

PalmOS Public Files

A proposed standard for interchangeable files under PalmOS

by Vince Lee, TealPoint Software

July 29, 2003

As the PalmOS platform continues to mature, there is an increasing need for standards to facilitate the interchange of images, text, and other data between applications made by different developers, and standardize storing of such files in RAM and external storage cards. The following is a proposal for a simple standard for creating and using universally interchangeable files. The objectives of this proposal are as follows:

- Promote transparent interchangeability of files between applications
- Promote transparent interchangeability of files between desktop PCs and PalmOS handhelds
- Allow transparent transport of files between RAM and external VFS cards

1) PalmOS-format files in RAM

The primary issue concerning PalmOS-format files in memory is the necessity for a unique CreatorID for each file. Some defacto standards on the platform such as Doc files, have tied the file format to the CreatorID of a specific (and now defunct) application. This not only creates an advantage for a single manufacturer (thereby discouraging others from adopting the format), but creates confusion amongst customers, who may inadvertently delete all files of a given CreatorID when deleting the application which has the same CreatorID.

CreatorID

To alleviate these problems, I propose we adopt the standard CreatorID 'PUBL' for all public shared files, one for which no application will ever be created. Accordingly I have reserved this CreatorID in the online registration database, and would like to assign it to PalmSource should this standard be adopted.

TypeID

To make sure file types do not conflict, the TypeID of any new shared files should be set equal to a CreatorID registered with PalmSource. This can be the CreatorID of an actual application owned by the developer proposing the standard, or simply a generic file type registered in the CreatorID database to make it unique.

2) PC-format streamed files in RAM

Images, sounds, text, and music make up much of the files that would benefit from a shared file format. Defacto file standards such as Jpeg, Wave, RTF, and Midi already exist for these types of files on the desktop, so it is not surprising that many developers are using PalmOS "steamed" files and adopting these formats in their own applications. This not only takes advantage of existing libraries and open-source code, but avoids the nightmare of creating a completely separate set of file standards and bi-directional converters for PalmOS, which would isolate the platform to the detriment of us all.

And yet, two small details currently keep applications from sharing such files: Type and CreatorID. Indeed, many image viewing programs share compatible Jpeg-based file formats but cannot share data simply because they use different Type and CreatorID values.

Standard Files

I propose that all standard streamed files use the 'PUBL' public CreatorID, and adopt a TypeID based on the file extension of the corresponding PC file in all caps. For instance, Jpeg-format streamed files would have a CreatorID of 'PUBL' and a TypeID of '.JPG'. Plain text files would have a CreatorID of 'PUBL' and a TypeID of '.TXT'. Files with extensions shorter than 3 letters would be padded with spaces, so C-language files would have a TypeID of '.C '.

3) Private streamed files on VFS

The last proposed standard for file handling is a relatively minor addition, but one still worth mentioning. When copying PalmOS streamed files to a VFS device, one can either convert the file to its PRC/PDB format, or leave it as a linear file stored in VFS. The former case is well known and can be handled with no loss of information. When "linearizing" the file is desired, however, the Type and Creator are lost, leaving file utilities unable to restore the file completely when copying back to RAM.

I propose tacking on the file CreatorID and TypeID, separated by underscores, to the end of non-public streamed files when copying them in linear format. Thus, a photo viewer which created a non-public (for some reason) linear file mypic.jpg with Type 'Phot' and Creator 'PSee' would represent the same file on VFS as 'mypic_PSee_Phot.jpg'. Third party file moving programs could recognize this format when copying the files back and forth so they would continue to work with the owning application. Placing the CreatorID before the Type is consistent with the formatting of some system files.

4) Application Implementation

The above proposed standards are currently supported in our application TealMover 1.5, which is a utility for moving files to/from VFS cards, and in TealDoc 6 when reading or manipulation plain text files. To implement similar behavior in an application, we suggest the following:

- When creating a PalmOS-format database in RAM that can be shared by applications, set the CreatorID to 'PUBL' and the Type to a unique value registered in the PalmSource CreatorID database.
- When representing a standard PC-format linear file in RAM, create a PalmOS streamed file and set the CreatorID to 'PUBL' and the TypeID to '.JPG', '.TXT', '.GIF', etc.
- When representing a non-public streamed file on a VFS device in its linear (non-PDB) format, tack on the CreatorID and Type to the end of the file name before the file extension, if any (myname_CrID_Type.ext).
- When moving a linear file from VFS to RAM that has the above naming convention, create a streamed file using the imbedded CreatorID and TypeID values, removing them from the filename in the process.

my \$.02.

-Vince