

Common Questions & Answers

Copyright (c), Freeman-Teresa Software, 1995

This is a "How-to" guide to Freeman Installer/Constructor. It covers most of the questions that you may have in using this product, ranging from the basics to the more advanced topics. For coverage of the most basic skills, please refer to the four tutorials.

Q1: How do I use Freeman without a C++ compiler?

1. Don't use the commands on the Compile menu. Use those on the Interpret menu instead. Functionally they are the same.
2. Don't insert C/C++ code.

Q2: How do I use Freeman without a resource compiler (brcc.32exe or rc.exe)?

If you have a resource editor (not resource compiler!) that can write into EXE files, then you can make your modifications to fimain.rc take effect by taking one of the following approaches:

- Use the resource editor to edit c:\prj\usr\fimaini.exe directly. Both Resource Workshop and VC++ IDE are capable of this.
- Use the resource editor to open c:\prj\usr\fimain.rc, edit it, and finally save it into c:\prj\usr\fimaini.exe. You can't use VC++ IDE to do that since it can't handle fimaini.exe that was made with Borland C++ 4.5.

Either of these does exactly the same as what a resource compiler does.

However, if you have neither resource editor nor resource compiler, then:

1. Confine yourself to the English language.
2. You can't use ad dialogs and bitmaps.

Q3: How do I set the text in the welcome dialog?

Go to the event list and select the "Welcome" event item.

Q4: How do I set the default target directory?

Go to the event list, select the "i = PromptString()" event item, and change the "default value".

Q5: I have hundreds of files that should go into, say \$i\netscape, do I have to set the target directory file by file?

No. The file dialog supports multiple selection. You can select all those files and then set the target directory all at once.

Q6: How do I use AppStudio or VC++ IDE to edit c:\prj\usr\fimain.rc?

AppStudio or VC++ IDE is NOT 100% compatible with rc.exe. This means that if you go straight and use either to edit fimain.rc, it will be rendered uncompileable by rc.exe. Fortunately this can be prevented by:

1. Choose Resource | Set Includes in AppStudio or VC++ IDE.
2. In the box labeled "Read-Only Symbol Directives", enter:

```
#include "stddef.rh"
```

You only need to do this once, i.e., the first time you edit fimain.rc.

Q7: How come the wrong icon is used?

Make sure the icon index is 0 (if you want the first icon).

Q8: How do I prevent the constructor from compressing my readme.txt file?

In the file dialog, change "prepare" from "compress" to "copy", and change "transfer" from "decompress" to "copy". Be careful to do these together!

```
Prepare: Copy
Create:   ???
Transfer: Copy
```

Q9: How do I use my own icon in the installer instead of the sunset icon?

If you're using the "Interpret" commands:

1. Interpret | Make
2. Modify c:\prj\usr\finstall.ico at wish.
3. Interpret | Bind Resources

If you're using the "Compile" commands: simply modify c:\prj\usr\finstall.ico at wish.

Q10: How do I say something other than "copying temporary files, please wait..." in the splash dialog?

Use a binary editor to edit c:\finst22\install.exe (don't forget to backup the original). Locate the original message by searching for keyword "copying" and then use your own message to overwrite the original. Make sure your own message must be shorter than the original and it must be 0-terminated. In the next major upgrade you should be able to enter the message directly in the constructor.

Q11: How do I use my own icon in the splash dialog?

This is quite tricky. Use a resource editor (if you don't know what it is, you're done :-)) to edit c:\finst22\install.exe (don't forget to backup the original). Use the CONTENTS of your own icon to replace for those of the sunset icon. That is, do NOT delete the sunset icon and then add your own, but just replace for its contents. After that, try running the install. It may or may not work. In the latter case you should contact us to patch the install.exe for you (that is, if you REALLY want it that badly, otherwise don't waste our time :-)).

Q12: How do I drag a whole directory from Explorer or File Manager?

