Creating a recognizer plug-in for a MIME type on Symbian

Overview

It is sometimes necessary to launch a handler application (such as a viewer) for files selected, for example, through the File manager. However, if the MIME type of the file is not known, the handler application cannot be launched. In this case, you need to write a data recognizer. Recognizers are ECOM plug-ins that try to identify the type of data by its URI (filename etc.) and/or its contents. This code snippet demonstrates how to write a recognizer for TeX files (MIME type: `application/x-tex`).

Note: Data recognizers do not handle data, they just try to identify its type. Once the type has been identified, the data can be passed to the application that best handles it. For more information about implementing a data handler, see Creating a handler application for a MIME type using Symbian C++.

MMP file

The following capabilities and libraries are required:

```
CAPABILITY ProtServ
```

```
LIBRARY apmime.lib  // CApaDataRecognizerType
LIBRARY efsrv.lib   // TParse
LIBRARY euser.lib
```

Here is the complete MMP file:

```
TARGET RecognizerPlugin.dll
TARGETTYPE PLUGIN
UID 0x10009D8D 0x101FF1EC

CAPABILITY ProtServ
VENDORID 0

SOURCEPATH      ..\data
START RESOURCE 101FF1EC.rss
    TARGET RecognizerPlugin.rsc
END

USERINCLUDE   ..\inc
SYSTEMINCLUDE \epoc32\include
SYSTEMINCLUDE \epoc32\include\ecom

SOURCEPATH    ..\src
SOURCE CTexRecognizer.cpp

LIBRARY apmime.lib  // CApaDataRecognizerType
LIBRARY efsrv.lib   // TParse
LIBRARY euser.lib
```

PKG file

```
; Language - standard language definitions
&EN

; SIS file header
#{"RecognizerPlugin DLL"},(0x10280805),1,0,0

; Localized Vendor name
%{"Vendor-EN"}

; Unique Vendor Name
:"Vendor"

; Supports S60 v 3.0
[0x101F7961], 0, 0, 0, {"Series60ProductID"}

; Files to install
"\Symbian\9.1\S60_3rd_MR\Epoc32\release\GCCE\UREL\RecognizerPlugin.dll"
    - "!:\sys\bin\RecognizerPlugin.dll"
"\Symbian\9.1\S60_3rd_MR\Epoc32\Data\z\resource\plugins\RecognizerPlugin.rsc"
    - "!:\resource\plugins\RecognizerPlugin.rsc"
```

Resource file: 101FF1EC.rss

Note: The name of this file must correspond to the UID3 of the application.

```
#include <RegistryInfo.rh>

RESOURCE REGISTRY_INFO r_registry
    {
    // UID for the DLL
```


(C) Copyright Nokia 2013. All rights reserved.

```
    dll_uid = 0x101FF1EC;
    interfaces =
        {
        INTERFACE_INFO
            {
            // Common for all data recognizers
            interface_uid = 0x101F7D87;
            implementations =
                {
                IMPLEMENTATION_INFO
                    {
                    implementation_uid = 0x101FF1ED;
                    version_no = 1;
                    display_name = "RecognizerPlugin";
                    default_data = "RecognizerPlugin";
                    opaque_data = "";
                    }
                };
            }
        };
    }
```

Header file

```
#ifndef __CTEXRECOGNIZER_H_
#define __CTEXRECOGNIZER_H_

#include <apmrec.h>  // CApaDataRecognizerType

class CTexRecognizer : public CApaDataRecognizerType
    {
    public:  // Constructors and destructor
        CTexRecognizer();
        static CApaDataRecognizerType* CreateRecognizerL();
        virtual ~CTexRecognizer();

    public:  // Methods from base classes
        TUint PreferredBufSize();
        TDataType SupportedDataTypeL(TInt aIndex) const;

    private:  // Methods from base classes
        void DoRecognizeL(const TDesC& aName, const TDesC8& aBuffer);
    };

#endif /*__CTEXRECOGNIZER_H_*/
```

Source file

```
#include <apmrec.h>   // CApaDataRecognizerType
#include <f32file.h>  // TParse
#include <ImplementationProxy.h>  // TImplementationProxy

#include "CTexRecognizer.h"

const TUid KRecognizerUid = {0x101FF1EC};
const TInt KRecognizerImplUid = {0x101FF1ED};
const TInt KMaxBufferSize = 128;
_LIT8(KMIMETex, "application/x-tex");
_LIT(KTexExtension, ".tex");

CTexRecognizer::CTexRecognizer()
    : CApaDataRecognizerType(KRecognizerUid, CApaDataRecognizerType::ENormal)
    {
    iCountDataTypes = 1;
    }

CTexRecognizer::~CTexRecognizer()
    {
    // No implementation required
    }

/**
 * From CApaDataRecognizerType.
 */
TUint CTexRecognizer::PreferredBufSize()
    {
    return KMaxBufferSize;
    }

/**
 * From CApaDataRecognizerType.
 * Gets one of the data types that this recognizer can recognize.
 */
TDataType CTexRecognizer::SupportedDataTypeL(TInt /*aIndex*/) const
    {
    return TDataType(KMIMETex);
    }

/**
 * From CApaDataRecognizerType.
 * Attempts to recognize data denoted by aName and residing in aBuffer.
 */
void CTexRecognizer::DoRecognizeL(const TDesC& aName, const TDesC8& aBuffer)
    {
    // To keep the code simple, we only check the file name extension: If the
    // extension is ".tex", we are "ECertain" that the file is of type
    // application/x-tex.
    TParse parse;
    parse.Set(aName, NULL, NULL);
    TPtrC ext = parse.Ext();
    if (ext.CompareF(KTexExtension) == 0)
        {
        iConfidence = ECertain;
        iDataType = TDataType(KMIMETex);
        }
    }

// *** ECOM framework code ***

CApaDataRecognizerType* CTexRecognizer::CreateRecognizerL()
```


(C) Copyright Nokia 2013. All rights reserved.

```
    {
    return new (ELeave) CTexRecognizer();
    }

const TImplementationProxy ImplementationTable[] =
    {
    IMPLEMENTATION_PROXY_ENTRY(KRecognizerImplUid,
            CTexRecognizer::CreateRecognizerL)
    };

EXPORT_C const TImplementationProxy* ImplementationGroupProxy(
        TInt& aTableCount)
    {
    aTableCount = sizeof(ImplementationTable) / sizeof(TImplementationProxy);
    return ImplementationTable;
    }
```

Postconditions

A recognizer for TeX files is implemented. If the file of type `application/x-tex` is opened, for example, from the File manager, the `CTexRecognizer::DoRecognizeL()` method is called and the MIME type is recognized to be `application/x-tex`.

See also

- Creating a handler application for a MIME type using Symbian C++.
- Symbian MIME recognizers and opening files for editing.
- S60 Platform: Document Handler Example.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.