

4th Dimension™

Changes and Enhancements

This document describes changes and enhancements since the manuals were printed.

Topics that are covered include:

- Errata to the manuals
- Changes and Enhancements to 4th Dimension
- Creating Included Layouts
- The 4D xRef™ Utility
- The 4D External Mover™ (replaces 4D Mover™)
- The new 4D Customizer™

Errata

Programmer's Reference page 94 lines 29 and 35, Sentence should be vSentence.

Programmer's Reference page 103 line 6, (Name) should be (Last Name).

Command Reference page 255 table F-3, 12 and 13 in the table should be exchanged.

Changes and enhancements

This section describes changes and enhancements to 4th Dimension™ version 1.0.

Numeric formats

The list of default numeric formats has been changed for both the Format of field and the Format of variable dialog boxes. The new formats are now set to format numbers up to one million. Of course, the formats can be modified to your needs at any time.

Since some of the formats are longer, the format text area for the two dialog boxes has been expanded to two lines.

Figure 1 shows the new Format of field dialog box.

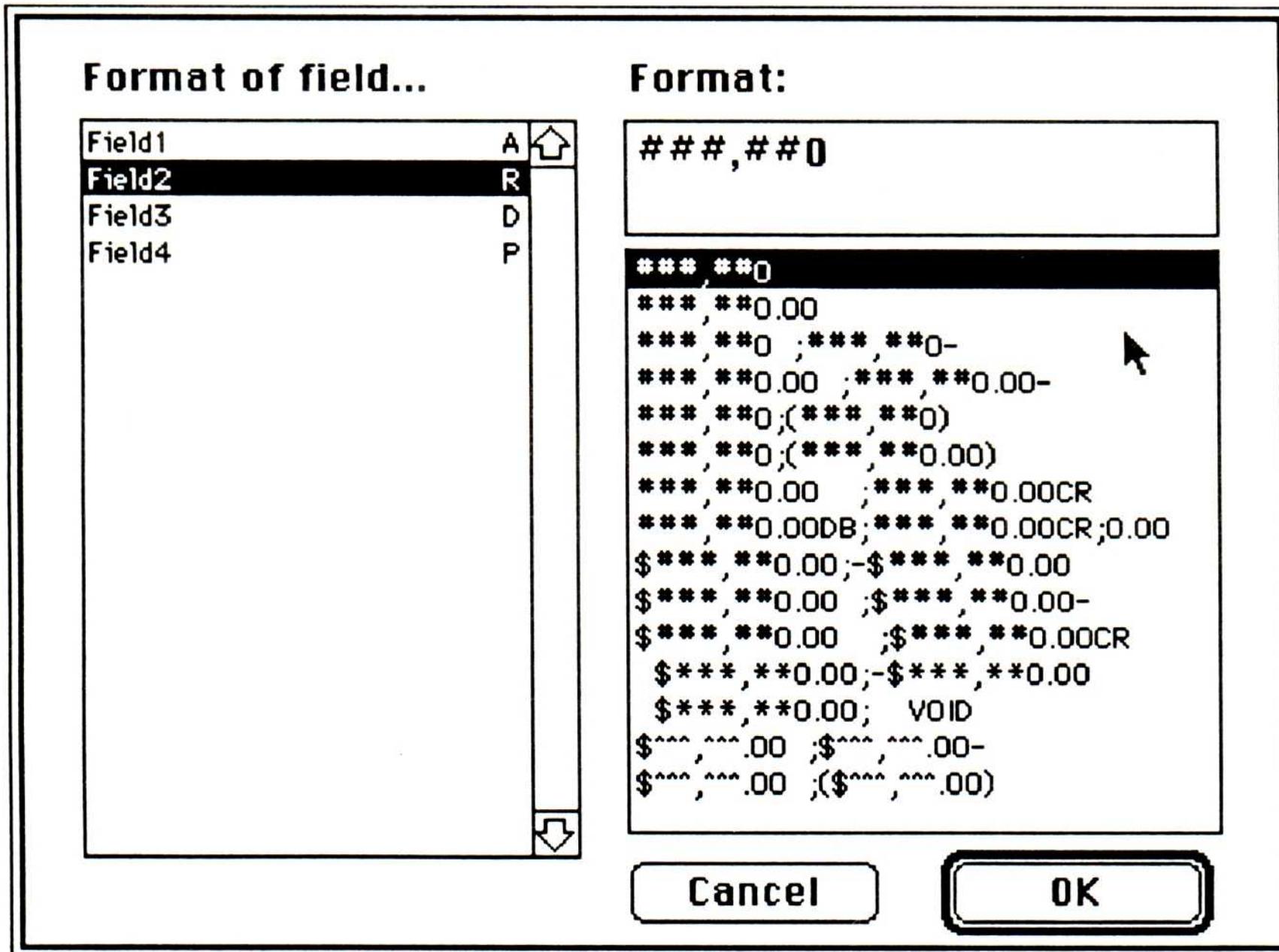


Figure 1
Format of field dialog

In the Tutorial on page 68, step 10 instructs you to add another # to the format. This is no longer necessary because of the new formats.

References:

Tutorial pages 66-69, 127
User's Guide pages 98, 107

Scientific notation

Scientific notation is now supported. Numbers can be entered with a positive or negative exponent. For example, 1,000,000 could be entered as 1e6 or 1e+6 and .001 could be entered as 1e-3.

Range of dates

Dates are now handled as Macintosh dates. This allows dates to be formatted by the International Utilities Package. The standard Macintosh date limits are January 1, 1904 to December 31, 2040. Years that are entered as two digits are automatically converted to this century. For example, the date 1/2/90 is interpreted as January 2, 1990.

References:

User's Guide pages 20, 337
Programmer's Reference pages 32, 38
Command Reference page 48

Searches on subfiles

Searches can now be made directly on subfiles using the search window. To search a subfield, click on the subfile name in the fields list. A list of the subfields will be presented. Click on the subfield you want to search.

A search on a subfile always returns a selection of records for the parent file. The selection contains the records that have one or more matching subrecords. The subselection for each record contains *all* subrecords, not just the matching subrecords.

References:

User's Guide page 186
Programmers Reference page 57

Line widths in the Layout editor

The line widths in the layout editor are now 1, 2, 3, and 4 pixels.

The 3 pixel line width allows you to draw a 3 pixel rounded corner box around the default button in a layout. The box should be drawn 1 pixel larger than the button. The default button in a layout should normally be the only Accept button.

References:

User's Guide pages 78, 84, 342

Menu editor keyboard equivalents

Keyboard equivalents have been added to the Menu editor.

Table 1

Menu editor keyboard equivalents

Menu item	Command key
Append Menu	R
Insert Menu	T
Append Item	F
Insert Item	G

Reference:

User's Guide page 344

Printing labels

A layout used for printing labels now uses the header and footer information. The header will print at the top of each page and the footer at the bottom. Since the LaserWriter cannot print all the way to the top and bottom of a page, this option allows you to skip the first and last rows of labels if necessary.

When using the ImageWriter you should set the Header to 0 and the Detail, Break, and Footer lines to the same position.

References:

User's Guide pages 196-197, 204-205

Recursive links

Recursive links are now represented by an oval and arrow pointing to the field with the recursive link. The display for a recursive link was a small bump on the side of a file box in the Design environment.

A recursive link is shown in Figure 2.

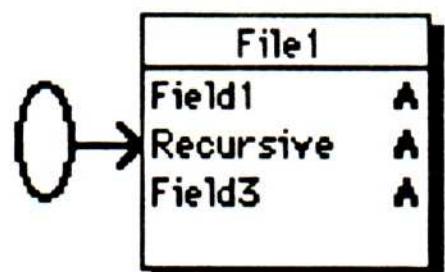


Figure 2
A recursive link

Reference:

User's Guide page 271

Procedure button in the layout dialog box

A button to open layout procedures is now included in the layout dialog box. To open a layout procedure, select the layout name and click the Procedure button.

Figure 3 shows the layout dialog box with the Procedure button.