Do it the same way as you would when dragging files. If you're using File Manager make sure the drag the directories in the right window as opposed to the left one. Once you drop the directory in the constructor it will be expanded into many individual entries, one for each file in that directory. If some of the files are already on the file list, the constructor will notify you of this. Then you should choose "skip all" so that those duplicated files will be skipped. The net result is for every file in that directory there will be exactly one entry on the file list. In the next major upgrade there will be real support for directory entries.

Q13: How do I display a background bitmap?

Note that we are talking about the optional bitmap appearing at the center of the blue

background, not those displayed in the ad dialogs. To make use of this function, use a resource editor to edit c:\prj\usr\fimain.rc. Add a bitmap called "BKBITMAP". You must use ascii ID. Anything ≥ 256 colors won't be used. You can't control the position of it. It will always be displayed at the center. In the next major upgrade there should be better support for this.

Q14: How do I re-arrange the order of the files, components, events and other various entries?

Drag'n'drop. It's that simple!

Q15: How do I copy an entry for a file, component, event and other various entries?

Drag'n'drop with the Control key pressed.

Q16: How do I install private DLLs?

All DLLs have the following default settings:

```
Target Dir:  $y
```

For private DLLs, you should set them to:

```
Target Dir:  $i
```

Q17: How do I resolve prepared file name conflicts?

Name conflict occurs when, say, you have both myfile.ext and myfile.exe. To resolve the conflict, you should set the parameters for myfile.ext like this:

```
Source Dir:  $s                               File: myfile.abc
```

This way myfile.ext will be compressed as myfile.ab_. Everything else can remain unchanged.

Q18: How do I install two different files with the same name?

Put them in different sub-directories:

```
Source Dir:  $s\mosaic                       File: readme.txt
Target Dir:  $i\mosaic                       File: readme.txt
```

```
Source Dir:  $s\winsock                      File: readme.txt
Target Dir:  $i\winsock                      File: readme.txt
```

Q19: How do I conditionally execute or skip an event?

There is a condition field associated with each event. You can enter a C/C++ integer expression there. At run time the expression will be evaluated. If the result is true (non-zero), the event will be executed; otherwise it will be skipped.

Most C/C++ integer operators have been implemented, including +, -, *, /, >, >=, <, <=, ==, !=, &&, ||, !, &, |, ^, ().

You can also use your own integer variables here (in fact, all variables are user-defined -- there is no concept of system variables in v2.2). To set/initialize their values, you can make use of the various events, such as getting Window's version, checking the selection state of a

component, checking the existence of a file, getting an INI value, getting a registration key value and etc.

Q20: How do I display readme.txt upon completion?

Add an "execute external program" event to run notepad:

```
Program command line: notepad $i\readme.txt
[x] Wait for its termination
Var for return code: <empty> <-- don't care about exit code
```

Check with tutorial 3 for more info.

Q21: How do I set up a multiple disk install?

No special steps are required. You don't even need to set the "disk description" for the files. Let the constructor do it for you.

Q22: How do I distribute zip files and unzip them during the install?

Generally it is not recommended that zip files be used since the (de)compression routine in Freeman Installer is incompatible with zip files. If you really want it that badly, you can:

1. APPEND an entry for your zip file in Interpret | Auxiliary Files or Compile | Auxiliary Files so that it will get copied into the temp directory at the very beginning of the install. You must append that entry at the end because the existing auxiliary files are far more vital to the installer and thus should be copied first. Your entry should look like:

```
Local path: d:\myapp\myfile.zip Prepare: Copy
Filename:  myfile.zip
```

Since it doesn't make much sense to compress a zip file, you can simply set "prepare" to "copy".

2. APPEND an entry for pkunzip.exe:

```
Local path: c:\pkzip\pkunzip.exe Prepare: Compress
Filename:  pkunzip.exe
```

3. Create an event to get the temporary directory so that we can execute pkunzip.exe:

```
Type: Get Temp directory
Var:  t
```

After the execution of this event, \$t will mean the temp directory. Since pkunzip.exe and myfile.zip are auxiliary files, they will be copied to \$t automatically at the beginning of the install and be deleted at the end.

