ID	CS000818	**Creation date**	February 11, 2008
Platform	S60 3rd Edition, FP2	**Tested on devices**	
Category	Symbian C++	**Subcategory**	

Keywords (APIs, classes, methods, functions): KCoreAppUIsNewEmailStatus

Note!

This API is not part of the public SDK. It can be found in the SDK API Plug-in.

Purpose

The **KCoreAppUIsNewEmailStatus** API is used for getting a notification of a new e-mail. The header file also contains the keys to set & get notifications for various indicators, such as POC indicator, USB indicator, and mobile TV recording.

This functionality is accomplished by listening to changes in Publish & Subscribe key values.

Note that while e-mail status P&S keys apply across all S60 3rd Edition releases (Feature Packs), the majority of the keys are introduced in Extensions plug-in package for S60 3rd Edition SDK for Symbian OS, for C++, supporting Feature Pack 2.

Use cases

1. An application needs to be notified of a change in the status of an e-mail. In that case you can use this API and can accordingly perform a particular operation based on it.

2. This API can also be used to set the status of various indicators.

Example code

Header files:

```
#include <coreapplicationuisdomainpskeys.h>
 #include <e32property.h>       //RProperty class
```

Link against:

```
LIBRARY   euser.lib//RProperty class
```

Capabilities:

```
WriteDeviceData  // for setting the e-mail status using RProperty::Set()
 ReadDeviceData   // for retrieving the e-mail status using RProperty::Get()
```

Get e-mail status:

The following code is used for retrieving the e-mail status:

```
RProperty iProperty;
 TInt iEmailStatus;
 // This is used for retrieving the e-mail status and the
 // result is stored in iEmailStatus
 iProperty.Get( KPSUidCoreApplicationUIs,
                KCoreAppUIsNewEmailStatus,
                iEmailStatus
              );
 // Compare it with TCoreAppUIsNewEmailStatus enum
 // for checking the e-mail status
 switch( iEmailStatus )
  {
  case ECoreAppUIsNoNewEmail:
       CEikonEnv::Static()->AlertWin(_L("No New e-mail"));
       break;
  case ECoreAppUIsNewEmail:
       CEikonEnv::Static()->AlertWin(_L("You have a New e-mail"));
       break;
  default:
       CEikonEnv::Static()->AlertWin(_L("Uninitialised"));
       break;
  }
```

Set e-mail status:

The following code is used to assig a value to the e-mail status: The 3rd parameter can take any values of the TCoreAppUIsNewEmailStatus enum defined in the header file.

Setting "ECoreAppUIsNewEmailStatusUninitialized" to the e-mail status:

```
iProperty.Set( KPSUidCoreApplicationUIs,
               KCoreAppUIsNewEmailStatus,
               ECoreAppUIsNewEmailStatusUninitialized
             );
```

Notification of change in e-mail status:

The following is the code snippet for getting a notification whenever there is a change in the e-mail status:

```
RProperty iProperty;
 // This is based on Active Objects.
 // The function creates a handle (this object) to the specified property.
 // This allows the caller to subscribe for a notification of changes
 // to this e-mail status.
 iProperty.Attach(KPSUidCoreApplicationUIs, KCoreAppUIsNewEmailStatus);
 // The function issues an asynchronous request to be notified when the
 // e-mail status is changed.
 iProperty.Subscribe(iStatus);
 SetActive();
 // RunL is called whenever the status changes.
```

Get Mobile TV Recording status: The following code is used for retrieving the status of Mobile TV Recording:

```
TInt status;
 RProperty::Get(KPSUidCoreApplicationUIs, KCoreAppUIsMtvRecStatus, status);
 switch(status)
   {
   case ECoreAppUIsMtvRecStatusUninitialized:
     // Uninitialized
     break;

   case ECoreAppUIsMtvRecStatusOff:
     // Off
     break;

   case ECoreAppUIsMtvRecStatusOn:
     // On
     break;
   }
```

Set Mobile TV Recording status:

The following code is used to assign a value to the mobile TV recording status:

```
TInt status;
 TInt err;

 RProperty::Get(KPSUidCoreApplicationUIs, KCoreAppUIsMtvRecStatus, status);

 if(status == ECoreAppUIsMtvRecStatusOff)
   {
   err = RProperty::Set( KPSUidCoreApplicationUIs,
                         KCoreAppUIsMtvRecStatus,
                         ECoreAppUIsMtvRecStatusOn );
   }
 else
   {
   err = RProperty::Set( KPSUidCoreApplicationUIs,
                         KCoreAppUIsMtvRecStatus,
                         ECoreAppUIsMtvRecStatusOff );
   }
```

The other indicator values, such as POC indicator and USB indicator, can be retrieved and set in the same way as above with the keys defined in the `coreapplicationuisdomainpskeys.h` header file.

Example application

ApplicationUIKeys