

NiconNotifyOSD

Sistema de notificaciones para las NikonApp

¿Que es NikonNotifyOSD?

Esta será una explicación general y global de lo que pretende ser el proyecto **NikonNotifyOSD** y lo que representa para todo el ecosistema de aplicaciones Nikon.

Todo sistema, toda aplicación, todo entorno informático o análogo debe notificar sobre sus eventos a los usuarios de forma tal que estos puedan saber en tiempo real y a ciencia cierta que esta pasando con una tarea o que ha pasado con su solicitud de actividad; partiendo de esta premisa y entiendo que toda la familia de NikonApps empiezan a ser parte de un solo sistema y que como tal estas deben comunicarse con el usuario final, nos vemos en la necesidad de crear un sistema de notificaciones único para todas las aplicaciones que les permita informar al usuario final sobre alguna tarea en ejecución terminada, informarle sobre alguna acción a ejecutar, informar sobre una advertencia o sobre una acción peligrosa que este a punto de realizar entre otras muchas notificaciones que un usuario final puede recibir dependiendo del sistema, aplicación o tarea que este desarrollando en ese momento.

NikonNotifyOSD será una librería de notificaciones al estilo de JOptionPane y toma como experiencia de usuario el sistema NotifyOSD del sistema operativo Ubuntu basado en el proyecto de software Gnome Desktop Environment. Esta librería estará escrita en lenguaje Java y será una librería de código fuente abierto liberada bajo licencia GPL V3, la librería será usada en todas las apps Nikon y será parte básica de los proyectos NikonDesktop y NikonOS, pero también podrá ser usada por otras aplicaciones Java que deseen incorporar y hacer uso de este proyecto si así lo desean sus desarrolladores y/o mantenedores.

A diferencia de los actuales sistemas de notificaciones el NikonNotifyOSD pretende convertirse en un sistema de notificaciones inteligente y que sea mas que solo mostrar mensajes al usuario en alguna parte de la pantalla, para permitir al usuario poder ejecutar tareas adicionales desde la mismas notificaciones y que además permita a toda la familia de NikonApp poder comunicarse unas a las otras enviando información sobre procesos o solicitando la ejecución de tareas especificas de una a la otra.

El desarrollo de la librería estará a cargo de la comunidad de software Libre NikonDevTeam desde donde se planeará y desarrollará toda la plataforma y donde se asegura que sea una librería de la comunidad y para la comunidad.

NiconNotifyOSD como funcionará

Ya sabemos un poco sobre lo que pretende ser NiconNotifyOSD, ahora conoceremos en teoría como debería funcionar en un futuro cercano el proyecto y que posibles prestaciones podría tener en un futuro mediano y largo para todas las apps que lo usarán.

Caso de ejemplo 1 :

El usuario X esta usando NiconPersonal para la gestión de sus contactos personales y como tal ejecutará acciones que generarán notificaciones de parte del sistema para informarle sobre el estado de la operación, bien sea que notifique sobre una tarea exitosa o una tarea cuyo final ha sido incorrecto; en ambos casos la mayor parte de Apps crearán ventanas informativas con la descripción del mensaje y en la mayoría de ocasiones proveerán un icono alusivo al tipo de mensaje emitido y hasta algún sonido que ayude a hacer llamativa la notificación para el usuario, supongamos que el usuario X decide eliminar un contacto de su libreta, en este caso NiconPersonal debería emitir un mensaje de confirmación solicitando verificar que el usuario X en verdad desea eliminar el contacto A de su libreta de contactos, pues bien es aquí donde entra en funciones NiconNotifyOSD, NiconPersonal hará uso de la librería para mostrar dicha información de solicitud al usuario creando un evento y mostrando dicho evento al usuario final luego de recibir una respuesta de NiconNotifyOSD NiconPersonal ejecutará la acción que el usuario ha solicitado a través de la librería.

Caso de ejemplo 2:

El usuario X esta usando NiconPersonal y su modulo NiconMail para el envío de un mensaje de correo electrónico al contacto Z de su lista de contactos de NiContacts, el mensaje ha sido enviado y esto debe notificarse al usuario, pues bien NiconPersonal hará uso de la librería NiconNotifyOSD para ello, bien podrá usar una notificación del tipo small (Esquinas del monitor) o long (Centro de la pantalla con botón de aceptar) en ambos casos se mostrará un mensaje al usuario notificando de que la tarea ha terminado exitosa o erróneamente.