4. Execute pkunzip to unzip the zip file:

```
Program command line: $t\pkunzip $t\myfile $i\
[x] Wait for its termination
Var for return code: <empty> <-- don't care about exit code
```

As said, pkunzip.exe will be present in \$t during the install, therefore the command line starts

with \$t\pkunzip. Note the use of \$i\ on the command line. It tells pkunzip to unzip the files into the install target directory. You can put whatever directory there to suit your purpose but don't forget to add that trailing "\".

Q23: How do my users start the installer from their workstation?

When you create the project, select a directory on the server as the project directory (such as w:\prj). Then your user can start the installer by running w:\prj\int\run\install.exe or w:\prj\com\run\install.exe, depending on you use the Interpret commands or the Compile commands.

Q24: How do my users build the disk set from their workstation?

When you create the project, select a directory on the server as the project directory (such as w:\prj). When you're satisfied with the install, set w:\prj\install.inf (or setup.inf) to read-only and make the constructor available to your users. Then your users can use the Build Disk Set command to build the disk set themselves.

Q25: How do I attach the user info to the installer once and for all?

When you create a new project, make sure you select the option "Save user info with installer".

If you didn't, then you will have to:

1. Choose Interpret | Auxiliary Files and create an auxiliary file entry like:

```
Local Path:  c:\finstall\defui.dat      Prepare: Copy
Filename:    ui.dat
```

2. Go to the event list and locate the "ask user info" event. Set it to:

```
[x] Save user info into installer
```

Q26: How do I remove the user info saved on the floppy disk (distribution)?

```
c:\>copy c:\finstall\defui.dat a:\ui.dat
```

Q27: How do I remove the user info saved on the hard disk (test run)?

Either issue Interpret | Purity or del c:\prj\int\run\ui.dat

Q28: How do I retrieve the user info from within my own application?

The user name and company name are stored in \$i\filog.ini encrypted along with their respective checksum. filog.ini looks like:

```
[control]
.....
User=k&3lk!A#
Comp=(*A@5fk2
ChkSumUser=4876
ChkSumComp=8751
.....
```

For example, to read the user name back, in your own app you can do something like this:

```
#include "scramble.h" /* available from us upon registration */

int r;
```

```

int chksum;
char user[40];
static char file[] = "filog.ini";
static char sect[] = "control";

GetPrivateProfileString(sect, "User", "abcd", user, 40, file);
chksum = GetPrivateProfileInt(sect, "ChkSumUser", 0, file);
r = 1;
r = r && scramble::decrypt(user);          /* try to decrypt it */
r = r && scramble::getchksum(user) == chksum; /* verify chksum */
if (!r)                                   /* fail */
{
    error("User info has been modified!");
    return FALSE;
}
return TRUE;

```

The source code to scramble.h and scramble.cpp will be sent to you on your request after you get registered.

Q29: How do I put the user info into my own INI file during install?

1. Add an "get INI value" event to retrieve the user info from filog.ini. Note that filog.ini will be in the Windows directory until the install completes, then it will be moved to \$i. Therefore, when we read the user info, filog.ini is still in the Windows directory.

```

dir:          $w
file:         filog.ini
sect:         control
entry:        User
var for value: myvar

```

Here we stored the encrypted user name in a string variable \$myvar.

2. Choose Edit | INI Entries and add an INI entry for it:

```

dir:          $i
file:         myapp.ini
sect:         mysect
entry:        User
value:        $myvar

```

Q30: How do I re-arrange the component order?

Drag'n'drop.

Q31: How do I set up full, typical, minimum, and custom install?

Check with tutorial 2 for detailed info.

Q32: How do I disable a component?

In the component dialog, select that component and set:

```

Allow user to select:    0

```

0 means false.

Q33: How do I change the background color and the progress bar color?

Sorry, you can't do this in this version. This may be supported in the next major upgrade.

Q34: How do I install a large file spanning across disk?