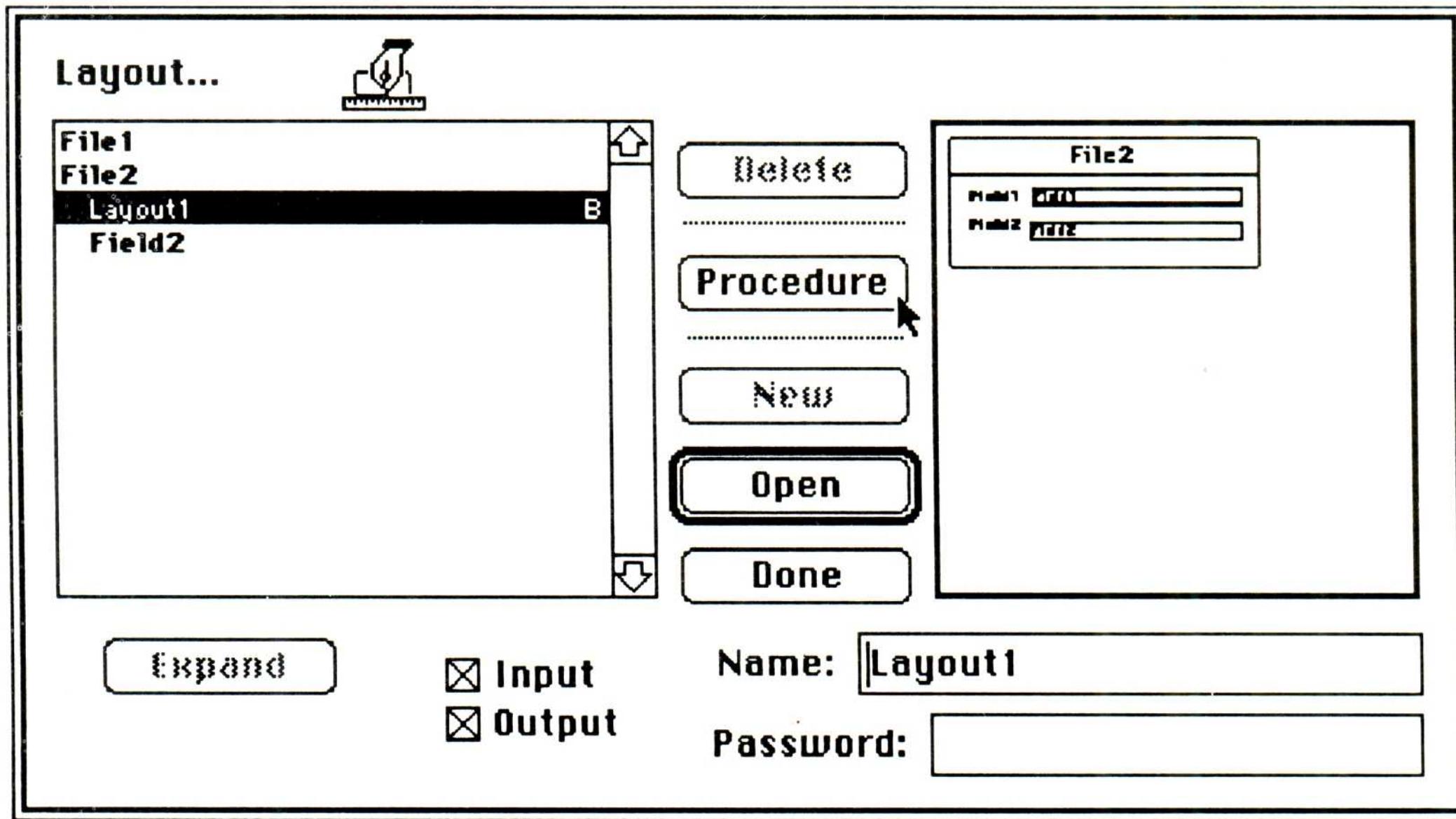


Figure 3
Layout dialog box

StartUp procedure

A global procedure may be created which will be executed whenever a database is opened. Any global procedure named StartUp will be executed when the database is first opened. This is useful for initializing variables, getting special information from the user, changing the current selection for a file by searching or sorting, or other processing that needs to be done only once.

Included layouts

Included layouts are used to display information from other files and to display and enter subfile data. The data is displayed in a scrollable area. You can create many styles for included layouts.

Following is a discussion of how an included layout is displayed and a description of the steps used to create a typical included layout.

How an included layout is displayed

An included layout is displayed within a scrollable area using the header, detail, and break lines as defined in the layout editor. This is done in the same way as records are displayed in the User environment. The following layout is for an included layout.

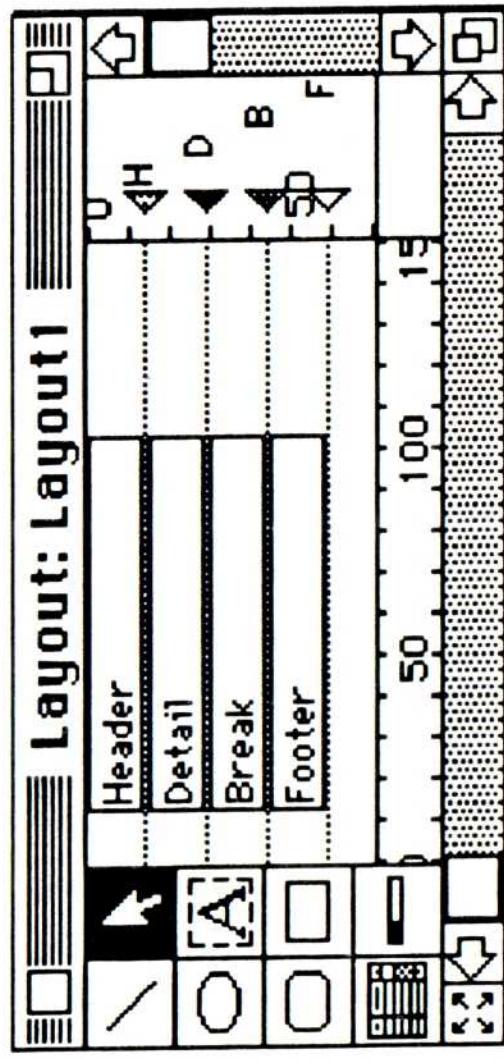


Figure 4
An included layout

A layout is created for the file in which the included layout will be displayed.

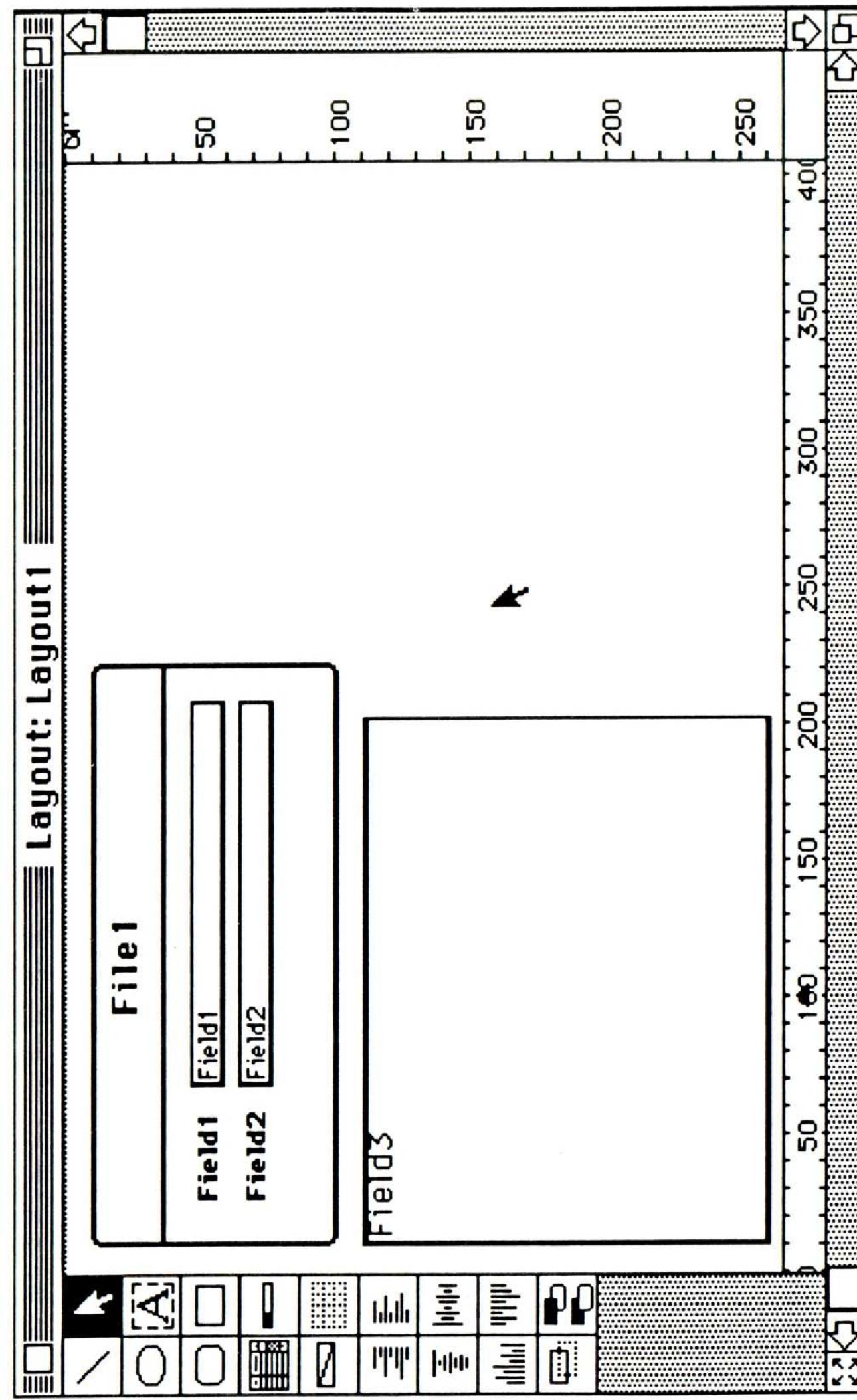


Figure 5
Layout that will display the included layout

The included layout area in figure 5 is labeled Field3 because it is from a subfile.

