

E32Image file format on Symbian OS 9

This article explains briefly the [E32Image](#) file format on [Symbian OS v9](#) or newer. Please check some links at the end of this article for more detailed information.

Introduction

The executable files on [Symbian OS v9](#), i.e., `.exe` or `.dll`, use a special format, called E32Image. It is based on ELF file format. It is different on pre-Symbian OS v9. The E32Image on pre-Symbian OS v9 is based on PE file format.

E32Image added some [Symbian](#)-specific information to the header. It also replaces function names with ordinal numbers. The tool used to translate ELF into E32Image is called `elftran` and `elf2e32`.

The difference between `elftran` and `elf2e32` is the input file. The `elftran` tool translates ELF file in ABLv1 format into E32Image. The `elf232` translates ELF file in newer format, ABLv2, into E32Image.

Like standard ELF executable file, E32Image contains of several sections, i.e.:

- **Code section.** It contains three parts: text section, export address table and import address table.
- **BSS section.** It contains un-initialized data.
- **Data section.** It contains initialized data.
- **Import section.** It contains the information about all imported functions used by the code.
- **Relocation section.** It contains relocation table needed by [Symbian OS](#) loader to load the code.

Header information

The header information of E32Image is defined in a header file called `f32image.h`. It is part of the public [Symbian SDKs](#) located at `\epoc32\include`. The declaration of the header is splitted into three parts.

The first part, `E32ImageHeader`, is the main part. It is quite similar to the header on [pre-Symbian OS 9](#).

```
class E32ImageHeader
{
public:
    TUint32 iUid1;
    TUint32 iUid2;
    TUint32 iUid3;
    TUint32 iUidChecksum;
    TUint iSignature;
    TUint32 iHeaderCrc;
    TUint32 iModuleVersion;
    TUint32 iCompressionType;
    TVersion iToolsVersion;
    TUint32 iTimeLo;
    TUint32 iTimeHi;
    TUint iFlags;
    TInt iCodeSize;
    TInt iDataSize;
    TInt iHeapSizeMin;
    TInt iHeapSizeMax;
    TInt iStackSize;
    TInt iBssSize;
    TUint iEntryPoint;
    TUint iCodeBase;
    TUint iDataBase;
    TInt iDllRefTableCount;
    TUint iExportDirOffset;
    TInt iExportDirCount;
    TInt iTextSize;
    TUint iCodeOffset;
    TUint iDataOffset;
    TUint iImportOffset;
    TUint iCodeRelocOffset;
    TUint iDataRelocOffset;
    TUint16 iProcessPriority;
    TUint16 iCpuIdentifier;
};
```

The second part, `E32ImageHeaderComp` is needed when the executable file is compressed.

```
class E32ImageHeaderComp : public E32ImageHeader
{
public:
    TUint32 iUncompressedSize;
};
```

The third part, `E32ImageHeaderV` contains some extra information.

```
class E32ImageHeaderV : public E32ImageHeaderComp
{
public:
```

```

SSecurityInfo iS;
TUint32 iExceptionDescriptor;
TUint32 iSpare2;
TUint16 iExportDescSize;
TUint8 iExportDescType;
TUint8 iExportDesc[1];
};

```

E32ImageHeader

- **iUid1**, **iUid2**, **iUid3**, are the first, second and third **UIDs** of the executable file respectively.
- **iUidChecksum**, is the checksum of the first three **UIDs**.
- **iSignature**, is a unique signature of E32 file, always has a value of 'EPOC'.
- **iHeaderCrc**, is the checksum of entire header, calculated with CCITT CRC-32 algorithm.
- **iModuleVersion**, is the version number for this executable.
- **iCompressionType**, is the **UID** of the library used to compress the executable file. It is 0 if the executable is not compressed or **KUidCompressionDeflate** (=0x101F 7AFC) if it is compressed with Deflate algorithm.
- **iToolsVersion**, is the version of **ELFTRAN** that generates this file.
- **iTimeLo**, **iTimeHi**, is the lowest and highest word of the timestamp when the file is created respectively.
- **iFlags**, flags for this executable. There are some flags defined in **f32image.h**, for example, **KImageDll**, **KImageNoCallEntryPoint**, etc.
- **iCodeSize**, is the size of code section, import address table, constant data and export dir.
- **iDataSize**, is size of initialized data.
- **iHeapSizeMin**, is the minimum size of the heap.
- **iHeapSizeMax**, is the maximum size of the heap.
- **iStackSize**, is the size of the stack.
- **iBssSize**, is the size of the un-initialized data section.
- **iEntryPoint**, is the offset into code of entry point.
- **iCodeBase**, where the code is linked for.
- **iDataBase**, where the data is linked for.
- **iDllRefTableCount**, is the number of DLLs imported by this program.
- **iExportDirOffset**, is the offset into the file of the export address table.
- **iExportDirCount**, is the offset of the export address table.
- **iTextSize**, is size of just the text section.
- **iCodeOffset**, is the file offset to code section.
- **iDataOffset**, is the file offset to data section.
- **iImportOffset**, is the file offset to import section.
- **iCodeRelocOffset**, is the relocations for code and constants.
- **iDataRelocOffset**, is the relocations for data.
- **iProcessPriority**, is the executables priority.
- **iCpuIdentifier**, is the identifier of CPU.

E32ImageHeaderComp

- **iUncompressedSize**, the uncompressed size of file. Remember that E32Image file might be compressed.

E32ImageHeaderV

- **iS.iSecureId**, is **secure ID** of the executable. It is usually the same as the **UID3** of the executable.
- **iS.iVendorId**, is **vendor ID** of executable. For third party application, it is usually 0.
- **iS.iCaps.iCaps**, is **capabilities** needed to run the executable. The definition of all **Symbian OS capabilities** can be found at **e32capability.h**.
- **iExceptionDescriptor**, is the offset in bytes from start of code section to Exception Descriptor, bit 0 set if valid
- **iSpare2**, reserved.
- **iExportDescSize**, is the size of bitmap section.
- **iExportDescType[1]**, is type of description of holes in export table.
- **iExportDesc[1]**, is the description of holes in export table.

Internal links

- [Application Binary Interface](#)
- [Introduction to E32Image](#)
- [E32Image file format on pre-Symbian OS 9](#)

External links

- [Symbian OS Executable File Format \(E32Image\)](#)
- [New Symbian OS 9 Executable File Format \(E32Image\)](#)
- [Portable Executable on Wikipedia](#)