Split up the file with Freeman Toolbox into small pieces (???K), then install these pieces onto the user's system. For the first part, set the create mode to "create"; For the rest, set the create mode to "append". You will need to take care of name conflicts!! Check with tutorial 4 for more info.

If you need to perform version checking, time stamp checking and etc. for the large file, you need to do it in another way. Suppose the large file is bigfile.dat. You need to:

1. Compress bigfile.dat with Freeman Toolbox to get bigfile.da_.
2. Split bigfile.da_ into small pieces bigfile.f01, bigfile.f02, ...
3. For bigfile.f01, set the parameters like:

```
Source Dir: $s           File: bigfile.f01
Target Dir: $i           File: bigfile.da_
Prepare:    Copy
Create:     Create
Transfer:   Copy
Uninstall: None
```

4. For the rest (bigfile.f02, bigfile.f03, ...), set the parameters like:

```
Source Dir: $s           File: bigfile.f02
Target Dir: $i           File: bigfile.da_
Prepare:    Copy
Create:     Append
Transfer:   Copy
Uninstall: None
```

5. Add a file entry to decompress bigfile.da_ which will be NOT be distributed by you, but created on the user's system by concatenating bigfile.f01, bigfile.f02, ... in the preceding steps. Set the parameters like:

```
Source Dir: $i           File: bigfile.dat
Target Dir: $i           File: bigfile.dat
Size:          xxxxxx    /* enter the original size here */
Cond:          Version info /* select the desired cond here */
Prepare:       None
Create:        Create
Transfer:      Decompress
Uninstall:    Uninstall
```

6. Add a file entry to del bigfile.da_ which has done its job. Set the parameters like:

```
Op type:      Delete      Dir: $i   File: bigfile.da_
```

Q35: How do I install a TrueType font?

1. Install myfont.ttf.
2. Set the "font file" flag for it in the file dialog.
3. Make an entry in Edit | Registration Keys:

```
Path: HKLM \ Software\Microsoft\Windows\CurrentVersion\Fonts
Field: My Font (TrueType)
```

(•) REG_SZ String expression: <full path to myfont.ttf>

Check with tutorial 4 for more info.

Q36: How do I display an "advertising dialog"?

Load the resource file `c:\prj\usr\fmmain.rc`, and create the usual dialog. Give it an ASCII resource ID like "mydlg". Clear the "visible" flag for this dialog (so that it only shows up when the program shows it, not when it is loaded by the system). Go back to the constructor and choose Edit | Advertising dialogs, and add an entry for "mydlg".

Check with tutorial 3 for more info.

Q37: How do I display a 256 color bitmap?

Load the resource file `c:\prj\usr\fmmain.rc`, create two bitmaps called "mybmp" (for 16 colors) and "mybmp256" (for 256 colors). You must use ASCII resource ID's for them. The naming of the 16 and 256 color bitmaps is VERY important! Put an Owner-Draw button in your advertising dialog where you want the bitmap to show up (this is also very important). Set the caption of the button to "mybmp", and the installer will do the rest. If you know MFC, you may have noticed that the concept is similar to that found in `CBitmapButton`.

Check with tutorial 3 for more info.

Q38: How do I move a project to another directory?

1. `c:\>mkdir newprj`
2. `c:\>copy c:\prj\install.inf c:\newprj`
3. Choose Interpret | Auxiliary Files, change all occurrences of "c:\prj" to "c:\newprj".

Q39: How do I create a non-English project?

When you create a new project, choose the desired language in the combo box:

Language: German

Q40: How do I port the English project to other languages?

1. Move the project to a new dir. See Q38 for more info.
2. Choose Edit | General Information, set "language" to the desired language.
3. When being asked whether to delete the user files, answer "yes".

Q41: How do I create a directory during the installation?

Install a dummy file into that directory. "Create directory" event will be available in the next major upgrade.

Q42: How do I extend Freeman's multi-lingual support?

