Triggers’ default behavior in Uniface 10

(Original creator: HenkVanDerVeer)

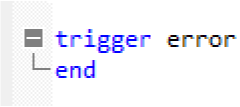
With the release of patch F205 for Uniface 10.2.02, the Uniface 10 compiler has changed to ensure compatibility with Uniface 9 for triggers having default behavior. This blog explains when and how Uniface handles ‘empty’ triggers and invokes default behavior. A small subset of the triggers in the Uniface model (*) falls back on default behavior if these triggers do not contain executable code. A typical example is the On Error trigger for a field or entity. If you do not define the trigger, the Uniface run time engine will still invoke default handling for error situations. If the trigger has been defined with executable code, only that code is executed, and the default behavior is suppressed. (*) see the Uniface documentation, section “Triggers with Default Behavior” for the complete list of applicable triggers.

When does a trigger ***not*** have executable code and revert to default behavior?

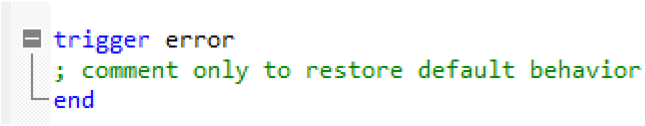
In Uniface 10, default behavior is invoked if any of the following conditions is true:

- the trigger is not declared at all
- the trigger declaration contains no executable code
- the trigger declaration only contains one or more pre-compiler directives that do not result in executable code
- the trigger is *undeclared.*

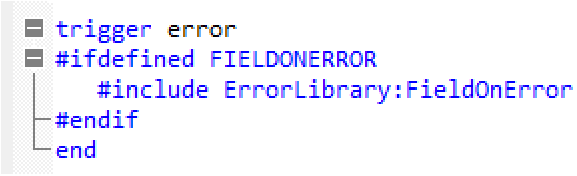
Some examples:



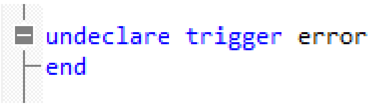
```
trigger error
end
```



```
trigger error
; comment only to restore default behavior
end
```



```
trigger error
#ifdefined FIELDONERROR
    #include ErrorLibrary:FieldOnError
#endif
end
```



```
undeclare trigger error
end
```

What is the impact of the Uniface Inheritance model, how to restore default behavior?

Behavior defined in code containers is inherited at ‘lower’ or ‘derived’ levels. Examples:

- a modeled entity subtype and its fields inherit from a supertype and its fields
- a component can inherit from a modeled component (called *component template* in Uniface 9)
- an entity or a field in a component’s data structure inherits from the modeled entity or field.

Inheritance can take place over multiple levels, but that’s beyond the scope of this blog. In Uniface 10, inheritance of code in containers is *module-based*. Code is contained in explicitly-declared triggers, entries and operations. If a trigger is declared again on the inheriting level, that definition takes preference and replaces the definition that was inherited from the higher level. To suppress an inherited trigger in Uniface 10, use any of the options described above: declare an empty trigger, declare a trigger with comment only, or undeclare the trigger on the lower level. An ‘empty’ trigger or undeclared trigger will fall back on default behavior if that is applicable for that trigger. The following table shows some examples:

Modeled Field trigger error	**Component field trigger error**	**Result**
not declared	*not declared*	Default error handling
trigger error end	*not declared*	Default error handling
trigger error if ($error = 0105) … some code return -1 endif	*not declared*	User defined error handling

trigger error if ($error = 0105) … some code return -1 endif	trigger error end	Default error handling
trigger error if ($error = 0105) … some code return -1 endif	undeclare trigger error end	Default error handling

What has changed since patch F205?

With the solution for Issue # 31689 in patch F205 (Uniface 10.2.02), explicitly-declared triggers that are effectively empty now fall back on default behavior, if that is applicable for that trigger. Before this patch, an explicitly declared trigger in Uniface 10 without executable code or with comment only would not only break inheritance, but also suppress its default behavior. Prior to F205, the only way to ensure that default behavior was invoked was to not declare the trigger or to undeclare the trigger. In case of inheritance of a trigger from a higher level, the only way to restore default behavior on the lower level was to undeclare the trigger.

What has changed since patch F206?

In patch F206 the automatic migration logic in Uniface 10 was changed to benefit from the modifications in patch F205. Before patch F206, the migration would attempt to assess whether a trigger container coming from Uniface 9 with potential default behavior contained comment or entry declarations only. If so, the trigger would be commented out or undeclared. This approach was not watertight and had a few disadvantages, like adding code during the migration (‘code pollution’) and causing additional compiler warnings compared to Uniface 9. When migrating a Uniface 9.6 or 9.7 export file into Uniface 10 after installing patch F206, all triggers, including those with potential default behavior, are migrated ‘as is’. Patch F205 is compatible with code migrated into a Uniface 10 using a patch prior to F206, i.e. there is no need to redo the migration. However, if you want to benefit from the changes in the migration, you should migrate after installing patch F206 or higher.