A frame was drawn around the included layout area. It is one pixel larger than the area except on the right side where it matches the included layout area border. You can create the frame by drawing a box or by selecting the included layout area and pressing Command-1, a shortcut for framing. If you use Command-1, remember to shrink the right side of the frame by one pixel.

Figure 6 shows the included layout during data entry.

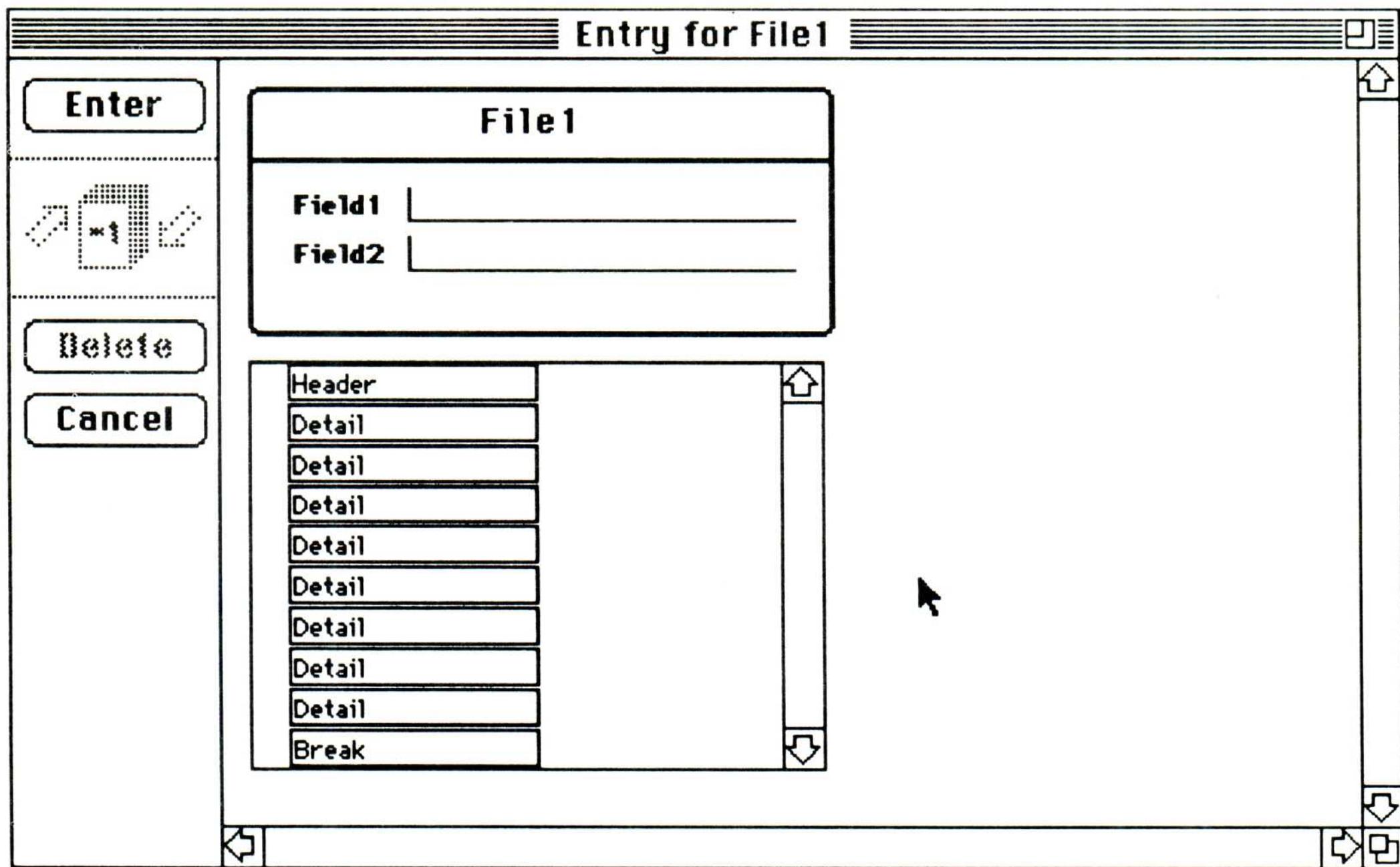


Figure 6

The included layout during data entry

4D xRef Utility

The 4D xRef utility prints a cross reference of a 4th Dimension database. Reports can also include the complete database design as defined in the Design environment of 4th Dimension.

A full report includes

- the structure description for each file
- the file procedure listings and flowcharts
- the layouts for each file
- the layout procedure listings and flowcharts
- the global procedure listings and flowcharts
- the password system
- the menus
- a cross reference listing

Starting 4D xRef

To start 4D xRef:

1. Double-click the 4D xRef icon or select the icon and choose Open from the File menu of the Finder.
4D xRef displays a standard Open File dialog box that allows you to select the existing database you want to print.
2. Open the database.
3. If the database has a password system the password dialog is displayed. Enter the password for the top level in the password system. Click OK to open the database or Cancel to quit.

Figure 7 shows the password dialog.

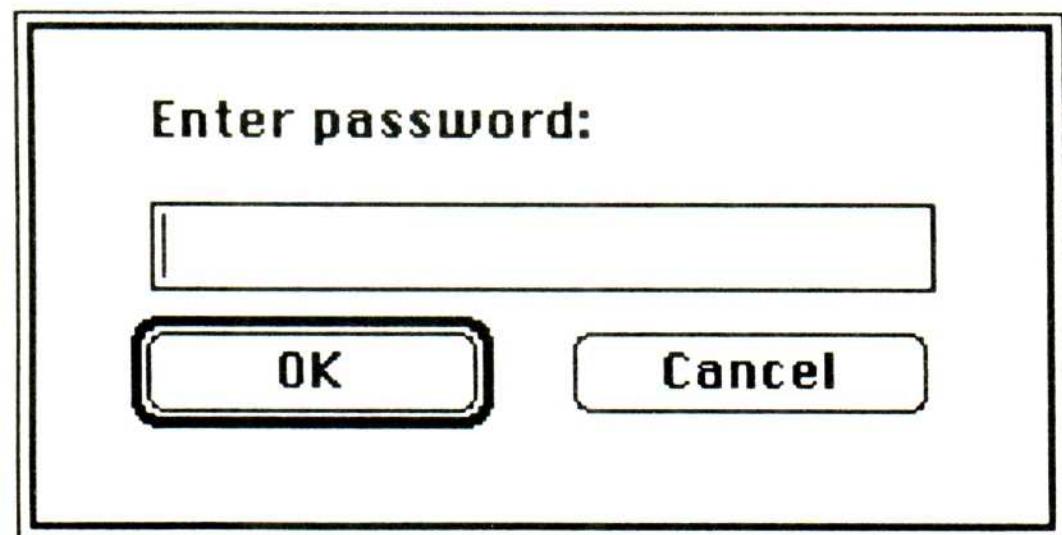


Figure 7
Password dialog box

Important

You cannot print a cross reference for a database without the top level password.

After opening a database the 4D xRef dialog is displayed. Use the dialog to set up the report to your needs. Click Print to print the report.

Figure 8 shows the 4D xRef dialog.