Freeman's multi-lingual support can be extended by yourself to cover languages like Portuguese, Russian, or others (including DBCS languages), without the need to access the source code to the installer. Everything needed has already been included in the evaluation copy. However, you can't translate the uninstaller since it is a public EXE in the Windows directory that could be overwritten later by its newer version in another language when the user installs a package from another vendor. It means that even if you have a v2.2 French uninstaller installed on your user's machine, when the user actually runs the uninstaller it might have been replaced by a v2.3 Dutch uninstaller.

For Freeman Installer to use a particular language (single byte character set or otherwise), the following conditions must be satisfied:

- The full name of that language is recorded in c:\finstall\factor.ini:

```
[lang-fullname]
0=English
1=French
2=German
```

- The short name of that language is recorded in c:\finstall\factor.ini:

```
[lang-shortname]
0=eng
1=frn
2=ger
```

- There is a language specific sub-directory under c:\finstall with the same name as the short name:

```
c:\finstall\eng
c:\finstall\frn
c:\finstall\ger
```

- Language specific fimain.rc, fimain.hlp can be found in that language specific sub-directory:

```
c:\finstall\eng\fimain.rc
c:\finstall\eng\fimain.hlp
c:\finstall\frn\fimain.rc
c:\finstall\frn\fimain.hlp
c:\finstall\ger\fimain.rc
c:\finstall\ger\fimain.hlp
```

Q43: How do I make a file survive uninstall?

Set "uninstall" to "None" in the file dialog. If your package is to be installed under Win95 you should enable uninstallation even for DLLs due to the DLL counter enforced there.

Q44: How do I make an INI entry or registration key survive uninstall?

Sorry, you can't do this in this version. This will be supported in the next major upgrade.

Q44: How do I customize the on-line help?

1. c:\>copy c:\finstall\fimain.hpj \prj\usr
2. c:\>copy c:\finstall\eng\fimain.rtf \prj\usr
3. modify c:\prj\usr\fimain.rtf at wish.
4. hc fimain

This is for the English version. For other languages, replace "eng" with "frn", "itl", etc.

Q46: How do I use registration keys?

Check with tutorial 2.

Q47: How do I delete, rename, or backup a file on the user's system?

Go to the file dialog, select the appropriate "op type".

Q48: How do I check the configuration of the user's system such as Windows version, math processor, CPU?

There are events available to detect the Windows version, CPU type (286, 386, ...), presence of math processor, whether running in 386 enhanced mode. Go to the event list and add the desired event. For example:

```
type: Get Windows version
var for major version: winmaj
var for minor version: winmin
```

winmaj and winmin are two integer variables that will be set. For example, when the installer is run under Windows NT 3.1 or 3.5, the version returned will be 3.10 (yes, 3.10 even for NT 3.5):

```
winmaj = 3
winmin = 10
```

When it is run under Win95:

```
winmaj = 3
winmin = 95
```

Q49: How do I proceed to the next floppy even though the current disk is not full?

By default, the constructor packs the files onto the floppies as much as it can. If you want to force it to proceed to next disk, you can set the "last file" flag for the file that is intended to be the last file on that disk. When the constructor finishes preparing that file, it will notice the "last file" flag and proceed to the next disk.

Usually you shouldn't use this feature. One exception I can think of is when you would like to organize the files according to their purpose. For example, you may want to put all the font files onto a disk and call that disk "Font Disk" instead of "Installation Disk 3". Of course, you do this at the cost of inefficient disk space usage (you leave some disk space unused).

Q50: How do I attach version information to DATA FILES?

Use INI version. Go to Edit | General Information and set the ini version like:

```
[x] Use INI version. This version is: 1.5.0.0
```

In this example, the version is v1.5

For the files to which you want to attach INI version, set "condition" to "INI version" in the file dialog.

Q51: How do I compress and/or decompress files without the constructor?

Use Freeman Toolbox. If you like, you can distribute it with your package so that the end users can decompress the files manually when necessary. However, you can't charge anything for it.

Q52: How do I retain the time stamps of my files?

