

Recognizing_file_types

The following function shows how to determine file types.

```
void GetFileType(const TDesC& aFileName, TDes& aFileType)
{
    TEntry FileEntry;

    if (CCoeEnv::Static()->FsSession().Entry(aFileName,FileEntry) == KErrNone)
    {
        TBuf8<255> FileBuffer;

        if(!FileEntry.IsDir())
        {
            TInt FileSize = FileEntry.iSize;

            if(FileSize > 255)
            {
                FileSize = 255;
            }

            if(CCoeEnv::Static()->FsSession().ReadFileSection
                (aFileName,0,FileBuffer,FileSize) == KErrNone)
            {
                RApaLsSession RSession;
                if(RSession.Connect() == KErrNone)
                {
                    TDataRecognitionResult FileDataType;

                    RSession.RecognizeData(aFileName,FileBuffer,*&FileDataType);
                    if(FileDataType.iConfidence > aResult.iConfidence)
                    {
                        aFileType.Copy(FileDataType.iDataType.Des());
                    }

                    RSession.Close();
                }
            }
        }
    }
}
```

If you don't have access to the file itself (for example because of the restrictions posed by the platform security) you can use this function:

```
void GetFileType(RFile& aFile, TDes& aFileType)
{
    TInt FileSize(0);
    TFileName RFileName;

    if(aFile.Size(FileSize)== KErrNone && FullName(RFileName) == KErrNone)
    {
        TBuf8<255> FileBuffer;

        if(FileSize > 255)
        {
            FileSize = 255;
        }

        if(aFile.Read(FileBuffer,FileSize) == KErrNone)
        {
            RApaLsSession RSession;
            if(RSession.Connect() == KErrNone)
```

Recognizing_file_types

```
{
  TDataRecognitionResult FileDataType;

  RSession.RecognizeData(RFileName,FileBuffer,&FileDataType);
  if(FileDataType.iConfidence > aResult.iConfidence)
  {
    aFileType.Copy(FileDataType.iDataType.Des());
  }

  RSession.Close();
}
}
```

In both cases the type of the file is returned in the *aFileName* argument variable as standard MIME type (for example image/gif for gif images etc) and in case the file is not recognized the *aFileName* will not be modified.