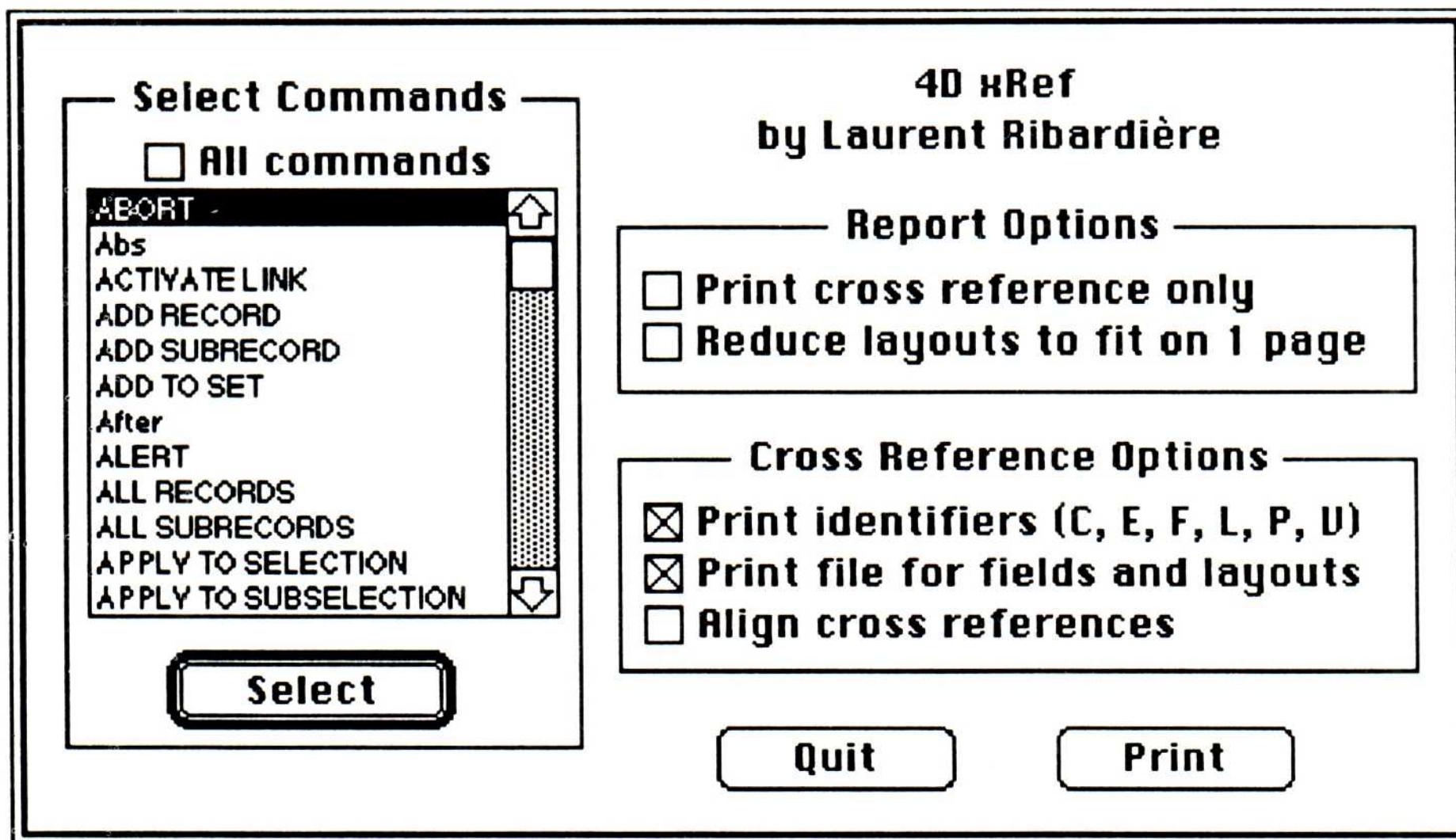


Figure 8
4D xRef dialog box

Report Options

Two options allow you to control the style of the report.

- **Print cross reference only** will prevent the database description from being printed. Only the cross reference will be printed.
- **Reduce layouts to fit on 1 page** will compress any large layouts so that they will fit on a page. If this option is not selected, large layouts will be printed on multiple pages.

Cross Reference Options

Three options control the style of the cross reference.

- **Print identifiers (C, E, F, L, P, V)** prints an identifier after each item in the cross reference.

Table 2
Cross Reference Identifiers

Identifier	Description
C	Command
E	External procedure
F	Field
L	Layout
P	Procedure
V	Variable

- **Print file for fields and layouts** prints the filename after all fields and layouts. For example, if Layout1 and Field1 belong to File1 then they would be printed as:

Layout1 [File1]
Field1 [File1]

- **Align cross references** will align the cross reference listing.

Selecting commands for the cross reference

You can select all, some, or none of the 4th Dimension commands to be included in the cross reference. Commands that are selected will be printed followed by the procedures in which they are used.

You select the commands in the “Select Commands” area. To include 4th Dimension commands in the cross reference, double-click on a command or select a command and click the Select button or press Return. A selected command will show a small plus sign in the right side of the commands list. To deselect a selected command, double-click on the command or select the command and click the Select button or press Return.

To include all commands in the cross reference, click on the “All commands” check box. Selecting “All commands” will override any selection you have made in the commands list.

Printing the cross reference

To print the report click on Print. 4D xRef will present the two printer dialogs for your printer. After clicking OK in both printer dialogs the report will be printed. The report is printed with a header on each page that includes the date, the title for that page (procedure name, filename, etc.), and the page number.

A cross reference includes all fields, layouts, variables, procedures and the 4th Dimension commands that you selected. Each is printed in bold on the left side of the page. Following each is a list of the procedures and layouts that they are used in.

Reports for complex databases can be quite large and take a long time to print. Remember that unless you select “Print cross reference only” the complete database design will be printed.

A report example

The following is an example cross reference listing for a simple one file database. The first two columns are generated by the cross reference listing. The third column is an explanation of the information in the cross reference listing.

The cross reference was printed with the following options selected:

- Print cross reference only
- All commands
- Print identifiers (C, E, F, L, P, V)
- Print file for fields and layouts
- Align cross references

Monday, January 1, 1990

Cross Reference Listing

ALERT (C)	[File1] (P) Procedure2 (P)	ALERT is a command used in the file procedure for File1 and the global procedure Procedure2.
BEEP (C)	[File1] (P) Layout1 [File1] (P) Procedure1 (P)	BEEP is a command used in the file procedure for File1, the layout procedure for Layout1, and the global procedure Procedure1.
Clock (E)	Layout1 [File1] (P) Procedure1 (P)	Clock is an external procedure used in the layout procedure for Layout1 and Procedure1.
Field1 [File1] (F)	Layout1 [File1] (L) Layout1 [File1] (P)	Field1 is a field in File1 that is used in Layout1 and the layout procedure for Layout1.
Field2 [File1] (F)	Layout1 [File1] (L) Layout1 [File1] (P)	Field2 is a field in File1 that is used in Layout1 and the layout procedure for Layout1.
INPUT LAYOUT (C)	Procedure1 (P)	INPUT LAYOUT is a command used in Procedure1.
Layout1 [File1] (L)	Procedure1 (P) Procedure2 (P)	Layout1 is a layout of File1 that is used in Procedure1 and Procedure2.
Procedure2 (P)	Procedure1 (P)	Procedure2 is a procedure used in Procedure1.
vCancel (V)	Layout1 [File1] (L)	vCancel is a variable used in Layout1.
vClock (V)	Layout1 [File1] (L) Layout1 [File1] (P)	vClock is a variable used in Layout1 and the layout procedure for Layout1.
vOK (V)	Layout1 [File1] (L)	vOK is a variable used in Layout1.
x (V)	Procedure1 (P)	x is a variable used in Procedure1.
[File1] (F)	Procedure1 (P) Procedure2 (P)	File1 is a file used in Procedure1 and Procedure2.

4D External Mover

4D External Mover is an enhanced version of 4D Mover. The new 4D External Mover now moves resources that are used by external procedures.

This section replaces the section on 4D Mover in the 4th Dimension Utilities and Programmer's Notes pages 15-23.

4D External Mover creates 4th Dimension external procedures from 68000 family object code, and transfers them between files much the way that the Font/DA Mover transfers fonts and desk accessories. An external procedure is a program written in any high level language such as Pascal or C and compiled to 68000 family code, or written in 68000 Assembly. Using 4D External Mover, you can add that procedure to 4th Dimension or your application.

Warning

Before you use the 4D External Mover, you should read Appendix D on external procedures in the *4th Dimension Command Reference*. 4th Dimension hands over complete control to an external procedure.

4D External Mover works with five types of files:

- **4th Dimension program:** You can add your own external procedures to your copy of the 4th Dimension program. There, they are available to any application running under the modified copy of the program.
- **4th Dimension runtime version:** You can add external procedures to the runtime version, as you can to the 4th Dimension program.
- **4th Dimension database application “.Res” files:** Adding external procedures to the database application “.Res” file makes them available only to that database application, but available when running under any copy of 4th Dimension or the runtime package.
- **68000 family object code files:** These are compiled and linked application files that contain a “CODE” resource that 4D External Mover converts into 4th Dimension external procedures..
- **External procedure Library files:** A Library file acts as a “holder” for one or more external procedures—much like a Font/DA Mover “suitcase” file. 4D External Mover creates Library files. A library file named “Proc.ext” file can exist in a database folder or in the system folder of the boot disk. Adding external procedures to a “Proc.ext” file in a database folder makes them available to that database. Adding them to a “Proc.ext” file in a system folder makes them available to all databases run under that system.

Preparing

Appendix D in the *4th Dimension Command Reference* contains complete guidelines for creating an external procedure. In summary, they are

- 4th Dimension hands over complete control to the external procedure. The procedure must “know” what to do, and “clean up” before it hands control back to 4th Dimension.

- The entire code must be one segment.
- All parameters in your procedure must be variable parameters. Parameters in your procedure and as described to 4D External Mover must be the same in type, order, and number.
- 4D External Mover copies a CODE segment from your compiled application, and pastes it into a resource 4DEX that belongs to the file in which you intend to put the external procedure. Depending on your compiler and linker, the CODE may begin with jump table information. 4D External Mover can remove these bytes before pasting the code in 4DEX.