The default behavior of the installer is to keep the time stamps when it is installing the files onto the user's system. You can't change that. If you want to make the time stamps of the files on your installation disks the same as those on your local hard disk, when you issue Interpret | Build Disk Set, select:

```
[x] Copy from source files
[ ] Use time of creation
[ ] Use: Date: [          ] Time: [          ]
```

Q53: The version of my package is v2.51a, how do I set the time stamps to 2:51a?

when you issue Interpret | Build Disk Set, enter:

```
[ ] Copy from source files
[ ] Use time of creation
[x] Use: Date: [1-1-1995 ] Time: [2:51:0   ]
```

Q54: How do I install the files as fast as possible?

Choose Edit | General Information. Set it as:

```
Minimum time to copy a file (ms):    0
```

Q55: How do I check if a certain component is selected?

For example, if you want to check if "full install" was selected by the user, then use something like:

```
Type: Check if a leaf component is selected
Check if this component is selected: [Full Install][v]
var for select state:                [isfull      ]
```

If the user really selects "Full Install" then the integer variable "isfull" will be set to true.

Q56: How do I search for a file on the user's system and put its directory into a variable?

Use the "search file" event followed by a conditional "split path" event:

```
Type: Search for a file
Search starts from dir:                [x] Search into sub-dir's
File specification:                    foo.*
Var for path:                          foopath
var for error:                         isfooexist

Type: Split a full path
cond: isfooexist <---- split the path only when foo exists
Full path to split: foopath
Var for directory: foodir
var for filename: <empty> <---- we don't care about filename
```

Q57: How do I check the existence of a file on the user's system?

```
Type: Search for a file
Search starts from dir:                [x] Search into sub-dir's
File specification:                    foo.*
Var for path:                          <empty> <--- don't care
var for error:                         isfooexist
```

```
Type: Display Message
cond: !isfooexist <--- execute only when isfooexist is false
Three lines of text:
[Dude, foo is not found on your system ]
[blah ]
[blah ]
```

Q58: How do I prompt the user to choose Yes or No?

```
Type: Prompt for yes or no
Three lines of text:
[Do you feel better now? ]
[blah ]
[blah ]
Context help ID: 0
Var for return: isyes
```

If the user clicks on "Yes", isyes will be set to true (i.e., 1).

Check with tutorial 3 for more info.

Q59: How do I prompt for a string?

```
Type: Prompt for a string
Four lines of text:
[Please enter your login name. You can ]
[find it on the cover of your package. ]
[blah ]
[blah ]
Var: loginname Default value: tongk@arch
Help ID: 0
```

Then you can use the variable like:

```
Type: Display Message
cond: 1
Three lines of text:
[Your login name is $logname ]
[blah ]
[blah ]
```

Or put it into your INI file.

Q60: How do I get an INI value?

Use the "get ini value" event:

```
Type: Get the value of an INI entry
Directory: $w
File: win.ini
Section: windows
Entry: run
Var for value: runvalue
Var for error: isrunexist
```

If win.ini looks like:

```
[windows]
spooler=yes
load=
run=myapp.exe
Beep=yes
NullPort=None
```

then runvalue will be set to "myapp.exe" and isrunexist to true (i.e., 1).

Q61: How do I get a registration key value?

Use the "get reg key value" event:

```
Type: Get the value of a reg key
Key path:
HKEY_CURRENT_USER \ Control Panel \ International
Field:      iCountry
Var for value: icountryvalue
Var for error: isicountryexist
```

Q62: How do I display the value of an integer variable in a message box?

Convert it with a "convert int to string" event and put the result in a string variable whose value can be displayed with something like: \$mystrvar.

Q63: How do I detect CD ROM?

Use the "get CD root dir" event. The error code is false (0) if there is no CD drive.

Q64: How do I compare two strings?

