F r a m e S c r i p t

Version 3.1

U s e r ’ s G u i d e

ELMSOFT INC.
7954 Helmart Drive
Laurel Maryland 20723
USA

Written and designed at Elmsoft, Inc., 7954 Helmart Drive, Laurel,MD 20723, USA

Table of Contents

Chapter 1

Installing FrameScript

FrameScript is delivered as a single installation file, which, when run, installs the entire product into a directory of your choosing. The form of the installation file name is **`Fsl3_XX.exe`**, where **`XX`** is the target FrameMaker version. So theFrameScript install file for FrameMaker 7.0 would be **`Fsl3_70.exe`**.

> **IMPORTANT:** You should not install one version of FrameScript over an older version. You should first uninstall the old version. You can, however, keep the old version arouind if you install the new version into a different directory. Only one can be active at a time for a particular FrameMaker installation.

Starting the installation

> **IMPORTANT:** Make sure that FrameMaker is **`NOT`** running when you install FrameScript. If it is running, then the registration step will not work.

> **IMPORTANT:** If you are installing an evaluation copy of FrameScript, ignore any references to the registration number. It will not be required.

You start the installation by double-clicking on the installation file. The InstallShield process takes over from there. It will let you choose the directory to install the product. The default directory is

```
C:\Program Files\ElmSoft\FrameScript3_XX
```

where XX is the target version of FrameMaker. Follow the instructions on the ensuing screens to create the correct directory structure and copy the files into the right places. When this is complete, a registration screen will appear. This screen will allow you to enter your FrameScript registration number. It will verify it and store it in the correct place. It will skip this step for evaluation copies. Since FrameScript is an add-on to FrameMaker, FrameScript must be registered with FrameMaker as well. This registration screen will also locate the correct version of FrameMaker and make the necessary entry in FrameMaker's add-on list. If you have more than one FrameMaker installation, you may have to choose to which version of FrameMaker you wish to register this version of FrameScript.

Summary of Steps

1. Double click on the installation file.

   The install file does not have to be in any special directory. You can keep it anywhere you keep your downloaded files.

2. Choose the directory to place the files.

   The default directory will be the correct one for most installations.

3. When the registration screen appears, enter your FrameScript registration number, user name, and optionally you company name.
4. Make sure the correct version of FrameMaker is identified. If not, choose the correct version.

   IMPORTANT: The registration program looks at the Windows Registry to find the correct location for the FrameMaker version corresponding to the FrameScript version you are installing. Most of the time, it will find the correct version. In some cases, however, especially if you have more than one copy of the same version of FrameMaker installed, the registration program may not be able to find it. If this happens, use the Browse button on the registration dialog, to select the correct version of FrameMaker. Select the FrameMaker executable file (**`FrameMaker.exe`** or **`FrameMaker+SGML.exe`**, depending on the version you have).

5. Press Register

Completion

When the above steps finish FrameScript should be ready to run. The next time you start FrameMaker you should see the FrameScript menu in the FrameMaker menubar. If this does not happen, look at the installation trouble shooting section.

Directory Structure

This directory structure should look like the following illustration.

```
MainDirectory
   SysScripts-- Sub-Directory, Contains various system scripts
   Docs      -- Sub-Directory, Contains documentation files (PDF)
   Lib       -- Sub-Directory, Contains library scripts
   Demos     -- Sub-Directory, Contains demonstration scripts and docs
   SampleScriptsSub-Directory, Contains the sample scripts
   Tutorial  -- Sub-Directory, Contains a tutorial
   fsl3_70.dll  File, The FrameScript Api client
   WinSys.dll   File, An auxillary client file
   fscript.ini  File, The customization file
   RegisterEsl.exeFile, The registration program
   EslReg.ini   File, contains the registration information
   RelNotes.pdf File, Release Notes for this release.
```

Installation troubleshooting

If the FrameScript menu does not appear when you start FrameMaker, first check to make sure that the above directory structure is in place. If not, then the installation failed for some reason. You may have to try to uninstall it before trying to install it again. See the uninstall instructions.

If the product installed correctly (the above directory structure is in place), but the FrameScript menu does not appear on the FrameMaker menu bar or if you get the message that you have an invalid registration number, then the registration step may have failed. You can try running this part again by running the RegisterEsl.exe program from the main directory. This will bring up the registration screen again. Check to make sure all the information is correct including the FrameScript registration number.

If the registration screen does not work or cannot find the correct version of FrameMaker, you can try to register the product manually. See below.

If every attempt fails and you need to contact Tech support, make sure you provide the following information:

- FrameMaker registration number
- FrameMaker version
- FrameScript registration number or evaluation
- Version of MS Windows
- A list of steps the you followed above and where in the installation it may have failed

Manual registration

IMPORTANT: Make sure that FrameMaker is **NOT** running when you manually install FrameScript. If it is running, then the registration step will not work.

To register FrameScript manually you will need to use a text editor, such as notepad.exe, located in the accessories folder (Start->Programs->Accessories->NotePad). Open the fscript.ini file located in your FrameScript directory. Under the [RegInfo] section enter you user name (and optionally company name) and your FrameScript registration number (RegNum), as follows:

```
[RegInfo]
User=My Name
Company=My Company
RegNum=03-3-5X-NNNN-AAAA-N
```

Make sure you replace the above entries with your information, especially your FrameScript registration on the RegNum line.

Save this file when you are finished. This will record your FrameScript registration information. The next step is to make FrameMaker aware of the FrameScript client.

To add FrameScript to the list of FrameMaker clients, do the following:

Use notepade again and open the file maker.ini, located in your FrameMaker directory. This file might also be fmsgml.ini if you are using the FrameMaker+SGML versions of Frame 6.0 or 5.5.6.

Locate the section called [APIClients]. Go to the end of this section and add the following line for FrameMaker versions 6.0 and below:

```
fsl=Standard,FrameScript,YourFrameScriptDirectory\Fsl3_XX.dll
```

For FrameMaker versions 7.0 or greater, add the following line:

```
fsl=Standard,FrameScript,YourFrameScriptDirectory\Fsl3_XX.dll,all
```

the last part is the name of the FrameScript client name found in the installation directory. Replace the YourFrameScriptDirectory with your actual FrameScript installation directory.

Save the file and re-start FrameMaker.

Uninstalling FrameScript

You uninstall FrameScript by using the MS Windows Add/Remove programs icon located in the MS Windows Control Panel. Bring up the control panel and double click on this icon. It will provide a list of installed software. Select the line identifying FrameScript and click uninstall.

Chapter 2

Using FrameScript

When FrameScript is installed, it will place a new menu on the FrameMaker menubar called FrameScript (though this may be changed using the customization options). This menu will contain siz commands (and optionally a sub menu). The Run, Install, Uninstall and Compile commands will be discussed in this chapter. The Options and Script Window commands will be discussed in subsequent chapters. These involve customization and script development respectively.

Using FrameScript primarily involves running and installing scripts. These scripts may have been written by you or provided by others. A script is usually a text file (commonly using the extension fsl) containing a set of FrameScript commands. In some case, tt can also be an object file (FrameScript style object file), commonly using the extension fso. These object files are generated using the Compile command. This is usually done by the script writer and is not of interest to someone just using scripts.

There are two kinds of scripts, standard scripts and event scripts. These are discussed more thoroughly in the Basics.pdf document. Standard scripts are designed to run and then return to the user. Event scripts are installed,which means they are loaded in the system and they wait for any of the system events for which they were programmed to wait. These events can be menu command events, messages or system notifications. The script developer (scripter) decides what type of script it will be when it is desgined. As a script user, you will need to know whether a script is a standard script or an event script before you can use it, because the menu commands will work differently for each type.

Running a script is simple. You just need to select the **Run** menu command, from the FrameScript menu, then choose the script file (containing a standard script) from the resulting dialog. This dialog allows you to navigate to any place on your hard disk (or network disk).

Installing scripts can be more complicated because there are two kinds of scripts that may be installed. Installing standard scripts is just a way of making it easier to access them by automatically creating a menu command which allows the user to run that script. When event scripts are installed, however, they are loaded into the system where they wait for their events to occur.

Compiling scripts is something that script writers do. This takes a script in a text file and converts it into an object file.