Caso de Ejemplo 3:

El usuario X esta usando NiconPersonal en su modulo NiconTwitt para gestionar su cuenta de twitter y el usuario @AAD le ha enviado un mensaje interno en su cuenta de twitter, pues bien se mostrará una notificación del evento al usuario en su pantalla y la una opción para responder a la misma desde la notificación en si.

Proceso técnico de uso

A continuación se generará un esquema gráfico de una línea de actividades que se ejecutarán al momento de usar la librería NikonNotifyOSD.

**Usuario X ----- ejecuta una actividad -----> Esta genera un ----->
EVENTO -----> que genera una -----> NOTIFICACION.**

El simple proceso anterior describe una línea de pasos a seguir luego que un usuario ha solicitado o ejecutado una actividad dentro de una aplicación, la ejecución de esta actividad generará un EVENTO que podrá ser mostrado al usuario a través de una notificación gráfica en pantalla.

Que es un Evento?

En NikonNotifyOSD se manejará un concepto clave y esencial que define la estructura básica del proyecto un EVENTO. En el proceso anterior el usuario X solicita a una app la ejecución de una tarea y como resultado esta app deberá informar sobre el resultado de esa tarea, pues bien un evento es definido como la información a mostrar al usuario luego de haber ejecutado una tarea o proceso y que deberá ser mostrada en pantalla.

El evento se considera la información como tal a mostrar al usuario en pantalla tales como el título del evento, una descripción básica o general de lo que se desea mostrar, el código del tipo de evento generado.

Que es una Notificación?

Una notificación se considera como la información mostrada en la pantalla del usuario y que expresa todo lo que el evento desea informar, es la síntesis de todo el proceso e el cual el usuario final puede conocer lo que la aplicación desea comunicarle.

Tipos de Notificaciones:

NiconNotifyOSD proveerá una amplia gama de tipos de notificaciones que podrán ser mostradas al usuario por pantalla, el siguiente listado será el listado de notificaciones soportadas para la versión 0.1.0 y a partir de ahí será incluidas y mejoradas en canto el sistema avance en el tiempo.

Las notificaciones se dividen en dos grupo las WindowNotify y las DesktopNotify, a continuación habrá un listado de cada una de las opciones para la version inicial:

WindowNotify

- **MessageNotify:** Estas serán ventanas de mensajes solo mostrarán texto e iconos.
- **InputNotify:** Estas serán ventanas para el ingreso de datos por parte del usuario se compone de icono, mensaje, campo de texto y boton cancelar y aceptar
- **ConfirmNotify:** estas serán ventanas de confirmación, se componen de Icono, Mensaje, y dos botones si / no.

Desktop Notify

- **DesktopMessage** esta será una pequeña ventana emergente autocerrable que mostrará información
- **DesktopConfirm** esta será una pequeña ventana emergente autocerrable que mostrará una información y además dos botones de opción cancelar o aceptar.

Cabe señalar que este listado es un listado básico de notificaciones en base a la versión inicial del proyecto, con el paso del tiempo el proyecto proveerá muchas más opciones de acuerdo a las necesidades o deseos del equipo de desarrollo.

NiconNotifyOSD en Java

Luego de haber entendido un poco de forma básica que desea NiconNotifyOSD y de que manera se espera que actúe vamos a dedicarnos a estructurar en si como funcionará y de que se compondrá NiconNotifyOSD desde el código fuente.

Ya se tiene por entendido que una notificación se compone de un evento que se muestra en pantalla informando al usuario sobre una actividad por realizar, en proceso o realizada por una app, pues bien en base a ello definiremos las clases que intervienen en el proceso y de que forma se comunicaran entre ellas para crear el sistema de notificaciones mas completo y poderoso del mercado.