Use the "compare two strings" event. It works just like strcmp() and stricmp(). For example:

```
Type: Compare two strings
String 1: $w                               [ ] Case sensitive
String 2: c:\windows
Var:      cmpresult
```

This event does NOT return true or false. It returns -1, 0, or 1. If the user's Windows directory is indeed c:\windows, then cmpresult will be set to 0. Note that 0 here is interpreted as "equal" rather than "false". If the Windows directory is not c:\windows then cmpresult will be set to -1 or 1, depending on if \$w is less than c:\windows or not. Therefore if you want to do something when they are indeed equal, you should write something like:

```
Type: Display Message
cond: cmpresult == 0 <---- cmpresult is 0 when they're equal
Three lines of text:
[Your Windows dir is indeed c:\windows ]
[blah ]
[blah ]
```

Q65: How do I abort the install?

Use the "abort install" event. No message will be displayed. The installer will just quit immediately.

Usually it is used in this manner:

1. Test some conditions.

2. If condition is false, display a message box.
3. If condition is false, abort the install.

Q66: How do I use '\$' in the strings?

If what you want is "abc\$def", then type in "abc\$\$def". Similar to printf("50%%"), if you know C/C++.

Q67: How do I uninstall an aborted installation?

This has been built into Freeman Installer. It can uninstall an installation aborted by the user or a severe error (like disk full). To make this work better, you should install the uninstaller onto the user's system as soon as possible, by making it the first file on the file list.

Q68: How do I insert your own C/C++ code?

This depends on how much code you want to insert. If it's just a couple of statements, you can put them in a "user code" event and everything is done. If you have a large chunk of code to add, you should:

- Put your function declarations in c:\prj\usr\fiuser.h:

```
fiuser.h:

extern void ownfunc1();
extern void ownfunc2(char targetdir[]);
```

fiuser.h will be #include'd into fiuser.cpp automatically.

- Define your own functions in, say, c:\prj\usr\ownfunc.cpp.
- Add ownfunc.cpp to c:\prj\usr\fimainc.prj or c:\prj\usr\fimainc.ide or c:\prj\fimainc.mak.
- Call your own functions in a "user code" event:

```
Type:      Execute user code
Cond:      1

Enter your own code here:

ownfunc1();
ownfunc2(i);          /* pass $i to it */
```

Q69: How do I put a version statement into my DLL?

Use your favorite resource editor to open your DLL. Add a version statement like the one shown below:

```
1 VERSIONINFO
FILEVERSION 2,2,0,0
PRODUCTVERSION 2,2,0,0
FILEFLAGSMASK 0x3fL
FILEFLAGS 0x0L
FILEOS VOS_NT_WINDOWS32
FILETYPE VFT_APP
FILESUBTYPE 0x0L
BEGIN
  BLOCK "StringFileInfo"
  BEGIN
    BLOCK "040904e4"
    BEGIN
```

```

VALUE "CompanyName", "Freeman-Teresa Software\0"
VALUE "FileDescription", "Freeman Uninstaller\0"
VALUE "FileVersion", "2.2\0"
VALUE "InternalName", "Uninstaller\0"
VALUE "LegalCopyright", "Copyright \251 Freeman Software. 1994\0"
VALUE "OriginalFilename", "llatsni3.exe\0"
VALUE "ProductName", "Freeman Installer & Uninstaller\0"
VALUE "ProductVersion", "2.20\0"
END
END

BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x409, 1252
END
END

```

Q70: How do I make a smallest installer?

1. If you have a C++ compiler and want to use the compile commands, then use them.
2. If you have a C++ compiler, want to use the compile commands and don't need Win32s compatibility, then check the "base image" option.
3. If you only use the interpret commands, don't have a resource compiler but do have a resource editor, then open c:\prj\usr\fimain.rc and save it into c:\prj\usr\fimaini.exe. Or equivalent, edit c:\prj\usr\fimaini.exe directly and delete the bitmaps.

Q71: How do I change the text messages or the size of the dialogs?

If you have a resource editor, then use it to edit c:\prj\usr\fimain.rc, save it, and then save it again but this time into c:\prj\usr\fimaini.exe. If you don't have a resource editor but do have a resource compiler, then edit c:\prj\usr\fimain.rc as a text file and then hit Interpret | Bind Resources.

End of Questions and Answers