Program Manager Group File Format

General Layout

A program manager group consists of a header and a variable size data space.

Program Group Header

The header identifies the group and specifies some global information. The structure is,

```
struct tagGROUPHEADER {
        char identifier[4];
        WORD wCheckSum;
        WORD cbGroup;
        WORD nCmdShow;
        RECT rcNormal;
        POINT ptMin;
        WORD pName;
        WORD wLogPixelsX;
        WORD wLogPixelsY;
        WORD wBitsPerPixel;
        WORD wPlanes;
        WORD cItems;
        WORD rgiItems[cItems];
  };
```

The *identifier* field contains the string 'PMCC'. If this string is not present in the first four bytes of a file, that file is not a valid group file. The *wCheckSum* field is used as an additional check on the validity of the file: the sum of each word in the file (in other words, the word at offsets 0, 2, 4, etc) must be 0. When writing a group file, *wCheckSum* should be set accordingly.

The *cbGroup* field is the size of the group file. If the file is larger than *cbGroup* when reading a file, the extra information should be ignored. When writing a file, the size of the file should be *cbGroup*.

The *nCmdShow* field contains a **ShowWindow**() index which specifies whether the group will be minimized (SW_SHOWMINIMIZE) normal (SW_SHOWNORMAL), or maximized (SW_SHOWMAXIMIZE). The point *ptMin* specifies the point at which the icon will be placed when the group is minimized. The *rcNormal* rectangle specifies the window rectangle when the window is in the normal position (not minimized or maximized). Coordinates are relative the Program Manager's MDI Client window.

The *pName* field is the offset in the file to the zero terminated title of the group. This title appears in the group's caption.

The *wLogPixelsX, wLogPixelsY, wBitsPerPixel,* and *wPlanes* fields contain the corresponding values returned from **GetDeviceCaps**() for a screen (or window) DC at the time the icons in this group file were extracted. If the values do not match the current display when a group is loaded, Program Manager will reextract the icons from the .EXE files.

The *cItems* field contains the size of the *rgiItems* field. Each entry in *rgiItems*, if nonzero, points to an item structure. Note that *cItems* is not the number of items in the group, since there may be NULL entries in *rgiItems*.

The first item in *rgiItems* will be the initially active item when the group is loaded.

Item Data

Item data may appear anywhere after the group header up to the limit of *cbGroup*. The offsets in *rgiItems* point to the following structure:

```
struct tagITEMINFO {
        POINT pt;
        WORD iIcon;
        WORD cbHeader;
        WORD cbANDPlane;
        WORD cbXORPlane;
        WORD pHeader;
```

```
		WORD pANDPlane;
		WORD pXORPlane;
		WORD pName;
		WORD pCommand;
		WORD pIconPath;
	};
```

The *pt* field specifies the location of the item's icon within the group window. The fields *cbHeader* , *cbANDPlane*, and *cbXORPlane* specify the sizes of each of the three icon components. These three components are stored at the offsets specified by *pHeader, pANDPlane*, and *pXORPlane*. The name of the item, the command line that it executes, and the path to the file containing its icon are specified by *pName, pCommand* and *pIconPath* respectively. Each of these is the offset from the beginning of the file to a zero terminated string. The *iIcon* field selects the icon within the icon source file if there is more than one. It is zero for the first icon in the file. This index has nothing to do with the identifier of the icon resource; it is in the order icons appear in the .EXE file.

When adding an icon to a group, its data is appended to the end of the item information section and the appropriate pointers are added. When an item is deleted, information stored above it is moved down to shrink the size of the group file.

Maintaining Compatibility

If you are going to read and write group files you must maintain compatibility with existing and future versions of Program Manager by carefully adhering to a few rules. Your best bet is to use the Program Manager DDE interface whenever possible. This interface will allow you to add groups and items without dealing with the group file format. Remember to treat direct access to the group files as a hack of last resort. These files were not originally designed to be accessed by programs other than Program Manager.

If the magic string in the header is not correct, you should not load the file; it is not a group file. If the checksum is not correct, you should assume that the file is damaged. If the file is smaller than the header size or the value of *cbGroup*, the file is damaged. If the file is larger than *cbGroup* you should checksum the additional data if you perform the checksum, but otherwise ignore it.
Never write a group file that is larger than *cbGroup* bytes long.

Do not leave "unused" space between structures in the icon data area. This space will not be recovered and will go to waste. This includes space at the end of the file (but less than *cbGroup*).
Do not attempt to add structures and data for your own use, since Program Manager will not be able to maintain the association between that data and the item if the user edits or deletes the item. The data will end up getting lost and wasting space in the group file.

If the screen metric fields do not match the metrics of the current display driver, do not use any of the icon data. If you update the screen metrics, make sure all icons in the group are in the appropriate format. Icons added to the group must match the format specified by the screen metrics fields.

Do not update a group manually while Program Manager is running. To modify a group while Program Manager is running, use the DDE commands documented elsewhere. Updating groups while Program Manager is running can cause Program Manager to function erratically (you will crash Progman).