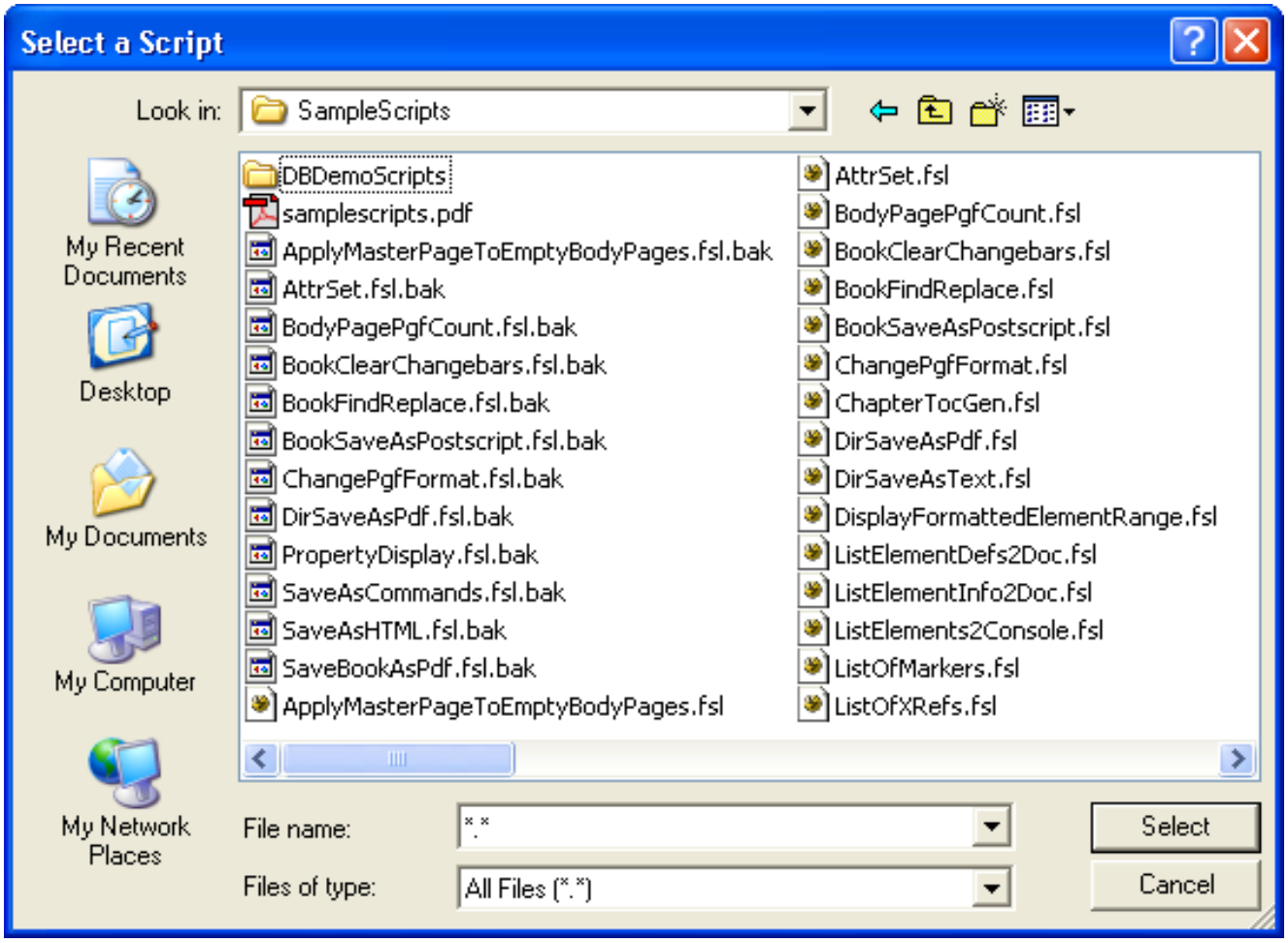
IMPORTANT: If you wish information on writing scripts, see the Basics.pdf document.

Running Scripts

Run command

You use the Run command to start a standard script. When you select this command a dialog box appears asking you to select the script you wish, See “Select a Script to Run” on page 6. If you select an event script instead of a standard script, nothing will happen.

Figure 2-1 *Select a Script to Run*



When you choose your script it runs immediately.

IMPORTANT: If you wish to stop a running script after it has started, press the ESC key. This causes the script to come to a halt immediately after the next command finishes.

Installing Scripts

Install Menu command

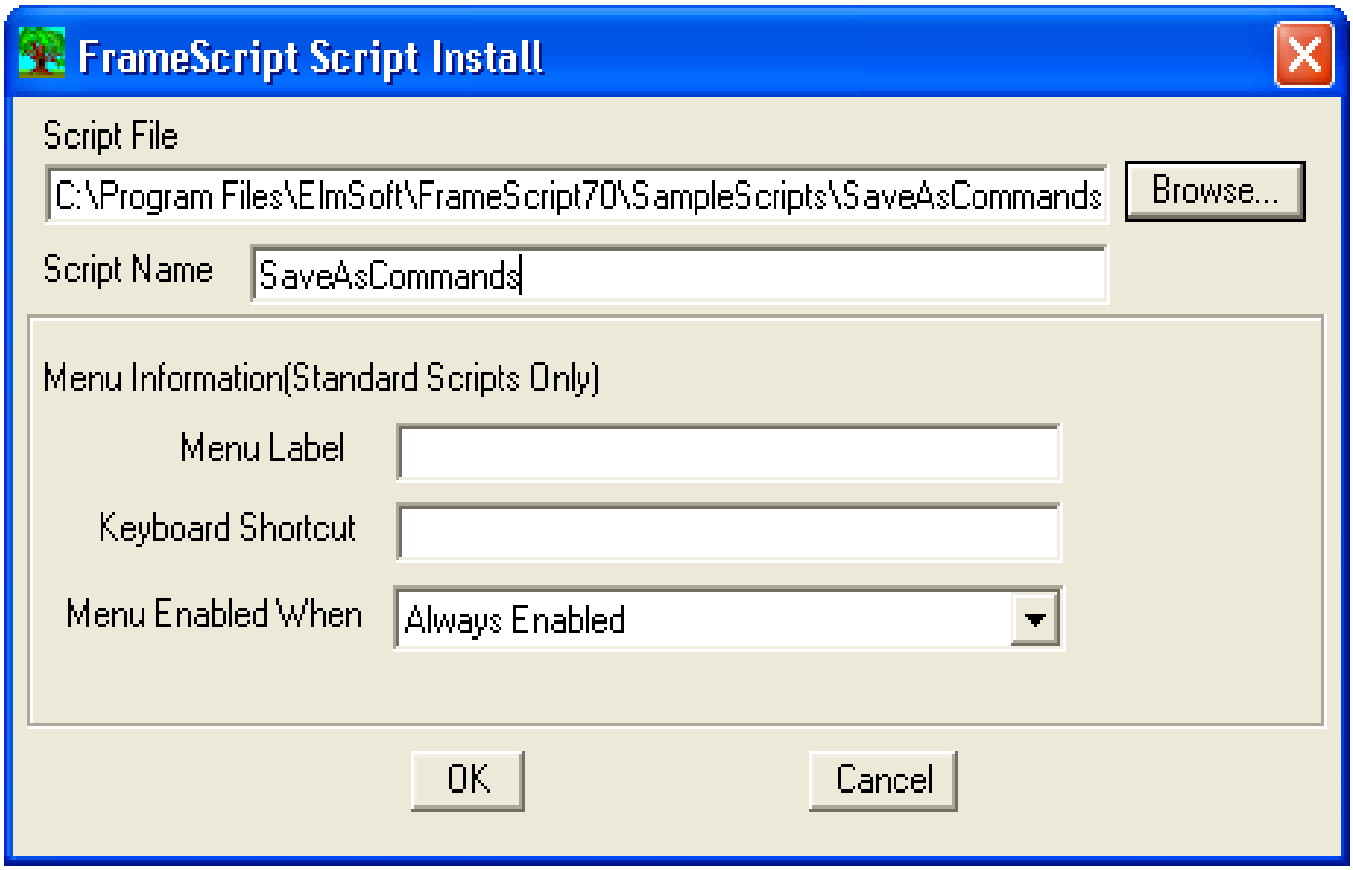
The Install menu command tells FrameScript to install a script into the FrameScript system. A standard script will be assigned a menu item for easy access. An event script will be initialized so it can handle any events it has defined. When you select this menu command, FrameScript will present the FrameScript script install dialog box (See Figure 2-3, “Install Standard Script,” on page 8). You can enter the file name of the script you wish to install (or more conveniently use the browse button to select a script file). Then enter a name for the script. This name is used to allow

you identify the script later if you wish to uninstall it. If you do not specify a name, FrameScript will make up a name (StandardScriptName n), which is not very helpful.

If you are installing an event script all you have to do is press the OK button and the script will be installed. The figure below shows the install event script dialog.

> **IMPORTANT:** A script is installed only for the current FrameMaker session. To have a script installed each time FrameMaker starts, use the Install script command in the Initial Script. See "InitScript=c:\FrameScript\myinit.fsl" on page 17