Starting

Start 4D External Mover from the Finder.

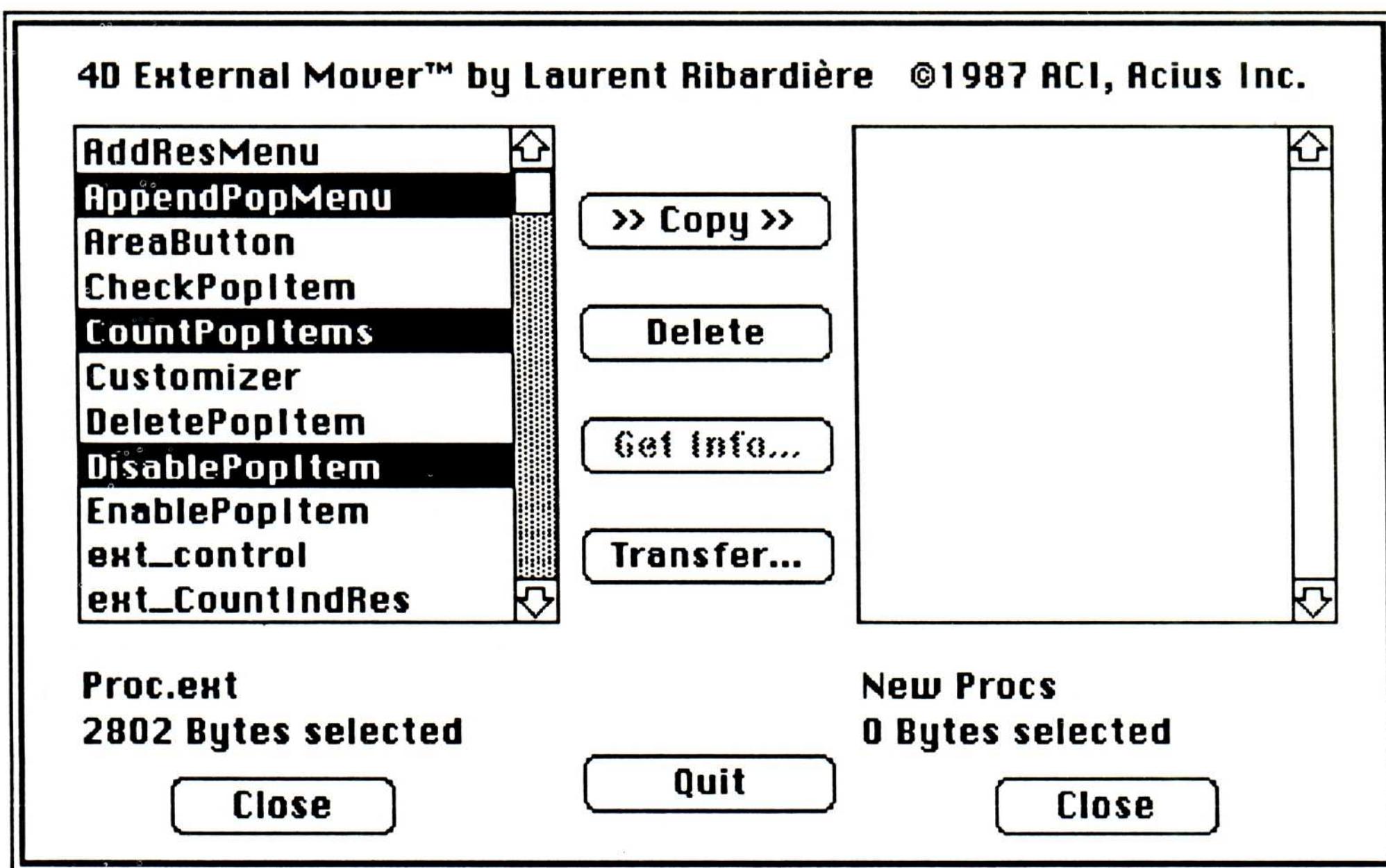


Figure 9
4D External Mover main screen

Figure 9 shows 4D External Mover's main screen. There are two files open, Proc.ext on the left and New Procs on the right. In Proc.ext, three external procedures have been selected and the size of the selected procedures and any bundled resources are shown below the file name. As with Font/DA Mover, you may Open a file on either side of 4D External Mover first.

Creating a Library file

Library files hold one or more procedures *that have already been converted* for use with 4th Dimension. They are like the “suitcase” files that hold DA’s and Fonts. New Library files are, of course, empty.

To create a new Library file:

1. Start 4D External Mover.
2. Click Open.

4D External Mover displays the open file dialog.

3. Click New.

4D External Mover displays a dialog box in which you can name the new Library file. Figure 9 shows this dialog box. 4D External Mover proposes the name “Proc.ext” as a default Library file name.

3. Click Save.

This creates the new Library file and opens it in one side of 4D External Mover’s main screen.

You may now copy external procedures into the Library file.

Transferring code to an external procedure

After compiling and linking your source code, you can transfer it into an external procedure using 4D External Mover. Transferring an application is a different action than adding the external procedure to 4th Dimension, the runtime version, an application, or to a Library file.

Important

4D External Mover does not cannibalize the original code. You can safely transfer any appropriate code—4D External Mover will leave the original code intact.

To transfer code to an external procedure:

1. Start 4D External Mover, and click Open (either side).

4D External Mover displays a standard Open dialog box. The dialog box recognizes only applications and Library files. To open other files, for example database .res files, hold down the option key while clicking Open.

2. Open the destination file for your conversion.

Often, this will be a Library file, but may be the 4th Dimension, runtime, or database application. It’s generally easier and faster to convert several applications into external procedures and house them in a Library file than to convert them directly to the 4th Dimension program.

4D External Mover will list the external procedures in the file.

3. Click Transfer. 4D External Mover displays the Move a code segment dialog shown in Figure 10.

Enter the CODE segment number you want to transfer in the first box. Enter the number of bytes that should be stripped from the beginning of the code in the second box.

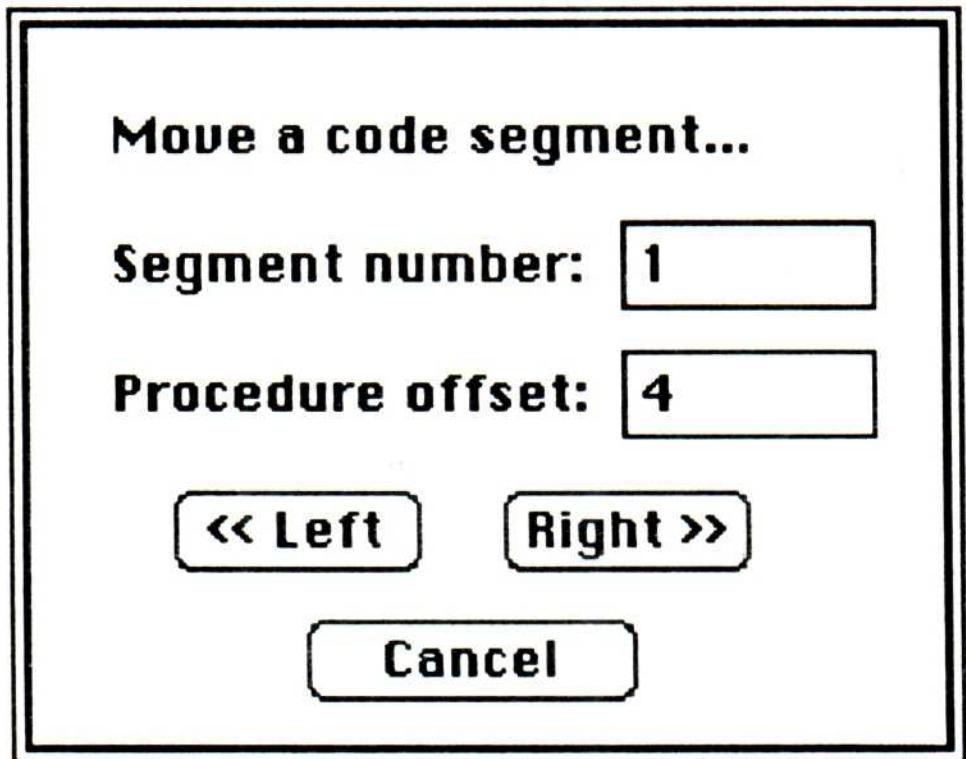


Figure 10
Move a code segment dialog

4. Click the side to which you want the procedure moved.

4D External Mover displays the Open file dialog.

5. Open the application you want to transfer.

4D External Mover displays a screen on which you can enter the parameters of your procedure (and their order), add comments, and enter the procedure name. Figure 11 shows this screen.

