

# Archived:How to retrieve image thumbnails on S60 3rd Edition



**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}}`.

## Overview

This article demonstrates how to retrieve image thumbnails on S60 3rd Edition.

## Description

In S60 2nd Edition, CAPAIImageUtil class provides GetThumbnailL() method to get thumbnails for images saved in the Gallery. In S60 3rd Edition, the Photo Album API has been removed and is replaced by the Media Gallery API, which does not provide any method for directly accessing image thumbnails.

## Solution

The directory path for thumbnails is in the following format:

```
<DRIVE>:\ Data \ Images \ _PALbTN \
```

In Nokia Nseries devices, the directory structure used for Gallery is different from other devices:

```
<DRIVE>:\ Data \ Images \ <YearMonth> \ _PALbTN \ <R1XR2> \
```

where R1 and R2 are device-specific resolutions. There may be multiple versions (resolutions) of the same thumbnail.

Note: The \_PALbTN directory is marked as hidden. To make the code work in all S60 3rd Edition devices, you have to do a recursive search in all the subfolders of the image folder to find the thumbnail of the image. Usually, thumbnails will use the same file name as the main image file, but a different extension. For example, if the image file name is some\_picture.jpg, then the thumbnail will be named some\_picture.jpg\_170x128. (Simplified) code for recursive image/thumbnail search:

```
RFs fs;
User::LeaveIfError(fs.Connect());
CleanupClosePushL(fs);
// assumes phone memory is used for storing images
_LIT( KImagesPath, "C:\\Data\\Images\\" );
// to list known image file(s) and thumbnail(s), use imagename.ext*
// to list thumbnail(s) only for a known image, use imagename.ext_*
// to list all available images and thumbnails, use *.*
_LIT(KImageName, "some_picture.jpg*");
CDirScan* scan = CDirScan::NewLC( fs );
scan->SetScanDataL( KImagesPath,
                  KEntryAttNormal|KEntryAttHidden,
                  ESortNone,
                  CDirScan::EScanDownTree );

FOREVER
{
    CDir* dirlis = 0;
    TRAPD(error, scan->NextL(dirlis));
    if (error | !dirlis)
    {
        break;
    }
    delete dirlis;
    FindFile( fs, scan->FullPath(), KImageName );
};
CleanupStack::PopAndDestroy(2); // scan, fs
void FindFile( RFs& aFs, const TDesC& aDir, const TDesC& aImageWild )
{
    CDir* dirList;
    TFindFile FindObj( aFs );
    TInt j = FindObj.FindWildByDir( aImageWild, aDir, dirList );
    if( j == KErrNone )
    {
        for ( TInt i = 0; i < dirList->Count() ; i++ )
        {
            TFileName fullname(aDir);
            fullname.Append((*dirList)[i].iName);
            // fullname now contains path and name of image/thumbnail
        }
    }
}
```