Figure 2-2 *Install Event Script*

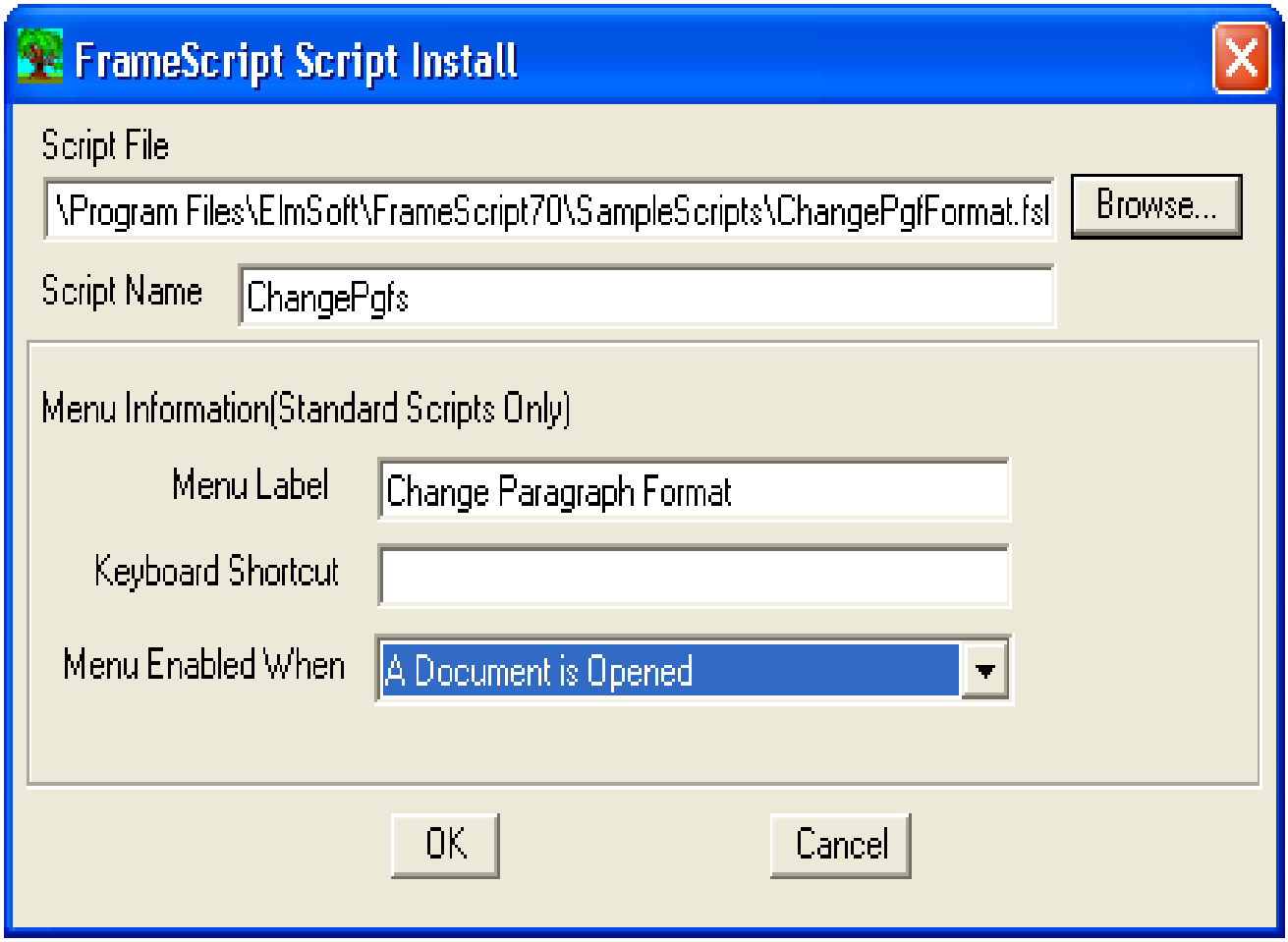


If you are installing a standard script, you should supply more information. Since installing a standard script creates a menu command for easy access, you should enter the text of that menu command. If this is left blank a default name will be chosen. You can optionally supply a keyboard shortcut and you can also specify when the menu command will be activated. The following table shows the text to use for each type of keyboard shortcut.

Keyboard key combination	Text for the keyboard shortcut field
Alt-Fn (where Fn is any function key)	**`~/Fn`**
Ctl-n (where n is any letter)	**`^n`**
Ctl-Fn (where Fn is any function key)	**`^/Fn`**
Sft-Fn (where Fn is any function key)	**`/Fn`**

The figure below shows the install standard script dialog.

Figure 2-3 *Install Standard Script*

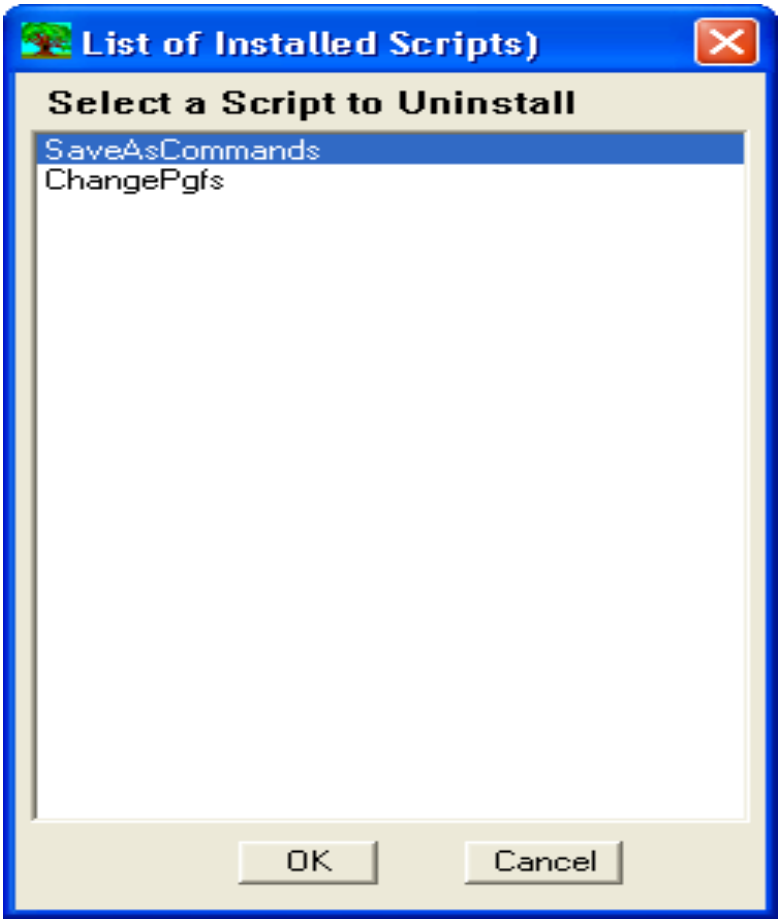


Uninstalling Scripts

Uninstall Script Menu Command

The Uninstall script command removes a previously installed script from the script space, making it unavailable to the user. When the user selects this command, a dialog box will appear with a list of all the installed scripts, see Figure 2-4. Select the one you wish, then press the OK button. The selected script will be removed.

Figure 2-4 *Uninstall Script (Windows)*



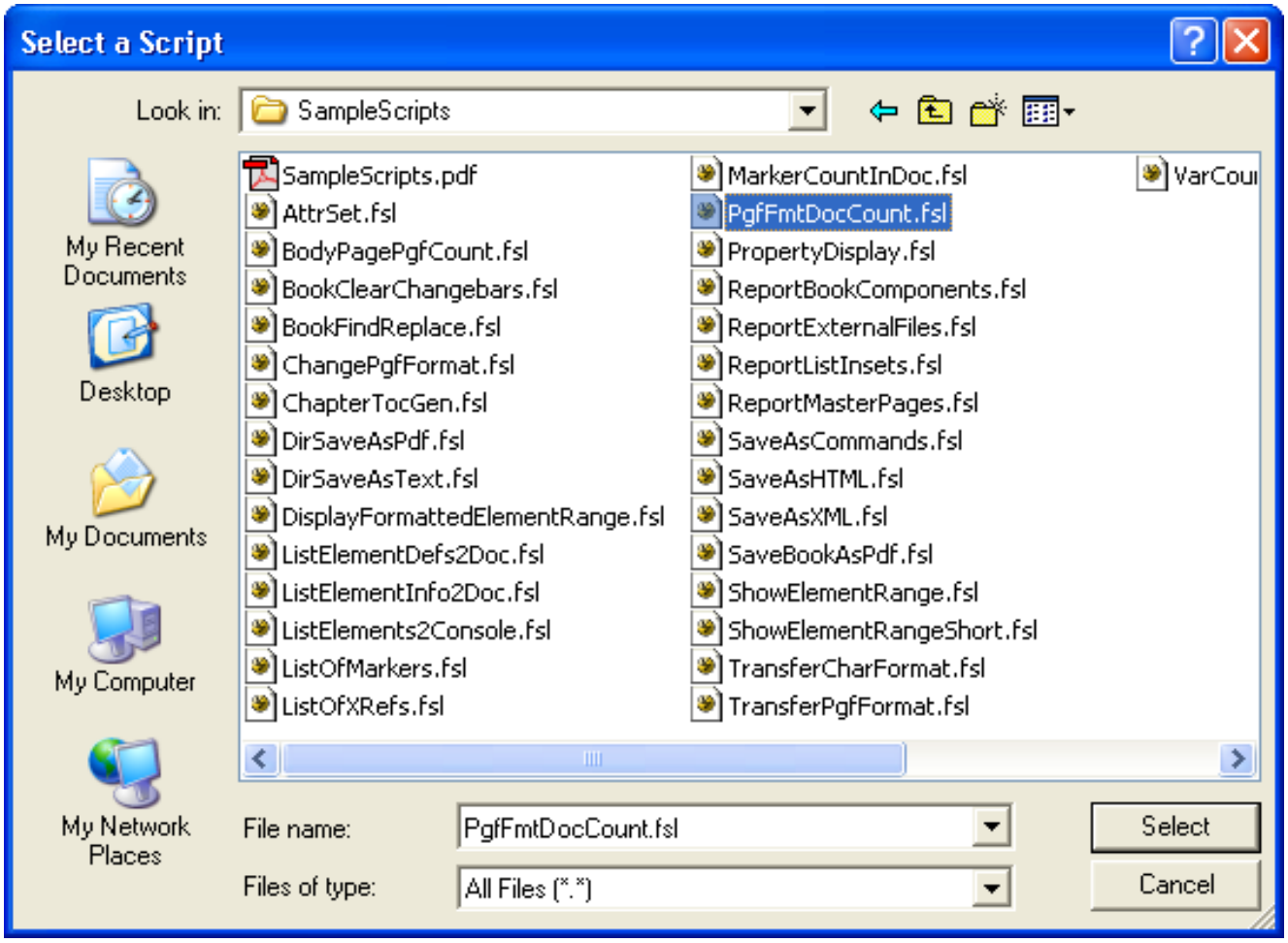
Compiling Scripts

The compile menu command allows the script writer to convert a script from the text file format to the FrameScript object format. The primary reason for doing this is to distribute a script without giving the user the ability to modify it (or even look at it). This is useful in organizations where a small number of script writers write scripts for others to use and they do not want the user modifying them. It is also useful for script developers who sell scripts to others.