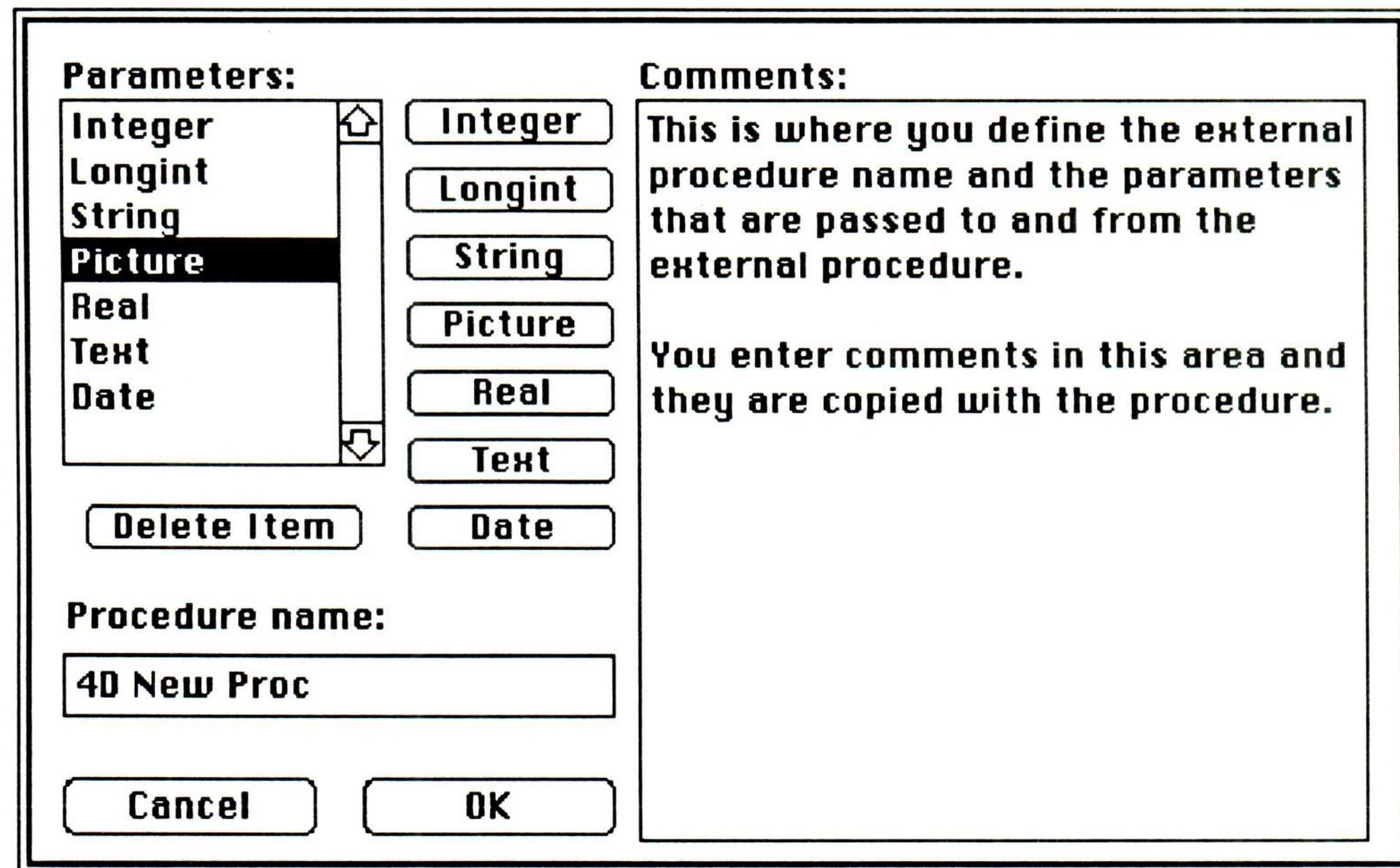


Figure 11
Parameters, comments, and procedure name

6. When you're finished entering parameters, comments, and the procedure name, click OK or Cancel.

Entering parameters

You must enter the proper number and type of parameters that you have in the declaration block of your procedure. The parameters must also be in the same order as they are in your procedure source code. 4D External Mover does not need to know the parameter names—just their number, type, and order.

To create a list of parameters matching that in your procedure:

- Click a parameter type from the list of seven buttons to the right of the parameter list box.
- Click a parameter in the list to select it. Click below the existing parameters, but inside the list box, to select no parameter.
- 4D External Mover always adds a new parameter above any selected parameter.
- If there is no selected parameter, 4D External Mover adds the new parameter to the end of the list.
- To delete a parameter, select it and click Delete Item.

Entering comments

You can enter information about a procedure in the Comments area. 4D External Mover will display these comments (and the rest of the parameter and procedure name information) when you click the Get Info button on 4D External Mover's main screen.

To enter comments:

- Begin typing. 4D External Mover will enter the characters you type in the Comments area. Text will wrap automatically.
- Press Return for a combined line feed and carriage return; the Enter key is the equivalent of clicking OK.

Entering a procedure name

The name for the external procedure that you enter into the Procedure Name area need not match the name that the procedure had in its source code form. However, the name you give it here is the name you must use when calling the procedure in 4th Dimension.

The name of an external procedure may be up to 15 characters long. This is the same restriction as for procedure names within 4th Dimension.

Copying external procedures

Once you have converted code to an external procedure, you may transfer it from file to file and from one copy of 4th Dimension to another. Copying an existing external procedure from a Library file (for example) to a copy of 4th Dimension is the same as installing it in that program.

To copy an external procedure:

1. Start 4D External Mover and open a file containing at least one external procedure. You may double-click a Library file to both start 4D External Mover and open the file.
2. On the other side of 4D External Mover's main screen, open a destination file. This may be 4th Dimension, 4D Runtime, or a Library file. To open a database.res file, press the Option key while clicking Open.
3. Select the external procedures you wish to copy from the source file.

The **Copy** button activates. Angle braces show which way the copy will proceed.

Click to select an individual procedure, Command-click to select non-contiguous procedures, and Shift-click to select contiguous procedures.

4. Click **Copy**.

4D External Mover copies the procedure from one location to the other.

Deleting an external procedure

To delete external procedures from a file, open 4D External Mover and the file, select the procedures you wish to delete, and click **Delete**.

4D External Mover will confirm that you do, indeed, wish to delete the selected items. Click **Yes** to continue, **No** to stop.

Command button summary

This list summarizes the command buttons from the main screen of 4D External Mover and what they do.

- **Copy** transfers an existing external procedure from one file to another. When you transfer an external procedure to 4th Dimension, 4D Runtime, or a database application, it is equivalent to installing it.
- **Delete** deletes the selected external procedure from its file.
- **Get Info** displays the 4D External Mover screen showing parameters, comments, and procedure name for the selected procedure. You may change any or all of the three items.
- **Transfer** opens the application file you want to convert into an external procedure, and displays the 4D External Mover screen showing parameters, comments, and procedure name.
- **Quit** exits 4D External Mover.
- **Open/Close** opens and closes source and destination files that hold external procedures.

Bundling resources with 4th Dimension external procedures

The following section describes 4th Dimension capabilities that are of interest to advanced developers. You should be comfortable with pascal and Macintosh resources before reading this section.

4th Dimension provides support for bundling resources with external procedures. 4D External Mover will copy resources that are bundled with external procedures.

External procedures can use resources. For instance, putting up a modal dialog will require a "DLOG" resource and a "DITL" resource. If the resource ID of the DLOG is 5000, for example, then there can be some potential problems. This resource ID may be used by 4th Dimension, so there will be a conflict, or some other external procedure may use its own DLOG with the ID equal to 5000.

To solve these problems, a resource bundle (resource type 4BND) is used. Each external procedure that uses resources must have a bundle that contains a list of all the resources the procedure uses, a local resource ID for each resource, and a global resource ID number. In the external procedure, the local ID is used to determine the global ID of the resources. Thus, if the global ID numbers of an external procedure conflict with existing resources, then the mover changes the global ID numbers. Because the external procedure only uses the local ID numbers, it can always access the resources it needs.

The mover also takes care of owned resources. For instance, if there is a DLOG with ID equal to 5000 and it uses a DITL of ID 5000, then when the mover needs to change the DLOG resource ID to 15000, it will also change the DITL ID to 15000. The mover will also change all the resources used in the DITL, and so on, so there is no loss of resources when they are moved.

The resources are bundled through a new resource type, 4BND. To create bundle resources easily, you can first install the TMPL resource template into your copy of ResEdit. 4BND resources bind your external procedures and resources together so that 4D External mover can move all the resources required by your external procedure together.

To install the ResEdit template for the 4BND resource:

1. Start ResEdit.
2. Open the file "ResEdit Template for 4BND" on the Utilities Disk.
3. Select the TMPL resource in the Template file.
4. Select Copy from the Edit menu.
5. Close the Template file.
6. Open ResEdit.
7. Select Paste from the Edit menu to paste the new TMPL resource into ResEdit.
8. Close ResEdit, saving the file when prompted.

