Using files with S60 and Maemo Platforms

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article is believed to be still valid for the original topic scope.

Comparison

```
RFile file;
GnomeVFSHandle *handle = NULL;
```

```
User::LeaveIfError(file.Open(rfs, aFileName, EFileRead));
vfs_result = gnome_vfs_open(&handle, file_name, GNOME_VFS_OPEN_READ);
```

```
HBufC8* buf = HBufC8::NewLC(size);

temp_buffer = g_malloc(finfo.size + 1);
memset(temp_buffer, 0, finfo.size + 1);
```

```
User::LeaveIfError(file.Read(bufPtr));
gnome_vfs_read(handle, temp_buffer, finfo.size, &in_bytes);
```

```
CleanupStack::PopAndDestroy(2); // file, rfs
gnome_vfs_close(handle);
```

Comparing S60 and Maemo Platforms

S60 Platform

```
// ----------------------------------------------------
// CBrCtlSampleAppContainer::ReadFileLC(const TDesC& aFileName)
// ----------------------------------------------------
//
HBufC8* CBrCtlSampleAppContainer::ReadFileLC(const TDesC& aFileName)
    {
    RFs rfs;
    RFile file;
    User::LeaveIfError(rfs.Connect());
    CleanupClosePushL(rfs);
    User::LeaveIfError(file.Open(rfs, aFileName, EFileRead));
    CleanupClosePushL(file);
    TInt size;
    User::LeaveIfError(file.Size(size));
    HBufC8* buf = HBufC8::NewLC(size);
    TPtr8 bufPtr(buf->Des());
    User::LeaveIfError(file.Read(bufPtr));
    CleanupStack::Pop(); // buf
    CleanupStack::PopAndDestroy(2); // file, rfs
    CleanupStack::PushL(buf);
    return buf;
    }
```

Maemo Platform

```
/* read file */
gchar* read_file_to_buffer( MainView* main, gchar* file_name )
{
    GnomeVFSResult vfs_result;
    GnomeVFSHandle *handle = NULL;
    GnomeVFSFileSize in_bytes;
    GnomeVFSFileInfo finfo;

    gchar *temp_buffer = NULL;

    /* try to get file info */
    vfs_result = gnome_vfs_get_file_info(file_name, &finfo, GNOME_VFS_FILE_INFO_DEFAULT);
    if ( vfs_result != GNOME_VFS_OK ) {
        interface_error( UTFVIEWER_ERROR_OPEN_FAILED, main );
        return NULL;
    }

    /* try to create handle to file */
    vfs_result = gnome_vfs_open(&handle, file_name, GNOME_VFS_OPEN_READ);
    if ( vfs_result != GNOME_VFS_OK ) {
        interface_error( UTFVIEWER_ERROR_OPEN_FAILED, main );
        return NULL;
    }

    /* allocate memory for temp_buffer */
    temp_buffer = g_malloc(finfo.size + 1);
    memset(temp_buffer, 0, finfo.size + 1);

    /* read from file to buffer */
```


(C) Copyright Nokia 2013. All rights reserved.

```
    gnome_vfs_read(handle, temp_buffer, finfo.size, &in_bytes);

    /* close file */
    gnome_vfs_close(handle);

    return temp_buffer;
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.