Compile Menu command

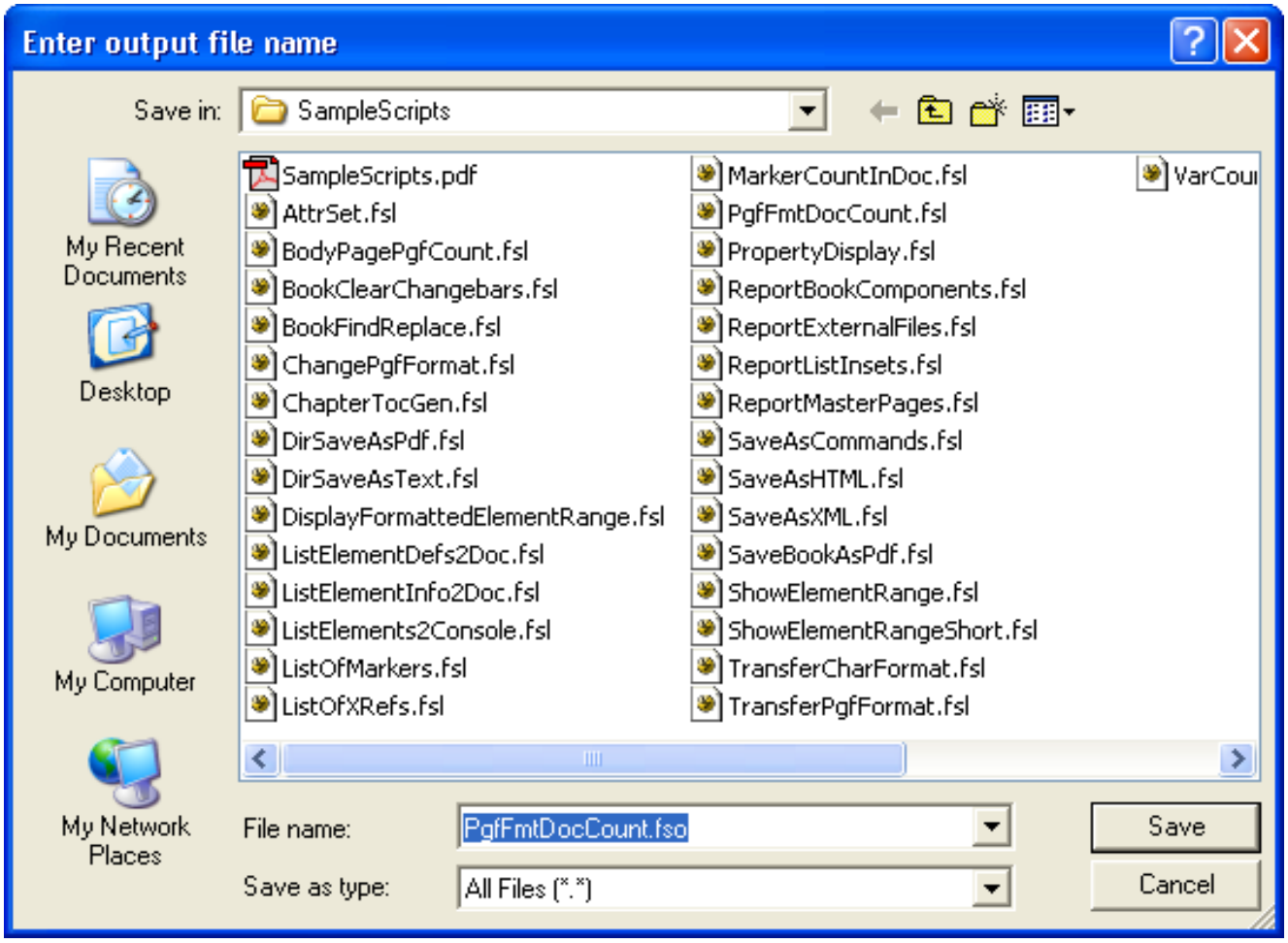
When you select the FrameScript->Compile command a dialog box appears asking you to select the script to compile. This screen is very similar to the one used to select a script to run (See “Select a Script to Compile” on page 10).

Figure 2-5 *Select a Script to Compile*



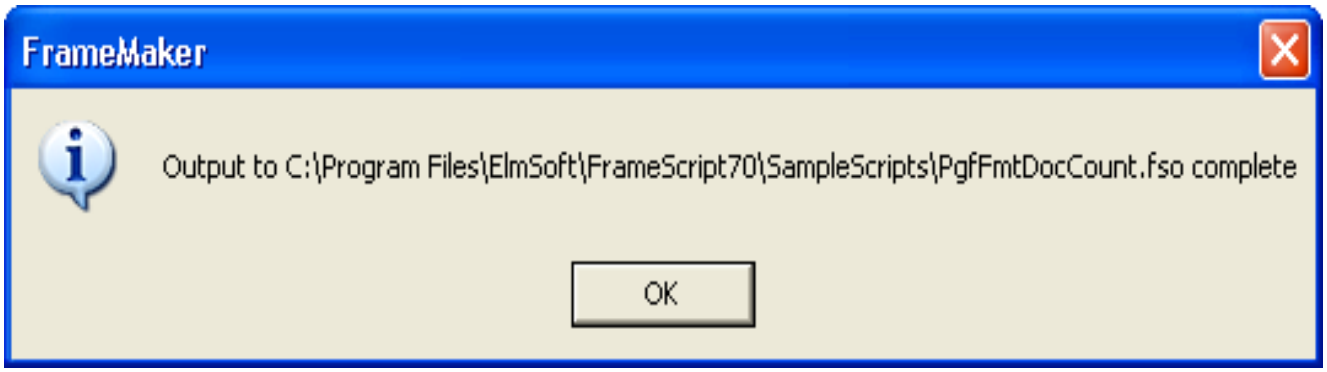
When you select a script, however, it does not run it but instead it puts up a second dialog asking for the location to put the compiled script (See “Select the output file for the compiled script” on page 11). It starts in the same directory as the source script and places a default name using the name of the source script with a fso extension. Most of the time, you will want to keep it that way. If you don’t, then you can change the name (and/or directory) to suit your needs.

Figure 2-6 *Select the output file for the compiled script*



A confirmation message appears when the process is complete.

Figure 2-7 *Confirmation of the compiled script function*



Chapter 3

Customizing FrameScript

You can customize the way that the user interface looks, how FrameScript responds to errors, how it searches for scripts and other items. You do this with the Options menu item

Options Menu Command

You can modify the customization options by using the Options menu command. This command brings up a dialog box which has three panels selected via a drop down box. The General panel allows you to select the initial script, log file and error handling. The Search path panel defines how FrameScript searches for scripts and the Menus panel lets you change the names of the menu items and it also lets select which menu items (if any) will appear.

Use the drop down box to change to a different panel.

General Panel

The initial script field allows you to select a script to run when FrameMaker (and FrameScript) starts. The Keep initial data checkbox says to keep the global data space available when the initial script terminates.This allows you to define read-only variables for all other scripts to use. You would want to turn this off if these variables might interfere with variables defined in other scripts.

If the Signon screen checkbox is on, then a small window will appear whenever FrameScript starts. Uncheck this to turn off that behavior.

The following figure shows the general panel.