You can now use ResEdit to create a new 4BND resource. Open the file that contains your 4DEX resources. Create a new 4BND by selecting New from the file menu and choosing 4BND in the dialog.

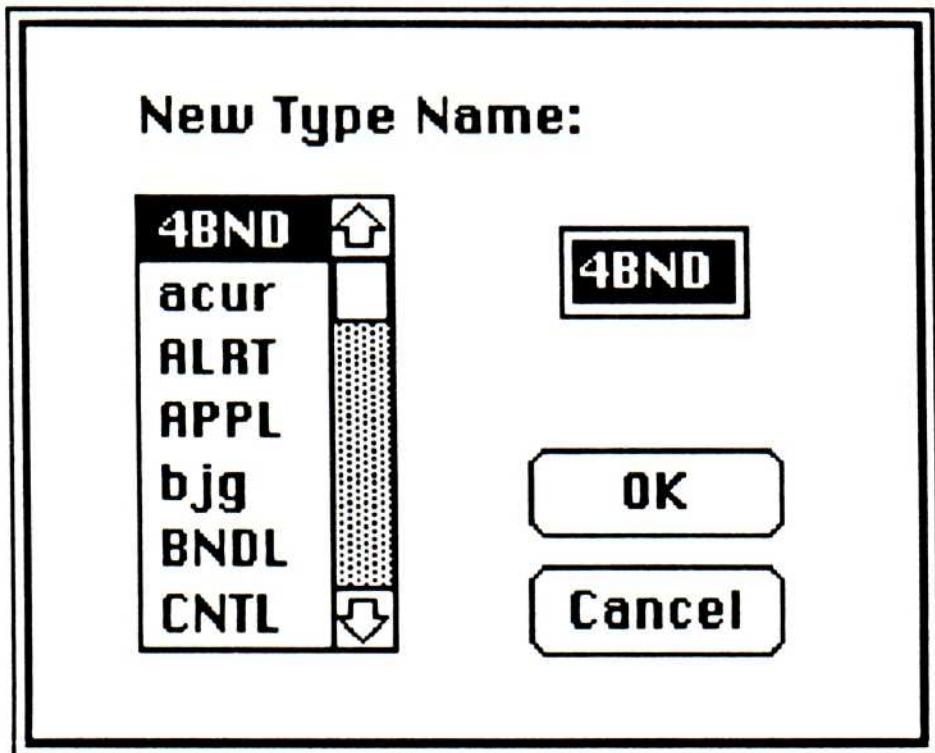


Figure 12
ResEdit dialog to create a new resource

After opening the new 4BND type, again select new to create a new 4BND. Select the 5 asterisks and select New again. You will have two new entries to enter your first bundled resource. The resource ID of the 4BND should be the same as its associated external procedure and should be greater than 15000.

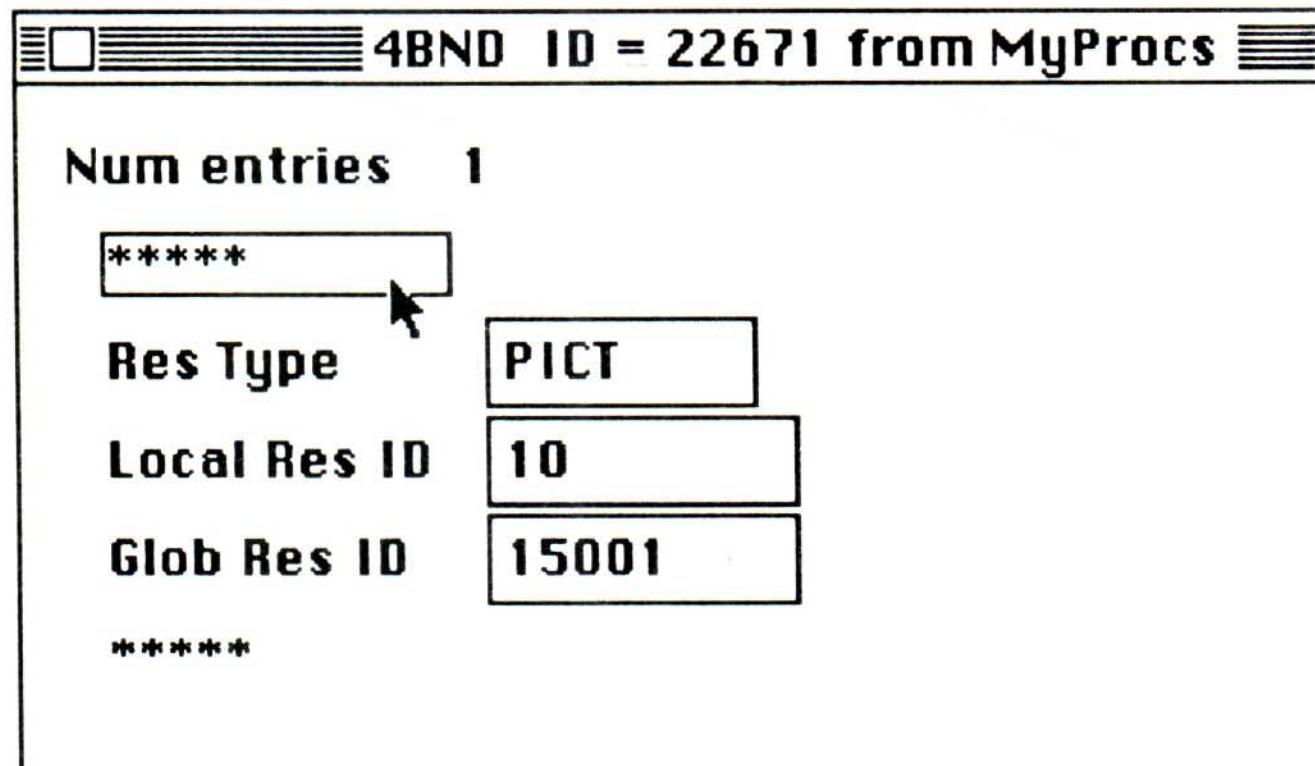


Figure 13

ResEdit dialog to create a new 4BND resource

In the 4BND resource you enter the resource type in the first area, Res Type. This is always a 4 character string. The Local Res ID is the resource number that you've used to access the resource from within your external procedure. The Glob Res ID is the actual resource number of the bundled resource.

In Figure 13 a new 4BND has been created. It will allow an external procedure to access a PICT resource using a local ID of 10. The PICT resource is actually stored in a resource with ID 15001.

The 4BND is a resource list. To add more resources to the list, select the 5 asterisks and select New from the file menu.

Important

Resources that you bundle with your external procedures should always have a resource ID of 15000 or greater, so that there is no conflict with any 4th Dimension resources .

Example using bundled resources

The following example is an external procedure written in MPW Pascal. The external procedure takes an integer as an argument which is the local ID of the new cursor. The cursors are stored in CURS resources that are bundled with the external procedure through a 4BND resource.

Figure 14 shows the 4BND resource that was created to bundle two cursors with the external procedure.

4BND ID = 30264 from Example	
Num entries	2

Res Type	CURS
Local Res ID	0
Glob Res ID	20001

Res Type	CURS
Local Res ID	1
Glob Res ID	20002

Figure 14

4BND Resource for the example

Since the 4BND ID is 30264 the ID of the external procedure (4DEX resource) is also 30264.

The local numbers for the cursors are 0 and 1. These are the numbers that you would use when calling the procedure from 4th Dimension. The global numbers, 20001 and 20002 are the actual resource numbers of the cursors. Remember that the global resource numbers may be changed by 4D External Mover. You should always access the resources through the local numbers.

Example:

```
program OtherCurs;
Uses Memtypes, Quickdraw, OSIntf, PasLibIntf, ToolIntf, PackIntf;
{$R-}
{$D+}

var DummyInt: integer;
```

Include the following code in your procedure

{ The CurProcResid function is assembly code that returns the 4BND resource number. }

```
Function CurProcResid: integer;
INLINE $3EAD, $DE4E;

Function GetResNum (T: ResType; Localid: integer): integer;
Forward;
```

{ This is the example procedure. It sets the cursor based on the local resource number it receives. }

```
procedure ChangeCurs (var WhichCurs: integer);
begin
  SetCursor (GetCursor (GetResNum ('CURS', WhichCurs) )^^ );
end;
```

Include the following code in your procedure

{ Pass GetResNum the resource type and local ID of the resource and it returns the global ID. }

```
Function GetResNum;
Type
  ResEntry = record
    Typ: ResType;
    LocalNum, GlobalNum: integer;
  end;
  ResTab = record
    nbEntry: integer;
    Tab: array [1..1] of ResEntry;
  end;
  ResTabPtr = ^ResTab;
  ResTabHandle = ^ResTabPtr;
var i: integer;
  r: ResTabHandle;
  Done: boolean;
begin
  GetResNum := 0;
  Done := false;
  i := 1;
  r := pointer (GetResource ('4BND', CurProcResId));
  if r <> Nil then
    begin
      while ((i <= r^^.nbEntry) and (not (Done))) do
        begin
          with r^^.Tab[i] do
            begin
              if (t = Typ) and (LocalNum = Localid) then
                begin
                  GetResNum := GlobalNum;
                  Done := true;
                end;
            end;
          i := i + 1;
        end;
    end;
  end;


---


begin
  ChangeCurs (DummyInt);
end.
```

4D Customizer

This section replaces the section on 4D Customizer in the 4th Dimension Utilities and Developer's Notes. 4D Customizer has been enhanced to make it easier to use. It now uses a single dialog instead of a series of menus.

