Uniplex

Form-Building Tools Guide

Manual version: 9.20
Document version: V1.0

THIS PAGE INTENTIONALLY LEFT BLANK

Manual version: 9.20
Document version: V1.0

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

INTRODUCTION **1**

Overview 3
- Screen Builder 3
- Formfill 3
- Customized Database Forms 3

CHAPTER 1 SCREEN BUILDER **5**

Accessing the Screen Builder 7

Creating a Screen 9
- Example Screen Template 10
- Create a New Screen 12
- Enter the Title of the Screen 12
- Enter the Layout of the Screen 12
- Enter the Field Definitions 13
- Map User Input to Operating System Command Arguments 20
- Enter the Command Line 21
- Enter Screen Help 23

Building a Screen 23

Configuring a Screen into the Menu System 24

CHAPTER 2 FORMFILL **25**

Accessing Formfill 27

Creating a Formfill 28
- Create a Formfill 28
- Edit a Formfill 29
- Build the Screen and Print Forms 30

Modifying a Formfill 31
- Modify a Screen Formfill 31
- Modify a Print Formfill 32
- Include Format Information 33
  - Common Formats 36

CHAPTER 3 CUSTOMIZED DATABASE FORMS....39

Accessing Customized Forms....41
Creating a Customized Form using the Standard Form....43
Create a Standard Form....43
Standard Form Template....44
Change the Title of the Form....45
Change the Screen Layout....45
Change Field Access and Use....46
Creating a Customized Form Without using a Standard Form....51
Customized Form Template....52
Create a New Form....53
Enter the Title of the Form....53
Enter the Layout of the Form....53
Enter the Field Definitions....54
Label Definitions....55
Specify a Default Entry in a Field....56
Specify a Display Attribute for a Character Field....57
Specify the Allowable Entries in a Field....57
Display Prompts....59
Joining Tables....59
General....62
Specify Access....62
Add Comments to a Form....63
Add Help to a Form....63
Enter the End Section....64
Syntax Rules....64
Build a Form....64
Edit a Form....65
Use a Customized Form....65

Introduction

THIS PAGE INTENTIONALLY LEFT BLANK

Overview

Uniplex provides three form-building tools for you to create and build your own customized forms.

This guide describes how you can use each of these tools to create different types of forms and screens. An overview of each tool is given below; for complete details refer to the remaining chapters of this guide.

Screen Builder

You use the Uniplex Screen Builder to provide a screen interface for operating system commands and applications. The screens you can create enable inexperienced users to carry out operating system tasks without needing detailed knowledge of the operating system.

Formfill

You use Uniplex Formfill to design and complete the forms used in your office. You can create forms for any office application, such as invoices, purchase orders, expense reports and pay checks.

Note: The Formfill chapter in this guide explains how to create and maintain forms on your computer system. See the Formfill chapter in the Advanced Office System User Guide to find out how to complete and print out forms that you have designed.

Customized Database Forms

You can create a customized form for use with a database table and may include information on this form to help with its use, for example screen information and prompts. You can also restrict access to the table from the form. In addition, you can create customized forms that access more than one table at a time.

When you have created a customized database form, you can further customize the system by providing direct access to the form. In this way, you provide a fast method of data entry for data entry operators.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Screen Builder

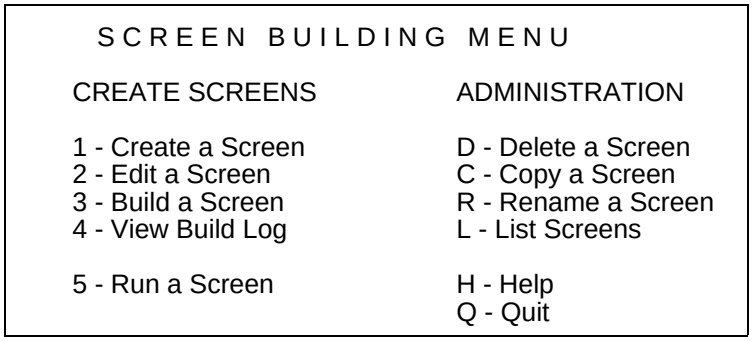
THIS PAGE INTENTIONALLY LEFT BLANK

Accessing the Screen Builder

To access the Screen Builder:

o Select the Screen Builder option from the Uniplex Main Menu.