Figure 3-1 *Options Dialog (General Panel)*

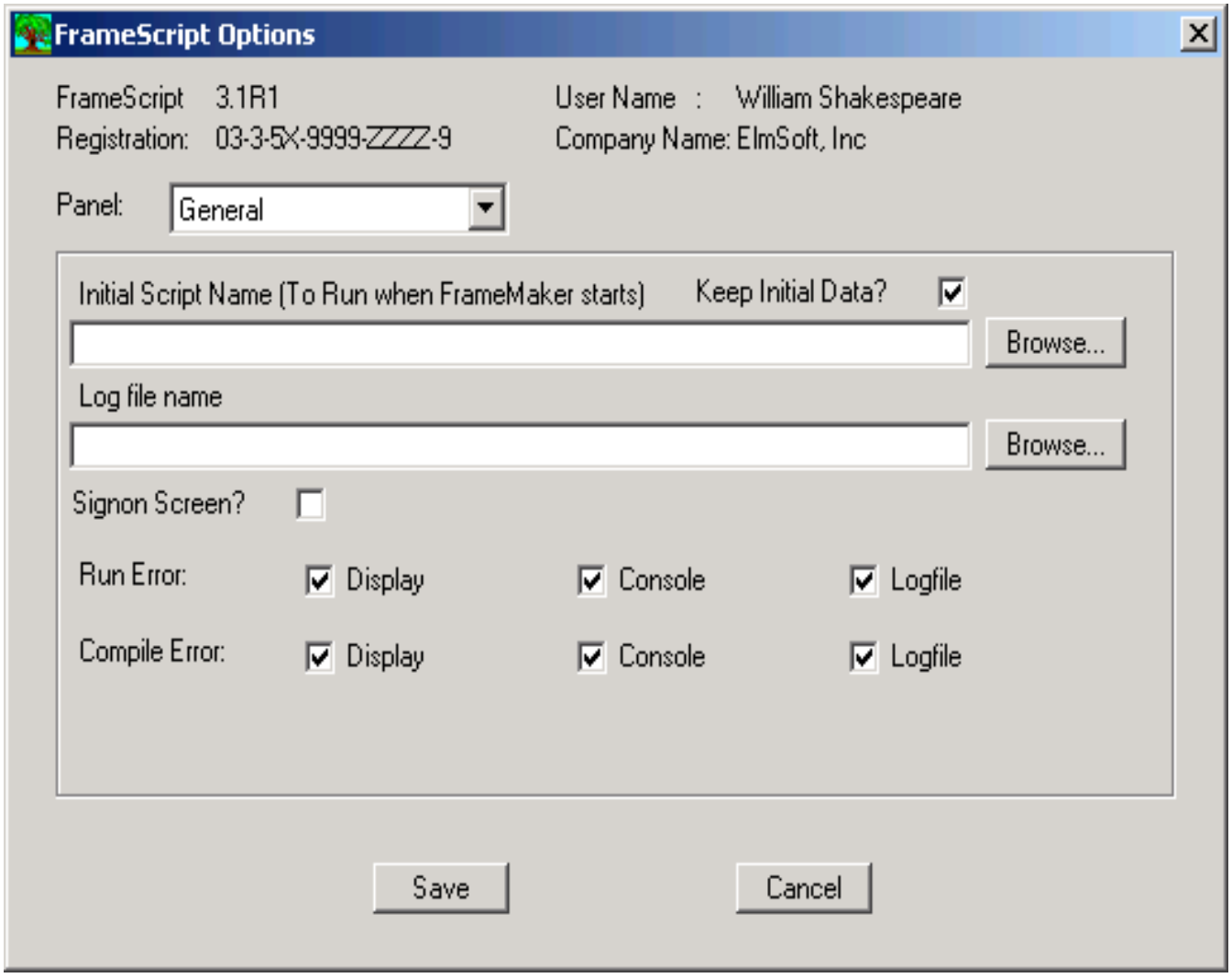
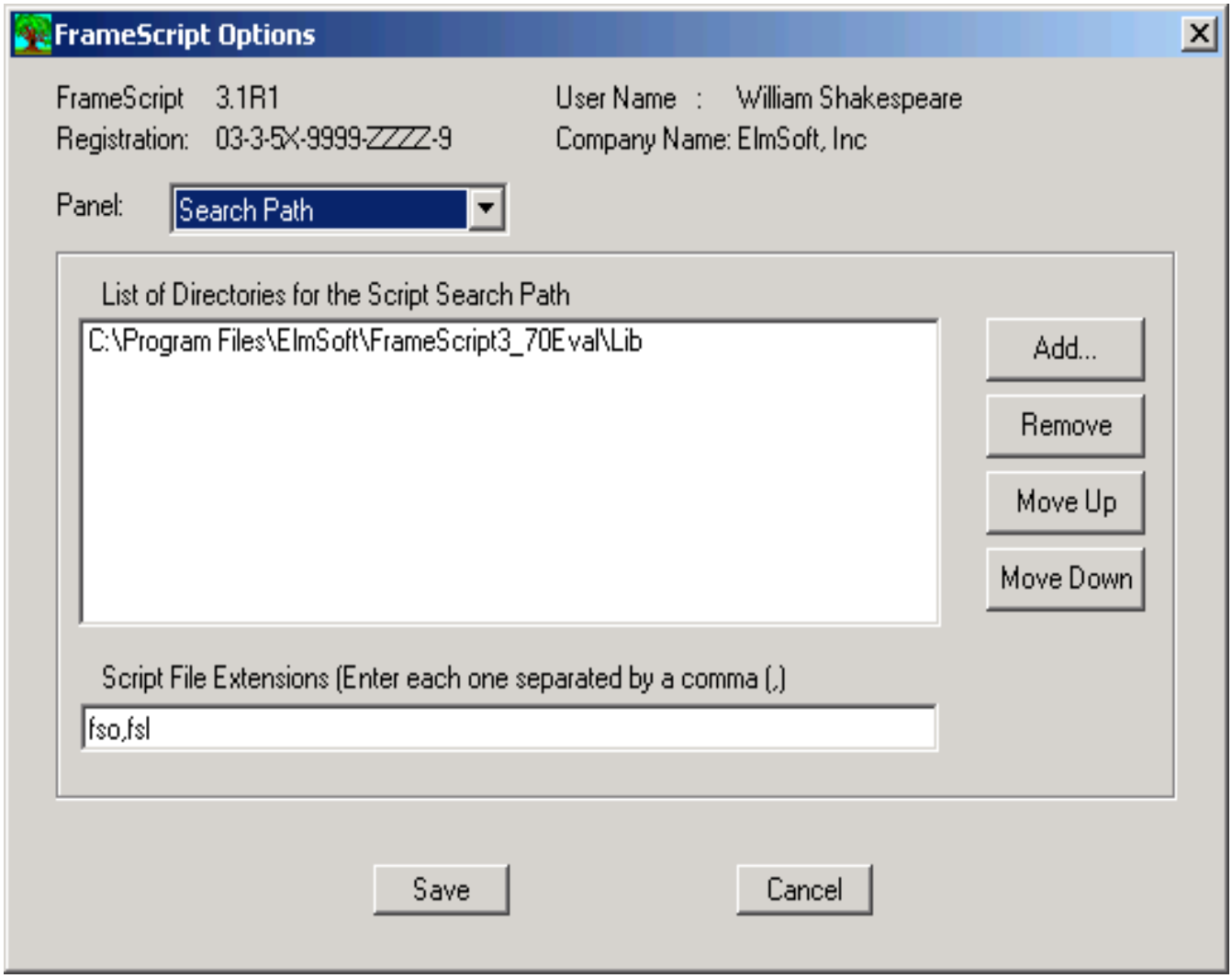


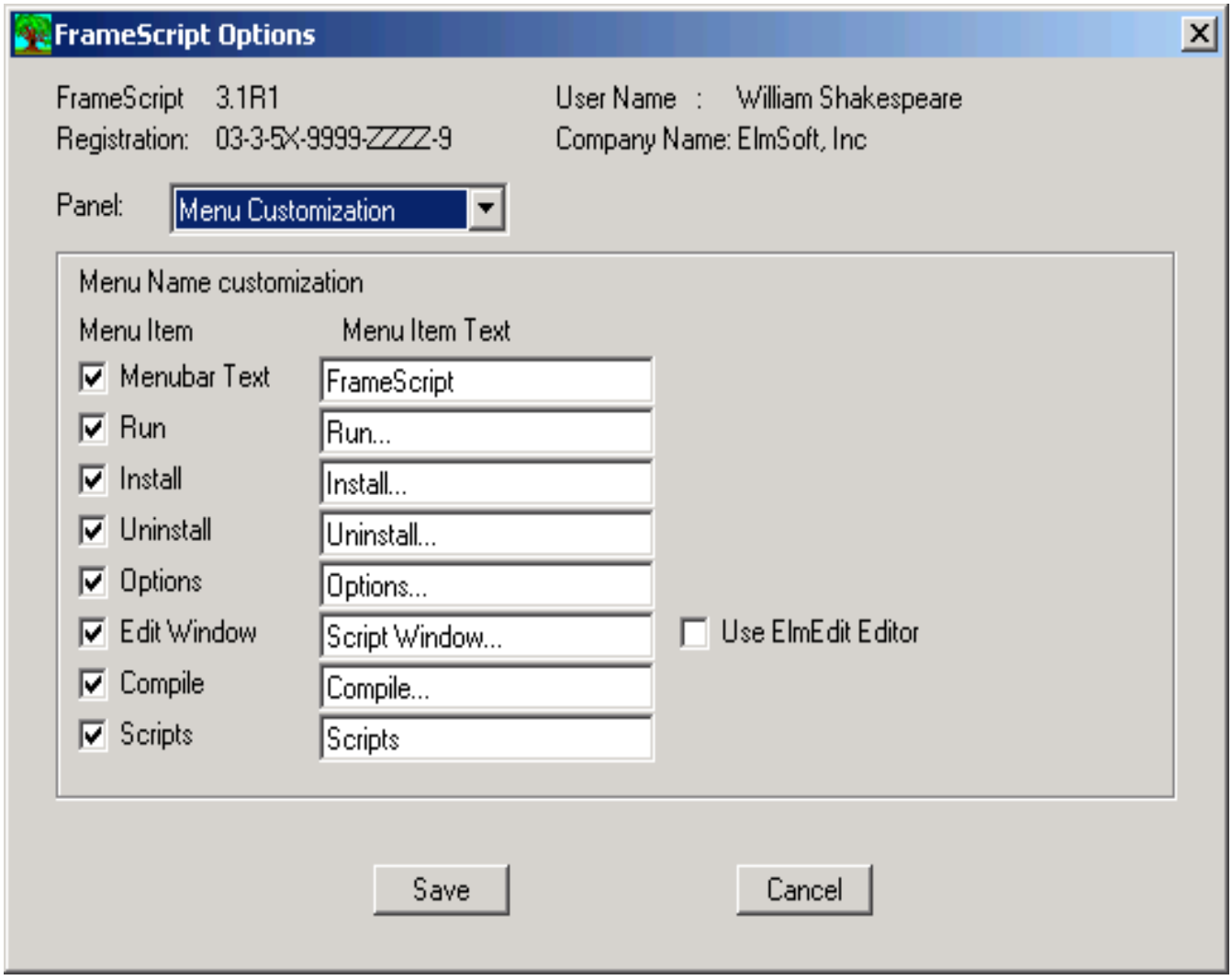
Figure 3-2 *Options Dialog (Search Path Panel)*



Search Path Panel

If a FrameScript command tries to access another script (the install script command for example), and a full pathname is not specified, FrameScript will search the directories in the search path for that file. If the file extension is not given, then it will use the set of file extensions specified here. Use the buttons to add new entries or change the order of the directories.

Figure 3-3 *Options Dialog (Menus Panel)*



Menus Panel

This panel lets you customize the user interface. A check in the checkbox indicates that the corresponding item will be available to the user. You may also change the text of the menu item. If you turn off the Menubar text, then the FrameScript menu will not appear at all.

Customization using the Ini File

FrameScript provides a customization file which allows you to set various options. Ordinarily, you will use the Options dialog box to modify these items, but you can use a text editor to modify them manually. Make sure the FrameMaker is not running when you do this. The name of the customization file is `fscript.ini`. It must be located in the same directory as the FrameScript client program (`fsl3_xx.dll, where xx is the target FrameMaker version`).

The following table shows the section names, options, a description and the default values that will be used if it is not specified in this file.

