

Retrieve_General_phone_information_using_Shared_Data_API

The *PSVariables.h* provides Publish & Subscribe keys related to general phone information like availability of SIM, GPRS, IRDA, Network etc.

Note: The *PSVariables.h* can be found in *Shared Data API* which is a part of the SDK API Plugin for S60 3rd SDK MR.

Example code

GPRS Availability:The following code displays GPRS Availability status:

```
void CPSVariablesAppUi::GetGPRSAvailabilityStatus()
{
    TInt value;
    TInt ret=RProperty::Get(KUidSystemCategory, KPSUidGprsAvailabilityValue, value);
    if(KErrNone==ret)
    {
        CAknInformationNote* informationNote = new (ELeave) CAknInformationNote;
        TBuf<30> msg;
        switch(value)
        {
            case EPSGprsAvailable:
            {
                msg.Copy(_L("Gprs Available"));
                informationNote->ExecuteLD(msg);
                break;
            }
            case EPSGprsNotAvailable:
            {
                msg.      Copy(_L("Gprs Not Available"));
                informationNote      ->ExecuteLD(msg);
                break;
            }
            case EPSGprsAvailabilityUnknown:
            {
                msg.Copy(_L("Gprs Availability Unknown"));
                informationNote      ->ExecuteLD(msg);
                break;
            }
        }
    }
}
```

Status of Auto Lock:The following code displays the Auto Lock status (On/Off):

```
void CPSVariablesAppUi::GetAutoLockStatus()
{
    TInt value;
    TInt ret=RProperty::Get(KUidSystemCategory, KPSUidAutolockStatusValue, value);
    if(KErrNone==ret)
    {
        CAknInformationNote* informationNote = new (ELeave) CAknInformationNote;
        TBuf<30> msg;
        switch(value)
        {
            case EPSAutolockOff:
            {
```

Retrieve_General_phone_information_using_Shared_Data_API

```
msg.      Copy(_L("Auto Lock Off"));
informationNote    ->ExecuteLD(msg);
    break;
    }
    case EPSAutolockOn:
    {
msg.      Copy(_L("Auto Lock On"));
informationNote    ->ExecuteLD(msg);
    break;
    }
    }
}
}
```

SIM change information: Indicates if the current SIM card is the same as the previous one. It detects the SIM change every time the phone is switched ON.

Case1: Say SIM1 is present in phone - phone switched OFF - removed SIM1 - inserted SIM2 - phone switched ON - Running the application notifies "SIM Changed".

Case2: Say SIM1 is present in phone - phone switched OFF - Again phone switched ON - Running the application notifies "SIM not Changed".

Case3: Say SIM1 is present in phone - phone switched OFF - removed SIM1 - inserted SIM2 - phone switched ON - Again phone switched OFF - phone switched ON - Now running the application notifies "SIM not Changed".

The following code displays SIM change information:

```
void CPSVariablesAppUi::IsSIMchanged()
{
    TInt value;
    TInt ret=RProperty::Get(KUidSystemCategory, KPSUidSimChangedValue, value);
    if(KErrNone==ret)
    {
        CAknInformationNote* informationNote = new (ELeave) CAknInformationNote;
        TBuf<30> msg;
        switch(value)
        {
            case EPSSimNotChanged:
            {
                msg.Copy(_L("Sim Not Changed"));
                informationNote->ExecuteLD(msg);
                break;
            }
            case EPSSimChanged:
            {
                msg.      Copy(_L("Sim Changed"));
                informationNote->ExecuteLD(msg);
                break;
            }
        }
    }
}
```

SIM status information: Indicates current status of SIM card.

Case1: Say SIM card is present in phone and is ok.

Example code

Retrieve_General_phone_information_using_Shared_Data_API

Case2: Say SIM card is not present in phone.

Case3: Say SIM card is present in phone but rejected by operator.

The following code displays SIM status information:

```
void CPSVariablesAppUi::SimStatus()
{
    TInt value;
    TInt ret=RProperty::Get(KUidSystemCategory, KUidSIMStatus, value);
    if(KErrNone==ret)
    {
        CAknInformationNote* informationNote = new (ELeave) CAknInformationNote;
        TBuf<70> msg;
        switch(value)
        {
            case ESASimOk:
            {
                msg.Copy(_L("Sim present in phone and it is ok"));
                informationNote->ExecuteLD(msg);
                break;
            }
            case ESASimNotPresent:
            {
                msg. Copy(_L("Sim not present in phone"));
                informationNote->ExecuteLD(msg);
                break;
            }
            case ESASimRejected:
            {
                msg. Copy(_L("Sim present in phone but rejected by operator"));
                informationNote->ExecuteLD(msg);
                break;
            }
        }
    }
}
```

Network Strength:The following code displays Network strength in bars (1 to 8):

```
void CPSVariablesAppUi::GetNetworkStrength()
{
    TInt value;
    TInt ret=RProperty::Get(KUidSystemCategory, KPSUidNetworkBarsValue, value);
    if(KErrNone==ret)
    {
        <50>TBuf;
        AppendMsg(value);
        CEi3baEmv()->InfoWinL(_L("Network Strength in bars: "),msg);
    }
}
```

It also provides the keys for getting following information:

1) Status of GPRS connection: Gprs suspended, context active etc.

2) Status of IRDA connection: IRDA Irlap layer loaded, connected, blocked, etc.

- 3) **Silent Mode:** profile silent Activated or not.
- 4) **Network Status:** Availability YES/NO.
- 5) **Network Strength:** Low, Medium, High, etc.
- 6) **SIM Inbox/Outbox:** Empty or Documents inside.
- 7) **Calls forwarding status:** All calls, No calls, Only on Line1, etc.
- 8) **Sim Sms Memory Status:** Sim Sms Memory Full or Not Full.
- 9) **Sim Ready Status:** Indicates if the SIM card is ready to send SIM card contacts information or not.
- 10) **SIM lock status:** Sim Lock Active, Restriction ON, etc.

.....

Example project

The following sample application is tested in Nokia N71.

[Example application](#)

Related Link:

- [Get SIM status in S60 2nd Edition](#)
- [Discussion in forum](#)