How to Create a Direct File Store

```
#include <s32file.h>
Link Againts : estor.lib
```

Creating a Persistent Direct File Store

```
// The call to ReplaceLC() will create the file if it does not exist, otherwise
//it will replace any existing file.

CFileStore* store = CDirectFileStore::ReplaceLC(aFs,aFileName,EFileWrite);

//set the store's type

store->SetTypeL(TUidType(KDirectFileStoreLayoutUid,KUidAppDllDoc,aAppUid));

// Create a stream dictionarary

CStreamDictionary* dictionary = CStreamDictionary::NewLC();

// We then Create, write and close a stream

RStoreWriteStream stream;

TStreamId id = stream.CreateLC(*store);

TInt16 i = 0x1234;

stream << i;

stream.CommitL();

CleanupStack::PopAndDestroy(); // stream

// you must use an instance of RStoreWriteStream, whose CreateL() and
// CreateLC() functions return a TStreamId. Once writing the stream is
// complete, you need to add an entry to the stream dictionary, making an
// association between the stream ID and an externally known UID:


dictionary->AssignL(aAppUid,id);

// create a stream to contain the stream dictionary,
// and mark it as being the root stream

RStoreWriteStream rootStream;

TStreamId rootId = rootStream.CreateLC(*store);

rootStream << *dictionary;

// commit all the changes made to the store and then to free its resource
// rootStream.CommitL();


CleanupStack::PopAndDestroy(); // rootStream

CleanupStack::PopAndDestroy(); // dictionary

store->SetRootL(rootId);

store->CommitL();

CleanupStack::PopAndDestroy(); // store
```

Reading a Persistent Store

```
CFileStore* store = CDirectFileStore::OpenLC(aFs,aFileName,EFileRead);

CStreamDictionary* dictionary = CStreamDictionary::NewLC();

RStoreReadStream rootStream;

rootStream.OpenLC(*store, store->Root());

rootStream >> *dictionary;

CleanupStack::PopAndDestroy(); // rootStream

TStreamId id = dictionary->At(aAppUid);

CleanupStack::PopAndDestroy(); // dictionary

RStoreReadStream stream;

stream.OpenLC(*store, id);

TInt16 j;

stream >> j;

CleanupStack::PopAndDestroy(); // stream

CleanupStack::PopAndDestroy(); // store
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.