TABLE 1. FScript.ini File customization options

Section	Option	Description
[GENERAL]	SignonScreen	This option specifies whether you wish to have the FrameScript signon window appear when you start a Frame session. e.g. SignonScreen=Yes If not specified, the signon screen *will* appear. A value of No means it will not appear.
	GlobalDataSpaceForInitialScript	This option specifies whether you wish to have the global data space from the initial script (if any) saved as redad-only global variables. e.g. GlobalDataSpaceForInitialScript=Yes If not specified, the global variables *will* be saved. A value of No means they will not be saved.
[FILES]	LogFile	This option allows you to specify the name and location of the FrameScript log file. This log file is where FrameScript writes comments and error messages. e.g. LogFile=c:\fscript.log If not specified then no logfile will be generated.
	InitScript	This option allows you to specify the name of a script to run at the start of a FrameScript session. You must specify the complete pathname of the initial script to run. e.g. InitScript=c:\FrameScript\myinit.fsl If not specified, then no script is run when FrameScript starts.
[Defaults]	ScriptExtensions	This option specifies the default file extensions that FrameScript will search for when not specified. e.g. ScriptExtensions=fso,fsl
[ScriptSearchPath]	n (A sequence number)	This option specifies a set of directories that FrameScript uses to search when a complete path is not specified. e.g. 1=c:\Program Files\ElmSoft\FrameScript3_70\Lib 2=c:\Program Files\ElmSoft\FrameScript3_70\SampleScripts

TABLE 1. FScript.ini File customization options

Section	Option	Description
[Directories]	SearchScript	This option specifies the initial directory to look for for scripts when the user selects the Run... menu command. e.g. SearchScript=c:\FrameScript\SampleScripts If not specified, then FrameScript will start looking in the same directory as the FrameScript client program.
[MENUS]	RunMenuItem	This option specifies whether you wish to have the Run... menu item on the FrameScript menu or not. e.g. RunMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	InstallMenuItem	This option specifies whether you wish to have the Install... menu item on the FrameScript menu or not. For Example: InstallMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	UninstallMenuItem	This option specifies whether you wish to have the Uninstall... menu item on the FrameScript menu or not. For Example: UninstallMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	OptionsMenuItem	This option specifies whether you wish to have the Options... menu item on the FrameScript menu or not. For Example: OptionsMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	ScriptWindowMenuItem	This option specifies whether you wish to have the Script Window... menu item on the FrameScript menu or not. For Example: ScriptWindowMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	CompileMenuItem	This option specifies whether you wish to have the Compile... menu item on the FrameScript menu or not. For Example: CompileMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	ScriptsMenuItem	This option specifies whether you wish to have the Scripts sub menu on the FrameScript menu or not. For Example:. ScriptsMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	MainMenuItem	This option specifies whether you wish to have the FrameScript menu appearing or not. If this is not on, then the other menus items will not appear For Example: MainMenuItem=Yes If not specified, the menu item *will* appear. A value of No means it will not appear.
	UseElmEdit	This option specifies whether you wish to use the ElmEdit editor or the standard script editor. For Example: UseElmEdit=Yes No is the default value, which means that the standard editor will be used..

TABLE 1. FScript.ini File customization options

Section	Option	Description
[MENUNAMES]	RunMenuName	This option specifies the label of the Run... menu item. Default: RunMenuName=Run...
	InstallMenuName	This option specifies the label of the Install menu item. Default: InstallMenuName=Install...
	UninstallMenuName	This option the label of the Uninstall... menu item. Default: UninstallMenuName=Uninstall...
	OptionsMenuName	This option specifies the label of the Options... menu item. Default: OptionsMenuName=Options...
	ScriptWindowMenuName	This option specifies the label of the Script Window... menu item. Default: ScriptWindowMenuName=Script Window...
	CompileMenuName	This option specifies the label of the Compile... menu item. Default: CompileMenuName=Compile...
	ScriptsMenuName	This option specifies the label of the Scripts sub menu. Default: ScriptsMenuName=Scripts
	MainMenuName	This option specifies the name of the FrameScript menu. Default: MainMenuName=FrameScript
[ErrorHandling]	RunErrorDisplay	This option tells if FrameScript will report a FrameScript command run error to the display (dialog box). Default: RunErrorDisplay=Yes If not specified, no reporting is done.
	RunErrorConsole	This option tells if FrameScript will report a FrameScript command run error to the console. e.g. RunErrorConsole=Yes If not specified, no reporting is done.
	RunErrorLogFile	This option tells if FrameScript will report a FrameScript command run error to the logfile. e.g. RunErrorLogFile=Yes If not specified, no reporting is done.
	CompileErrorDisplay	This option tells if FrameScript will report a FrameScript command option error to the display (dialog box). e.g. CompileErrorDisplay=Yes If not specified, no reporting is done.

TABLE 1. FScript.ini File customization options

Section	Option	Description
	`CompileErrorConsole`	This option tells if FrameScript will report a FrameScript command option error to the console. e.g. `CompileErrorConsole=Yes` If not specified, no reporting is done.
	`CompileErrorLogFile`	This option tells if FrameScript will report a FrameScript command option error to the logfile. e.g. `CompileErrorLogFile=Yes` If not specified, no reporting is done.

Chapter 4

Using the Script Window

Script Window

The Script window is a text editing window that allows you to create, open, edit, save and run scripts without leaving the FrameMaker environment. You may use the standard editing commands (including cut, copy, paste, find) to get the script the way you want it. You can save the script and then test it by pressing the run button. The run button works only on standard scripts, See “Script Window” on page 22.

You may have several script files open at the same time. Use the drop down box at the top of the window to change the active script. The Run button will run the script in the active window.

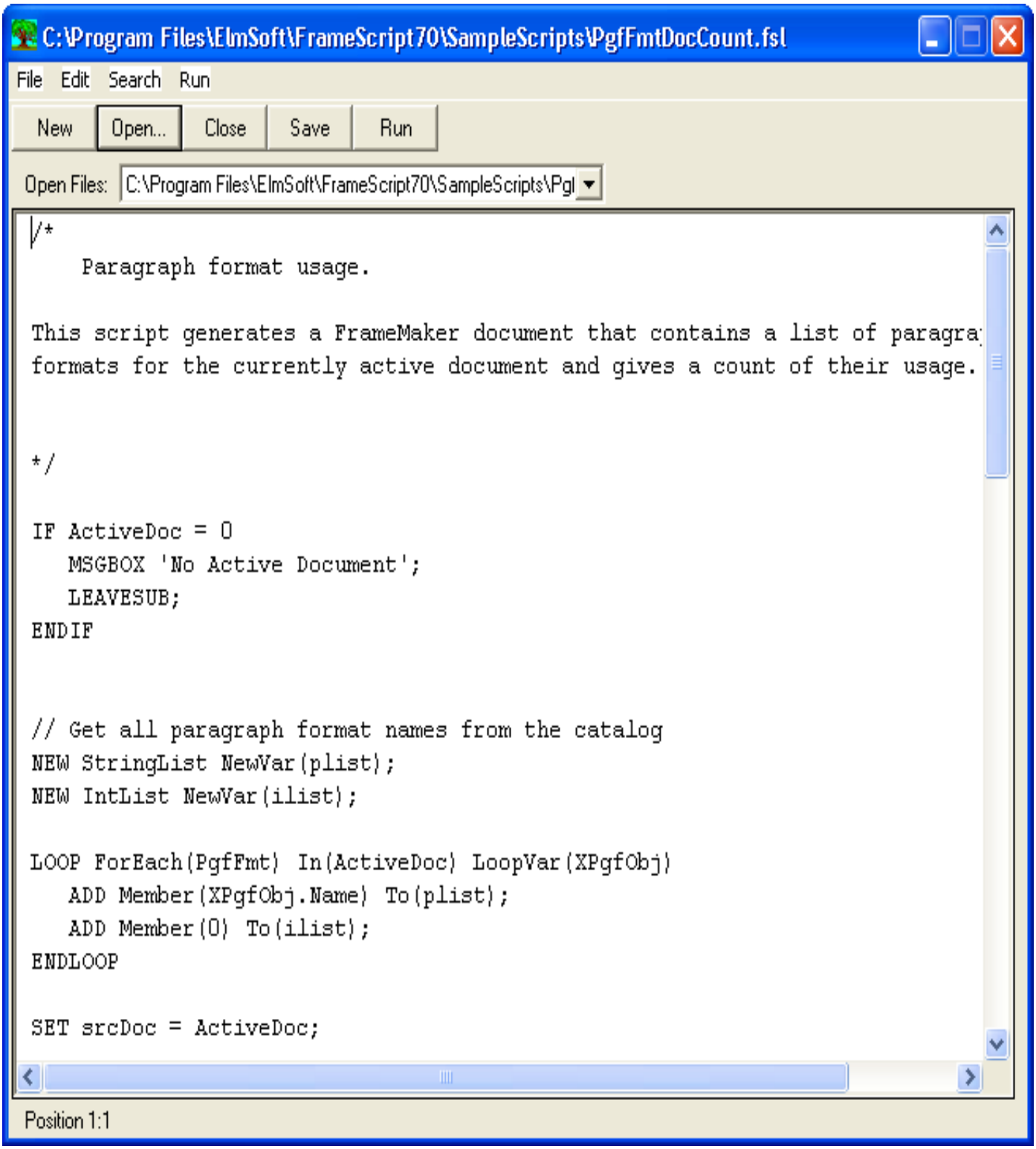
Script Window Menu Command

The Script window command brings up a script editing window. This is similar to notepad.exe except that you can edit multiple files at the same time. See “Script Window” on page 22 for a figure of the Script Window.

Using other editors

The script editor is provided as a convenience for script writers. But you do not have to use this editor. Since script files are standard text files, any standard text editor will work for creating and modifying scripts. There are many of these available, including the notepad.exe program that comes with MS Windows. Many customers have used UltraEdit from IDM Computer Solutions (www.ultraedit.com) and TextPad from Helio Software Solutions (www.textpad.com). These two text editors have programmable tools that allow direct connection to FrameScript via the RunEslBatch.exe program, described in the next chapter. Other editors may also have this feature.

