Obtaining battery status using HAL

Overview

This article shows how to obtain battery status using HAL

Basic Idea

HAL is hardware abstraction layer, which hides hardware specific interfaces from the application. You can get many . There are utilities **lshal**, **hal-find-by-property**, **hal-find-by-capability**, which are useful for investigating HAL database lshal manual page.

Then there is *libhal* library allowing your application to access and alter HAL database. As normally you need to add following lines to .pro file to start using libhal:

```
unix {
    CONFIG += link_pkgconfig
    PKGCONFIG += hal
}
```

This is Qt application, which uses libhal to obtain battery charge level in percents and current charge level expressed normally in mAh. `main()` function does initialization and shutdown of HAL context, searches for all devices with battery capability. Then `update_label()` function tries to read three keys from the first found device and set them as a label.

```
#include <hal/libhal.h>
#include <QApplication>
#include <QLabel>
#include <QDebug>

#define BATTERY_CAPABILITY      "battery"
#define BATTERY_PERCENTAGE      "battery.charge_level.percentage"
#define BATTERY_CURRENT         "battery.reporting.current"
#define BATTERY_UNIT            "battery.reporting.unit"

static void update_label(LibHalContext *ctx, const char *udi, QLabel *label)
{
    DBusError error;

    dbus_error_init(&error);
    int percentage = libhal_device_get_property_int(ctx, udi,
                                                    BATTERY_PERCENTAGE,
                                                    &error);
    if (dbus_error_is_set(&error)) {
        qDebug() << error.message;
        dbus_error_free(&error);
    }

    int current = libhal_device_get_property_int(ctx, udi,
                                                 BATTERY_CURRENT, &error);
    if (dbus_error_is_set(&error)) {
        qDebug() << error.message;
        dbus_error_free(&error);
    }

    char *unit = libhal_device_get_property_string(ctx, udi,
                                                   BATTERY_UNIT, &error);
```


(C) Copyright Nokia 2014. All rights reserved.

```
    if (!unit) {
        qDebug() << error.message;
        dbus_error_free(&error);
    } else {
        label->setText(QString("Battery info: %1 %2, %3%")
                       .arg(current).arg(unit).arg(percentage));
        libhal_free_string(unit);
    }
}

int main(int argc, char *argv[])
{
    DBusError error;
    QApplication a(argc, argv);
    DBusConnection *conn;
    QLabel label;

    LibHalContext *ctx = libhal_ctx_new();

    dbus_error_init(&error);
    conn = dbus_bus_get(DBUS_BUS_SYSTEM, &error);
    if (!conn) {
        qDebug() << error.message;
        dbus_error_free(&error);
    }

    if (!libhal_ctx_set_dbus_connection(ctx, conn))
        qDebug("Cannot init HAL context with DBus");

    if (!libhal_ctx_init(ctx, &error)) {
        qDebug() << error.message;
        dbus_error_free(&error);
    }

    int num_devices;
    char **devices = libhal_find_device_by_capability(ctx, BATTERY_CAPABILITY,
                                                      &num_devices, &error);
    if (devices) {
        if (num_devices > 1)
            qDebug() << QString("%1 batteries found").arg(num_devices);
        if (devices[0]) {
            update_label(ctx, devices[0], &label);
        }
    } else {
        qDebug() << error.message;
        dbus_error_free(&error);
    }

    label.setAlignment(Qt::AlignCenter);
    label.setFont(QFont("Nokia Sans", 35, QFont::Bold));
    label.show();

    int ret = a.exec();

    if (devices) {
        libhal_free_string_array(devices);
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
    if (!libhal_ctx_shutdown(ctx, &error)) {
        qDebug() << error.message;
        dbus_error_free(&error);
    }

    if (!libhal_ctx_free(ctx))
        qDebug("Cannot free HAL context");

    return ret;
}
```


(C) Copyright Nokia 2014. All rights reserved.