full project export, so when you open project WIZ_COMPILE, you will see the whole thing. You can download the export here: [download id="7581"] And here is a file with some test data for each object type: [download id="7585"] You will need the Uniface 10.3 DICT model to compile the tool. The new DICT model for Uniface 10.3 is delivered in umeta.xml in the uniface\misc folder of your development installation.

PLEASE NOTE: The sample tool does NOT take into account that a component may require compilation because a modeled entity or an IncludeScript has changed. See below.

Compiling components because a modeled entity has changed

Please note that the attached sample does NOT check if a component requires compilation because a modeled entity has changed. If you had this check in your Uniface 9 tooling, you also need to implement it in your new tooling. A Uniface 9 based example for this issue can be found here:

http://theunifaceuniverse.blogspot.nl/2011/04/change-entity-compile-forms.html You would need to simplify this for Uniface 10 because the modification date is only on the main development object.

Compiling objects because an IncludeScript has changed

Please note that the attached sample does NOT check if a development object requires compilation because an IncludeScript has changed. To implement this is quite tricky, as you would have to find the #INCLUDES within the IncludeScript code, and handle the fact that they can be nested. To correctly parse all of that might not be much faster than just compiling everything...