Figure 4-1 *Script Window*



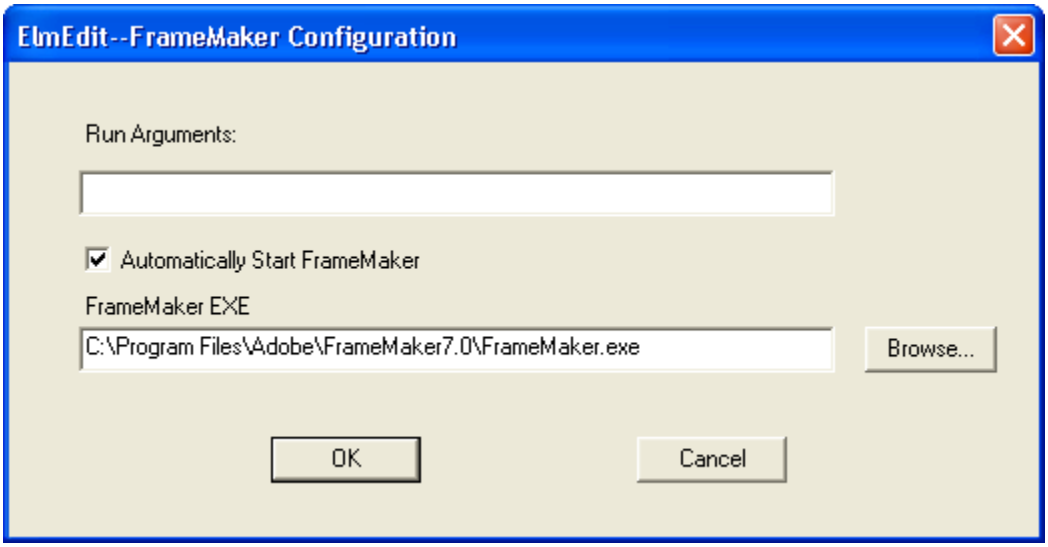
ElmEdit

ElmEdit.exe is a stand-alone, multi-file text editor that you may use to create and modify script files. This editor does not have all the bells and whistles of some of commercial text editors but it has an adequate set of features for standard text editing. Unlike the script window above, this editor is used outside of FrameMaker. You can save your script files and run them using the Run menu command. You can also run scripts directly from ElmEdit. The ElmEdit Run Menu command (or Run button) will send the contents of the current script to the FrameMaker program, where it will be run. If FrameMaker is not currently running, ElmEdit will attempt to start FrameMaker, unless configured not to do so.

Configuring ElmEdit for FrameMaker

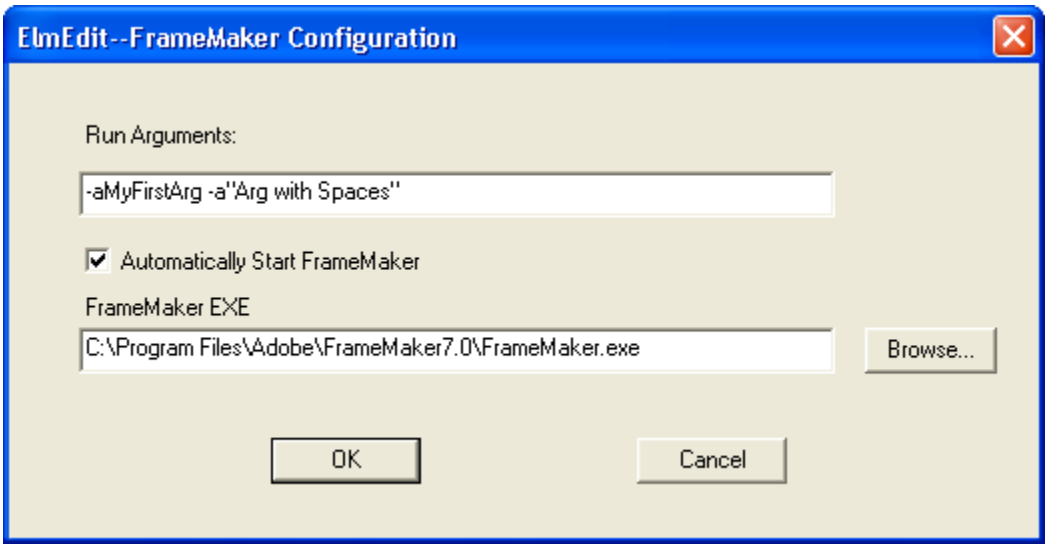
ElmEdit comes preconfigured to connect to FrameMaker without any configuration necessary. If FrameMaker is already running, it will receive the script. If FrameMaker is not running, ElmEdit will use the registry information to attempt to start FrameMaker. If there is some problem, or if you have more than one version of FrameMaker installed, then you may need to configure ElmEdit to launch the correct version of FrameMaker.

The Config->FrameMaker menu command will bring up the configuration window, as follows:



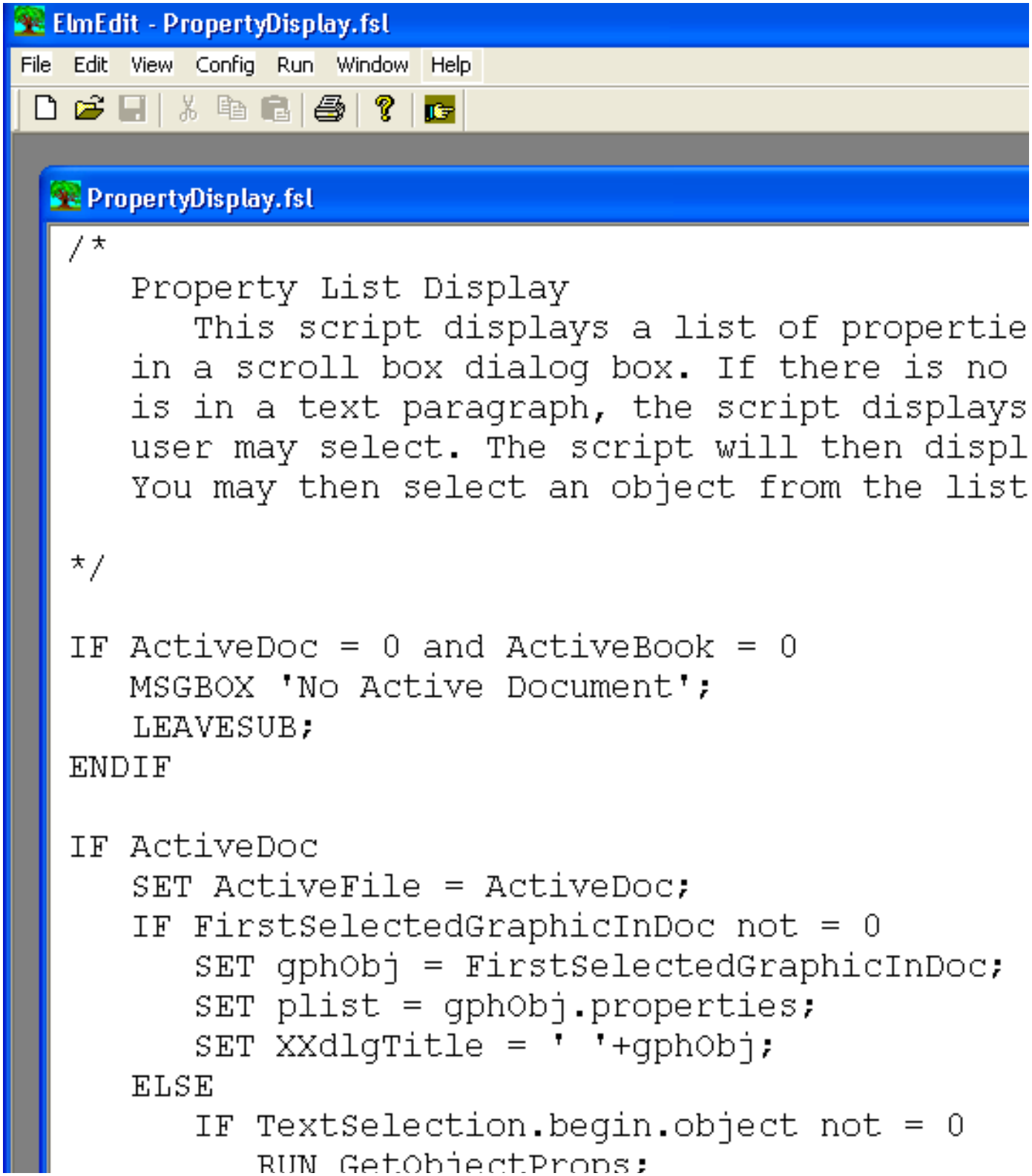
Use the Browse button to select the correct copy of FrameMaker. You can also elect not to start FrameMaker automatically, by unchecking the check box. If this is done, then ElmEdit will only run the script, if FrameMaker is already running.

The Run Arguments edit box is used to pass arguments to the script that you run. This is useful if you are testing a script that will run in batch mode (See Chapter 5, “Batch Processing.”). below is an example of setting script srguments.



Two arguments will be passed to the script will you run it from ElmEdit. These arguments are accessed by the **`Args`** array from the main script.

Figure 4-2 *ElmEdit*



Chapter 5

Batch Processing

RunEslBatch

RunEslBatch.exe is a batch oriented windows program that sends messages to FrameScript telling it to run scripts. RunEslBatch is also capable (if properly configured) to start FrameMaker if it is not already running, before attempting to send the message.

Using RunEslBatch

The format for running RunEslBatch is as follows:

```
RunEslBatch -f"ScriptFilename" [-aValue] ... [-aValue]
or
RunEslBatch -s"ScriptText" [-aValue] ... [-aValue]
```

This is how it might appear in a DOS batch file.

