

## Saving unsaved changes using uQueryCanDeactivate / avNotifyDeactivation

In EPC831 or later of the framework it is possible to add a redefined method to command handlers called uQueryCanDeActivate.

This method is invoked when the end user tries to move from your command handler to somewhere else in the framework, or when they try to close down the framework.

It is run when the user clicks on another command tab, or clicks on a new instance of the business object, or clicks on another business object or application.

It is particularly useful for saving unsaved changes.

The routine can check whether changes need to be saved. If so, the user can be asked "Do you want to save your changes before continuing?" (Yes/No). If they answer Yes, their changes can be saved.

At the time the routine is run, the command handler still has all its values as they were before the user attempted to move away. So checking for unsaved changes, and saving those changes, is easy.

To use it, you need to set the avNotifyDeActivation property in the command handler's initialize routine

```
* Handle Initialization
Mthroutine Name(uInitialize) Options(*REDEFINE)
* Do any initialization defined in the ancestor
Invoke #Com_Ancestor.uInitialize
* Activate Check for unsaved changes (Unsaved changes logic)
set #Com_Owner avNotifyDeactivation(TRUE)

Endroutine
```

Then add a redefined uQueryCanDeActivate routine to your command handler

```

* The Framework initiates this when the user moves to another command tab, or business
object instance, or business object, or application, or closes the framework.
* (The framework may initiate this method multiple times)

MTHROUTINE NAME(uQueryCanDeactivate) OPTIONS(*REDEFINE)
* Define_Map For(*Result) Class(#vf_elBool) Name(#Allow)

#Allow := True

* My flag indicating that an unsaved change has occurred
if '#pty_NeedsSaving *eq TRUE)'

* If something needs saving, ask the user if they want to save it

USE BUILTIN(MESSAGE_BOX_SHOW) WITH_ARGS('YESNO' 'YES' *Default *Default 'The notes have
been changed. Would you like to save your changes before continuing?') TO_GET(#MSG_RET)

if '#MSG_RET *eq YES'

* Save everything
<< my save logic>>

endif

#pty_NeedsSaving := False

endif

endroutine

```

A more complicated version could set #Allow to false if there was an error during the save, and in that case the user would not go to where they clicked, (or the framework would stay open if they were attempting to close it)

## ***Comments/Warnings***

You must set #Com\_Owner.avNotifyDeactivation to TRUE if you want to use it.

The uQueryCanDeactivate method may be invoked several times by a single user click, so it is important to reset (your equivalent of) #pty\_NeedsSaving in the routine, so that subsequent invokes do not check with the user again.

There are warnings in the feature help about using method uQueryCanDeactivate and property avNotifyDeactivation in version EPC831 of the framework. These warnings can be ignored.

This method can be useful when dealing with objects that don't signal when they have changed. By using this method you can compare a snapshot of the object's current status with its as-loaded status, just at the point the user has finished with the command handler. (Rather than checking for changes every second)