Las Notificaciones NiconNotify:

una notificación se ha definido como una ventana gráfica que es mostrada en pantalla con una información descriptiva que se muestra al usuario final, las notificaciones en el NiconNotifyOSD será estructuradas de forma tal que permita ser fácilmente usada por los desarrolladores y gráficamente útiles para el usuario, así definimos que NiconNotifyOSD maneja dos tipos de sistemas de Notificaciones:

1. **WindowNotify:** Son Notificaciones en modo ventana al estilo de las encontradas en el JOptionPane, estas notificaciones son mostradas en el centro de la pantalla y pueden ser utilizadas para mostrar información o bien ser usadas en el proceso de solicitud de información
2. **DesktopNotify:** Son notificaciones en modo globo de información, son pequeños componentes gráficos que no hacen uso del gestor de ventanas y que ocupan un espacio relativamente pequeño en comparación con las WindowNotify, su uso es meramente informativo.

Las notificaciones serán objetos del tipo JDialog y/o JFrame que permitirán cargar otros componentes menores del tipo JLabel y JButton para ser mostrados al usuario, una Notificación permitirá mostrar la información de un evento sucedido teniendo así que una clase que instancia un Objeto Notify recibirá en su constructor como parámetro un Objeto NiconEvent con la información a enseñar.

Los Eventos NiconEvent:

Un evento será definido como un objeto que contiene toda la información que el sistema desea enseñar al usuario, los eventos podrán ser tratados como objetos serializables y podrán ser almacenados en una base de datos o sistema de archivos, los eventos por lo general solo será mostrados a traves de un objeto NiconNotify o a traves de una terminal en algunos casos, los objetos NiconEvent tendrá un estado que ayudará a identificarlo dentro del NiconEventCenter, un evento podrá estar en los siguientes estados, 0 visto y tratado 1 visto 2 en espera, cada NiconEvent estara compuesto de la siguiente forma:

- Código del evento
- titulo del evento
- texto del evento

- tipo de mensaje
- tipo de Notificación
- stateEvent

Como se estructura y se usa NikonNotifyOSD

En el proceso de diseño se ha pensando en la posibilidad de un sistema versátil y potente, donde los eventos sean independientes y su complemento sean las Notify, juntas crean una plataforma versátil y muy prometedora además de ofrecer una implementación simple e inteligente.

NikonNotify se compone de dos cosas NikonEvent como abstracción de la notificación y WindowNotify como la notificación final, ahora un ejemplo de implementación

//Creando el objeto NikonEvent

```
NikonEvent event=newNikonEvent("Titulo de la notificación:"Mensaje a  
mostrar",NikonEvent.NOTIFY_OK,0,0);
```

//Mostrando la Notificación

```
NotifyFactory.showNotifyMessage(event);
```

NikonNotifyOSD provee una clase NotifyFactory encargada de servir cada uno de los métodos de acceso para todas las notificaciones servidas por la librería, pasando el objeto NikonEvent como parámetro.

Podremos por ejemplo usar un NikonEvent para todas las notificaciones servidas así estos objetos pueden pertenecer a varias notificaciones al mismo tiempo, en versiones posteriores cada objeto NikonEvent podrá ser almacenado en disco según el desarrollador lo necesite e incluso se trabajará en un nuevo sistema de Log para manipulación de errores que ahora podrán servir una NotifyLogMessage para servir los errores del sistema al usuario y permitir enviarlos al desarrollador para hacer su respectiva corrección.

Características a desarrollar para la Version 1.0

A continuación se detallará el listado de características para el lanzamiento oficial de la librería en el próximo mes de Julio:

- Reemplazo total de JOptionPane para las WindowNotify
- Serialización y persistencia de Objetos NikonEvent en disco
- Creación de las desktop Notify

Pendientes para próximas versiones

- Personalización gráfica de todas las NikonNotify, ahora los desarrolladores podrán adatar las WindowNotify y las desktopNotify a su app con colores y fuentes personalizados.
- NikonEventCenter como centro de notificaciones, todas las notificaciones podrán ser almacenadas en una ase de datos. Como el centro de notificaciones de MacOS X mountain Lion al lado derecho de desplegará un panel con todas las notificaciones pendientes o almacenadas en el sistema sin ser vistas, desde ahi podrán ser eliminadas.
- Nuevo NotifyReporErrorMessage para mostrar errores o bugs de tu app y permitir que puedan llegar a tu correo los objetos NikonEvent para ser tratados.