  Uniplex displays the Screen Builder menu as follows:



You can use the Screen Builder to create simple screen interfaces for various tasks. For example, you could provide screens for the following:

o File Management Tools

  For example, provide screens that copy files from one device to another, read and execute files stored on disk or tape, and make system backups.

o Security and Access

  For example, provide screens that set access permissions on files and directories or check who is using the computer system.

You can use the Screen Builder to:

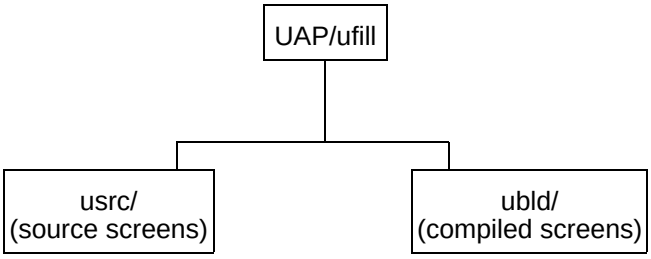
o Structure a command based on user input into field entries or selections in a screen.

o Execute a screen, passing parameters to another screen.

o Recursively execute a screen.

o Create multiple "paged" screens.

Screens are stored and executed either from the central UAP/ufill directory, or from the users own $HOME/UAP/ufill directory. In either case, the directory structure is as follows:



Creating a Screen

To create a screen, follow these steps:

1 Design the screen. See the next section Example Screen Template.

2 Create the screen. See the later section Create a New Screen.

3 Enter the title of the screen. See the later section Enter the Title of the Screen.

4 Enter the layout of the screen. See the later section Enter the Layout of the Screen.

5 Enter the field definitions. See the later section Enter the Field Definitions.

6 Optionally, map user input to operating system command arguments. See the later section Map User Input to Operating System Command Arguments.

7 Enter the command line(s) that the screen will execute. See the later section Enter the Command Line.

8 Optionally, enter help for the screen. See the later section Enter Screen Help.

9 Build the screen. See the later section Building a Screen.

10 Configure the screen into the menu system. See the section Configuring a Screen into the Menu System.

See the next section, Example Screen Template, for how a screen template is made up of sections. Then follow the detailed descriptions of how to create these sections.

Example Screen Template

:TITLE *title name*		Identifies the title of the screen.
:SCREEN		Start of screen layout.
free text **[***field name*_____**]**		Free text and field definition.
:LABELS		Start of label definitions.
field name = definition		Definition can include:
	type = *data_type*	data type of field.
	reply = "*list***"**	set of allowable entries.
	minimum = *n*	minimum value in field.
	maximum = *n*	maximum value in field.
	default = *value*	default value in field.
	prompt = "*text***"**	prompt to display for field.
	chars = '*x***'**	valid characters in field.
	display only	display only field.
:MAPS		identifies start of maps section.
map definitions		map definitions can be one or a combination of:
set $(*variable***)** *value*		force the variable to have the value.

if $(*variable***)** *condition value*
set $(*variable***)** *value*
.
else
set $(*variable***)** *value*
.
endif
		force the variable to have the value on the basis of if.. then.. else.. logic.
		
		(Note: Condition can be one of the logical operators: =, >, <, !=.)

map $(*variable***)** *condition v1 v2*		if a variable meets the condition in value 1, map the variable to that of value 2.
:COMMANDS *operating system commands*		starts the command section. Operating system commands can be included with:
	clear	clear screen prior to running command.
	pause	pause and prompt user after executing command.
	hold	after executing command, return user to the screen. This is useful if you call another screen in :COMMANDS, for example, ufill -*screen2*, which causes the current screen to be available for re-execution.
	reset	used with hold to reset fields to their default values.
	paste	used to put data in clipboard 9.
	write	used to write a text string to a file, ufill variable or shell variable.
	append	used to append a text string to a file, ufill variable or shell variable.
:HELP *help text*		help text to display with screen.
:C or *****		comments on the screen, not displayed with the screen.

Create a New Screen

To create a new screen, do the following:

1 Select the Create a Screen option.

Uniplex places you in the Word Processor.

2 Enter the screen description according to the following sections.

3 Press **Esc e** to save the screen.

Enter the Title of the Screen

Specify:

:TITLE
title name

where *title name* is the name of the screen. You can use up to 40 characters. Uniplex displays the title on line 4 of the screen, centered and highlighted.

