

Off-Line Documentation template:

Context change (background/for ground) switches

1) Operational Goals

Provide all the house keeping operations needed for all context changes, like clip conversion and deactivating controls.

2) Fundamental, "key", or cornerstone architectural requirements (POSTMORTEM)

- Need to decode the event and handle it
- Need to set flag `fInBackground` correctly
- Need to set `fSleepVal` correctly
- Need to convert clip (get a copy from system or give one to the system)
- Pass the `HandleOSEvent` message on the `fTarget` chain.

3) Model of the implementation fulfilling these key requirements (POSTMORTEM)

- a) In response to an `OSEvent` where the application is changing its context the application needs to correctly set the `fInBackground` flag and `fSleepVal` correctly.
- b) The application also needs to perform any necessary clip conversion at that time too.
- c) All the windows and documents need to be given a chance to do any house keeping as the application changes its status with respect to being in the background.

4) Impact/scope of the implementation on the existing body of code (POSTMORTEM)

- `DApplication::OSEvent` - decodes the event and if needed sets the `fInBackground` flag and `fSleepVal` correctly and it initiates the clip conversion process.
- `DEventHandler::HandleOSEvent` - provides for event handling of this type of event for all event handler objects
- `DScrollWindow::HandleOSEvent` makes sure that the scroll bars are correctly visible or NOT, by overriding the `DEventHandler::HandleOSEvent` member function.

5) Coding notes (gotchas, warnings, process thoughts, items to revisited later...)

- Had a problem with the `HandleOSEvent` not getting to `TEditWind` (because I forgot to declare the override as virtual) resulting in highlighted text not being un-highlighted.
- For ThinkC users, make sure that the finder flags are set such that suspend/resume messages get sent to your application. (the flag settings are under the "Set Project type..." menu option.

6) Testing notes(bug types, what made a bug hard to fix, what could have been done to catch it sooner....)

- I failed to catch the incorrect handling of the visibility of the scrollbars until after the release of the DinkClass demo, even after all the other testing and tweaking I still missed that user interface guideline requirement. I need to think of a way to catch these kinds of bugs before the public finds them!!!!

7) Process notes (what process did you follow, could it be improved)