With 4D Customizer, the developer can fine-tune the behavior of 4th Dimension and its applications. The 4D Customizer groups attributes this way:

- Enabling the pointer to move during procedure processing.
- Enabling desk accessories during printing or while 4th Dimension is displaying its progress "thermometers".
- Selecting new keys for data entry.
- Altering the Custom window size and type and its behavior while the application is running.

The 4D Customizer will customize the 4th Dimension program, 4D Runtime, a database itself or the file Proc.ext which can reside in the system folder. It modifies (or creates, as necessary) the CUST resource in the program or runtime or in the ".Res" file of a database.

Important

The CUST resource in a Proc.ext or ".Res" file of a database always overrides the program's CUST resource.

Warning

The 4D Customizer modifies the actual 4th Dimension program file or database. It is a wise policy to make a backup before altering it.

Starting 4D Customizer

Start 4D Customizer from the Finder.

Figure 15 shows how 4D Customizer looks when it starts.

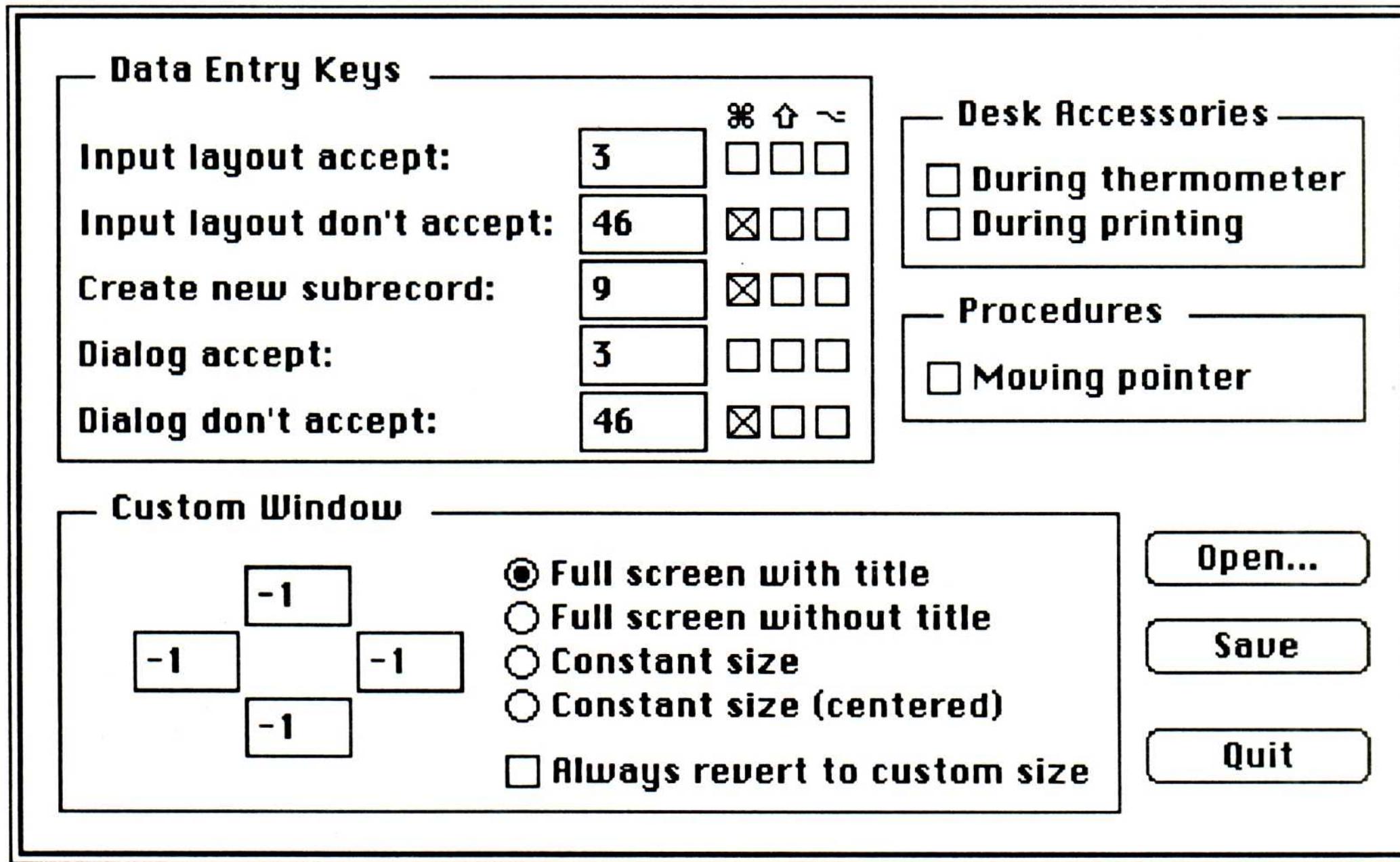


Figure 15
How the Customizer appears when it starts

There are four areas that control the customization.

- **Data Entry Keys** sets the keys and their modifiers that will be used during data entry as keyboard equivalents.
- **Custom Window** controls the appearance of the default window in the custom environment or the runtime version.
- **Desk Accessories** controls whether desk accessories can be accessed during certain 4th Dimension operations.
- **Procedures** allows monitoring of procedure execution through the use of a moving pointer.

In addition there are three buttons.

- **Open** displays a standard Open dialog box. Use the Open dialog box to select a copy of 4th Dimension, 4D Runtime, or a database's ".Res" file that you wish to customize.
- **Save** saves the changes you have made to the file you opened.
- **Quit** returns you to the Finder. You must save your changes before quitting.

To customize 4th Dimension:

1. Start 4D Customizer.
2. 4D Customizer will present the Open file dialog. Choose the copy of 4th Dimension, 4D Runtime, or the database ".Res" file that you wish to modify.
3. Make the changes you wish.
4. Click the Save button and Quit the program.

The following sections discuss the customization changes you can make.

Data entry keys

4th Dimension provides keyboard equivalents for clicking OK, clicking Cancel, and for adding a new subrecord in an included layout. Table 3 shows the program's default keyboard equivalents for these tasks.

Table 3
Data entry keyboard equivalents

Action	Keyboard equivalent
Clicking OK	Enter
Clicking Cancel	Command-. (period)
Adding a new subrecord	Command-Tab

To change the keyboard equivalents:

1. Enter the ASCII code of the key you want as keyboard equivalent.
2. If you want a modifier for the key check one of the three check boxes representing the potential modifiers: Command, Shift, or Option.

Important

If you enter an ASCII code that is generated with the use of a modifier key such as Command, Shift or Option then make sure to check the modifier boxes also.

For example, if you want the keyboard combination of Option-Tab to signal acceptance of a record (equivalent to clicking an OK button), you would enter the ASCII code 9 in the box and click the Option check box.

Important

Be wary of making the Return key the equivalent of clicking OK. If you do, you will not be able to use Returns in Text fields, or be able to use the Return key to move from field to field within a layout.

Setting the window in the Custom environment

You can control how the input window appears in the Custom environment. The options are

- **Full screen with title** produces a full size window displaying the application name in the title bar. This is the default choice.
- **Full screen without title** produces a full size window with no title bar.
- **Constant size** uses the four values to set window margins.
- **Constant size (centered)** uses the four values to set the window margins but always centers the window on the screen.

Important

The -1 values in the window margin boxes are 4D Customizer's default values, but do not in themselves produce a standard size window in 4th Dimension.

- **Always revert to custom size** affects how the window will behave when switching between the User and Custom environment. By default 4th Dimension will retain any changes the user makes to the window size and position when switching between the User and Custom environments. If this option is selected, 4th Dimension will always use the custom window settings when switching environments.

Enabling Desk Accessories

As a default, 4th Dimension disables desk accessories during printing and when displaying its progress thermometers on screen. Thermometers give you a graphic idea of how much you've processed and how much there is yet to go.

You can enable the desk accessories by checking the boxes "DA's Enabled During Printing" and "DA's Enabled During Thermometers".

Displaying a moving pointer during procedure execution

When 4th Dimension is executing a procedure it normally displays the standard arrow pointer. You can change this so that it displays a rotating disk—called "the beach ball"—instead. The beach ball pointer can be useful during development to give you some feedback when your procedures are executing.



Figure 16
The beach ball

You can add the beach ball to your own applications by checking "Moving pointer" in the Procedure area.

