

System Overview

VT-OS is a single-tasking operating system. The tasks of the system can be categorized into two groups: interrupt-level processes and foreground processes. In this chapter of system overview, some of the foreground processes are introduced in order to explain the system flow. The foreground processes of the system, which are discussed, are as follows:

- Battery Low Dialog
- Global Find
- Inlay

Before the discussion of the foreground processes, the *SystemHandleEvent* is first introduced. It is a foreground function that handles all the system events to perform different corresponding tasks.

1. *SystemHandleEvent*

All the foreground processes in the OS are event-driven. It means that the processes proceed according to events. For each application, there is an *EventLoop* where events are extracted from the event queue by using *EvtGetEvent* and then the events are passed down the hierarchy of system functions. If one of the system functions handled an event, then the event is not passed further down the hierarchy. The *EventLoop* is shown as follows:

```
void EventLoop()
{
    EvtType      event;

    do
    {
        EvtGetEvent(&event);
        if (!SystemHandleEvent(&event))
            if (!MenuHandleEvent(&event))
                if (!ApplicationHandleEvent(&event))
                    FormDispatchEvent(&event);
    }
    while (event.eventType != EVT_APP_STOP)
}
```

As shown in the above source code, the system foreground functions are all in the *SystemHandleEvent*. Inside the *SystemHandleEvent*, all the system foreground functions are placed in hierarchy too.

2. *Battery Low Dialog*

This foreground function is called to handle the situations of BATTERY POWER LOW and BATTERY DOOR OPENED.

When either one of the situation happens, interrupt is received and handled by interrupt service routine. Then the interrupt service routine sends *POWER_EVENT* back to the *BatteryWarningPopup* system foreground function. When the *BatteryWarningPopup* gets the event, it processes the event according to its type. The following table shows the two different *POWER_EVENT* and how *BatteryWarningPopup* processes the event.

<i>Events that are passed to BatteryWarningPopup</i>	<i>Actions to be taken by BatteryWarningPopup</i>
POWER_EVENT EVT_BAT_DOOR_OPEN	<ul style="list-style-type: none"> • The event indicates that the battery door in the back of the PDA is opened when the power of the PDA is turned on. • When <i>BatteryWarningPopup</i> received the event, it sends out two events to close the current running application and request the system to power down the PDA. The two events are <i>EVT_APP_STOP</i> and <i>EVT_POWER_DOWN</i> correspondingly.
POWER_EVENT EVT_MAIN_LOW	<ul style="list-style-type: none"> • The event indicates that the power level of the system is under the warning level. • When <i>BatteryWarningPopup</i> received the event, it popup a Battery Low Dialog Box to notify the user of the battery low message.

There are a number of functions for the Battery Low Dialog and are documented in another document.

3. Global Find

Global Find is a process that allows user to find keywords or a sentence in all databases of applications. When either clicking on the mapped inlay button or the shortcut button starts Global Find, input-screen of Global Find is popup. After the user inputted a search string and proceeded the searching, Global Find starts to search. The searching algorithm is as follows:

- a) The current active application is searched first. A copy of the current application is loaded and launched again with a launch command LAUNCH_CMD_FIND. Since another copy of the application is loaded, the status of the current active application is not altered.
- b) The applications, which are registered into the system and having Global Find capability, are loaded with a launch command LAUNCH_CMD_FIND.
- c) After all the applications are search, the searching result is popup

There are a number of functions that can be used for the application to handle the LAUNCH_CMD_FIND. They are documented in another document.

4. Inlay

There is an inlay at the bottom of the screen. It provides a number of regions for a number of predefined functions. The regions include OK, EXIT, Main Menu, Keyboard ABC, Keybaord 123, two customizable dots and Main.

When pen actions are within the bounds of one of the regions, the corresponding region is selected and event is sent to the current active application to notify the selection of a particular region.

The functions for handling the inlay actions are documented in another document.