Using the -f option, the ScriptFilename is the full path name of the file containing the script that you wish to run or a partial name of a script that is in the SearchPath. Using the -s option, the ScriptText is the actual text of a script to run. The -a option(s) are used to pass string values to a script. Use double quotes if the values contains spaces or special characters. These arguments are passed to the script like subroutine/function arguments. You can access them in a script using the `Args` array. `Args.Count` will give you the number of arguments passed. `Args[1]` will be the first argument, `Args[2]` the second, and so on. Also the name of each argument is of the form ArgN, where N is the order number (starting at 1) of the arguments.

For example, you could have RunEslBatch send the included script as follows:

```
RunEslBatch -s" Quit Session;"
```

This single line script will cause FrameMaker to quit running.

> **IMPORTANT:** Due to limitations with the DOS/Windows command line processing, the -s option is useful only for a short scripts without special characters.

Here is a sample batch file using RunEslBatch. You could also include other programs in the batch file.

```
RunEslBatch -f"MyScript1.fsl" -a"My String Value" -a"My other string value"
RunEslBatch -f"c:\MyScripts\MyOtherScript.fsl"
RunEslBatch -s" Quit Session;"
```

IMPORTANT: When running in batch mode, be sure that the scripts do not require any user interaction, because the script will wait for it. If you wish to run a long operation over night, this could be a problem.

Configuring RunEslBatch

RunEslBatch will run without being configured if FrameMaker is already running when the program is started. If FrameMaker is not running, RunEslBatch will check the registry for the current version of FrameMaker. It will automatically start this version of FrameMaker. If you have more than one version of FrameMaker installed and RunEslBatch starts the wrong version, you can configure it manually usig the configuration file. The configuration file is called EslBatch.ini and it is located in the Windows directory (WinNT for Windows 2000 systems). It has the following form:

```
[FM]
ExeFile=C:\Program Files\Adobe\FrameMaker7.0\FrameMaker.exe
```

The ExeFile keyword identifies the location of the FrameMaker exe file. The above example shows the usual location for a FrameMaker 7.0 installation. You will need to replace this with the complete path of your own FrameMaker installation. You will need a text editor to do this.

The RunEslBatch.exe and EslBatch.ini files are installed into the Windows directory by default. They can be located elsewhere as long as they are together.

IMPORTANT: If the RunEslBatch program is not in a directory which is part of the PATH, then you will need to include a full path to run it.

Configuring RunEslBatch for Text Editors

To create and modify scripts, you can use the built-in Script Window editor, but, since scripts are standard text files, you can also use any other standard text editor to perform the same function. Notepad.exe comes as part of MS Windows, so it it always available. It is a simple, single file at a time text editor. You can use this to write and modify scripts, then test them using the FrameScript Run command. Another text editor, WordPad.exe, also comes with MS Windows. It is a more advanced editor, but you have to remember to save the files as Text-Only files. The default is to save as RTF. RTF files will not work with FrameScript. There are also 3rd party solutions, which can be purchased separately from their respective software vendors. They are generally inexpensive and have evaluation versions available.

Our customers have recommended two 3rd party Text Editors, UltraEdit and TextPad. These editors are multifile text editors that have many compelling features, including syntax highlighting. Syntax highlighting lets you configure how the text file looks on the screen. You can have various types of words color coded to make it easier to read and modify the scripts. Many users have developed a syntax lists that are freely available. Also, both these editors allow you to run other programs from inside them. This means you can use RunEslBatch to run scripts directly from the editor windows. To do this you must first configure the editor to run this program correctly.

Note: UltraEdit is available from IDM Computer Solutions (www.ultraedit.com).

Note: TextPad is available from Helio Software Solutions (www.textpad.com)

Configuring for UltraEdit

To configure UltraEdit to run scripts with RunEslBatch, do the following:

- From within UltraEdit, on the Advanced menu, select the Tool Configuration menu item (Advanced->Tool Configuration...).
- A dialog box appears that lets you define a tool. On the Command Line field, enter the following:

```
RunEslBatch.exe -f"%f"
```

- On the Menu Item Name field, enter some text of your choosing. This text will appear as a label of a menu item in UltraEdit.
- Check the box labeled 'Check if Windows Program'.

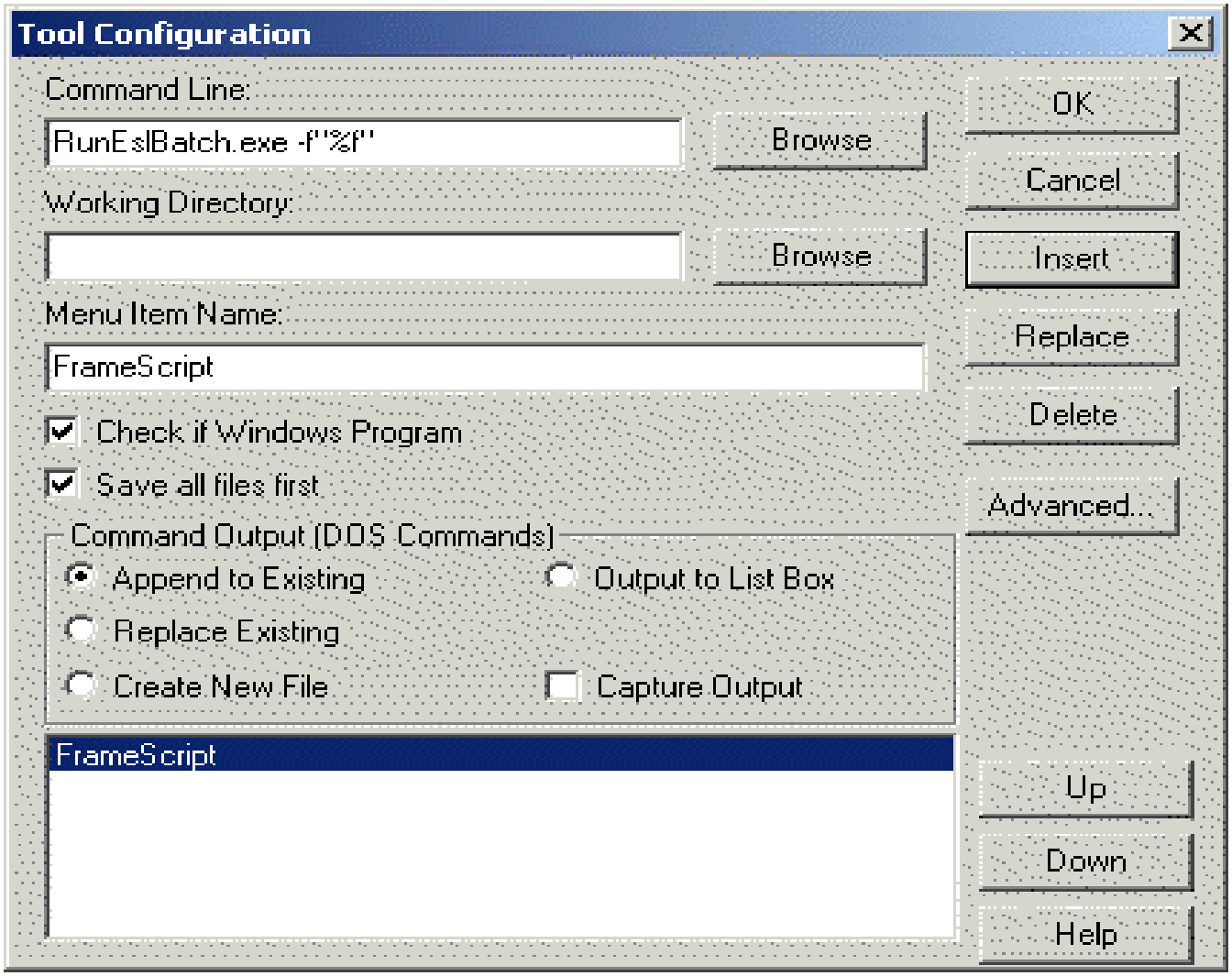
See "UltraEdit Tool Configuration dialog" on page 30 for a screen shot of this dialog.

- Press the Insert button to register the new command to UltraEdit.
- Press OK to finish.

A new menu item should now appear at the bottom of the Tools menu in UltraEdit. Whenever you select this menu item, RunEslBatch will run using the name of the currently active text file as its parameter. This means you can run the current script file whenever you wish just by selecting the menu item instead of switching to FrameMaker and using the Run command.

IMPORTANT: Make sure that the current file is saved before trying to run it.

Figure 5-1 *UltraEdit Tool Configuration dialog*



Configuring for TextPad

To configure TextPad to run scripts with RunEslBatch, do the following:

- From within TextPad, on the Configure menu, select the Preferences menu item (Configure->Preferences...).

  A dialog box appears.

- In the left panel, select the Tools item, then use the Add button (in the right panel) to add a new Program (Add->Program...).

  This will take you to a file selection dialog.

- Navigate to the Windows directory and select the RunEslBatch program.

  A new entry will appear in the list box.

- Rename this to something of your own choosing by selecting it, then making your changes.
- Press the Apply button to record the entry.

- In the left panel, expand the Tools item and select the new command.

  A dialog appears on the right panel. See “TextPad Preferences for Tools dialog” on page 31

  The Command line should be correct.
- In the Parameters text box, enter the following:

```
-f"$File"
```

- Press OK to finish

A new menu item should now appear at the bottom of the Tools menu in Textpad. Whenever you select this menu item, RunEslBatch will run using the name of the currently active text file as its parameter. This means you can run the current script file whenever you wish just by selecting the menu item instead of switching to FrameMaker and using the Run command.

IMPORTANT: Make sure that the current file is saved before trying to run it.

Figure 5-2 *TextPad Preferences for Tools dialog*