The :TITLE command must be the first command of the screen description. For example:

:TITLE
-- FIND A DOCUMENT BY NAME --

Enter the Layout of the Screen

Specify:

:SCREEN
screen layout

where *screen layout* is a pictorial representation of the screen you want the user to complete. You can use boxes to enhance the appearance of the screen. The full word processing facilities are available to you. You can create as many screen pages as are required. Each screen should be no longer than 16 lines. Separate each screen with a new :SCREEN command.

You define the input fields in the screen. You use the square brackets, [and], to define the position and length of the input fields. Within the brackets, you enter the field name and pad any remaining space with the underscore character.

For example:

Name (or name pattern) of document	[document_____]
Folder in which to commence search	[directory_____]
Select level of detail required	[detail_____]
Select where you want the output	[output_____]

This example creates 4 input fields named document, directory, detail and output.

Enter the Field Definitions

Specify:

:LABELS
field definitions

where *field definition*s define the type and usage of each of the fields you specified in the :SCREEN section.

You can enter the following types of field definitions:

- **Specify the data type:**

  field_name = **type** *data_type*;

  where *field_name* is the field you used in :LABELS and *data_type* is one of the following:

  char allows input of any characters, *the default*.

  dec restricts input to numerics only.

  int restricts input to integers only.

  date restricts input to dates only.

  layer Selects an item, either from the local or system wide installation layer.

For example:

detail = type char;

You can use an optional clause with char type fields which will contain files or directories, to set the valid filetype. You specify:

file = *write*

where *write* is one of:

create	file must not exist
append	file must exist
read	must have read access to the file
write	file will be created if it does not exist
directory	a directory (must have read permission)
program	an executable file

If *data_type* is set to **char**, an options clause can also be included, if needed:

options = *display_attribute*

where *display_attribute* can be one of the following:

zap	If there is a default set for the field, when the user begins to type in another value, the default entry is erased from the screen.
tolow	The user input to this field will be displayed in lower-case characters automatically. This option cannot be used together with **toup** or **nodisp**.
toup	The user input to this field will be displayed in upper-case characters automatically. This option cannot be used together with **tolow** or **nodisp**.
nodisp	The user input is not displayed on the screen; this feature is useful when passwords are entered. This option cannot be used with **tolow** and **toup**.

autoexpand This option can only be used with character fields that also have a **reply=** list defined. It causes such a field to be auto-expanded whenever it becomes active (that is, it operates as if the user has pressed **Esc !** on entering the field).

If such a field is the only input field, that is, this is a one-field form, or it has other fields which are all defined as **display only**, then the effect of **Enter** or **Quit** on this field applies to the whole form. That is, **Enter** causes the :COMMANDS section to be run; **Quit** quits the form. Otherwise, **Enter** or **Quit** on this field will move to the next field on the form.

Restriction: You must not create a form which contains two or more fields which are all of this type. If you do so, the user will be unable to **Quit** or action (**Enter**) the form.

- **Specify the layer type:**

field_name = **type**=**layer, file**=*write*, **reply=**"*list*" [,**default**=*n*];

where *field_name* is the field used in :LABELS, *write* is one of the **file** options listed above, and *list* is the name of the file/directory to which access is requested, *for example "UAP/dbs/defaults*.

If the **file=directory** option is used, then write access must be available for that directory.

You also have the possibility of including an optional *default* parameter, where the valid entries are as follows:

1 = local

2 = system-wide

3 = network-wide (no longer supported)

A subset of the layer options above is made available in a scrollable field, depending on the user's access rights to the requested files in each of the given layers, if they exist.

o Specify the allowable entries (if it is a character field):

reply = "*list***"**

where *list* is a list of the allowable entries in a field separated by an exclamation mark (!). For example:

detail = type char, reply = "Name only!Full details";

At runtime the user will be able to scroll through the reply list by pressing the SPACEBAR.

You can specify a default value from the reply list using the syntax:

default = *n*

where *n* is the nth value in the reply list. For example:

detail = type char, reply = "Name only!Full details", default = 2;

This would set the default to the second value in the reply list, that is, "Full details".

It is also possible to set a default by using a shell command to pass the value to the field:

default = "**command***"**

For example:

folder = type char, default = "*pwd";

This will pass the name of the user's current working folder to the field 'folder' by default.

If you set a default by using the result of a shell command, the statement is only executed once, when the screen is loaded.

You can specify that a field will contain the results of an operating system command, using the syntax:

reply = "*command***"**

The output from the specified *command* will be passed into the field. For example:

detail = type char, reply = "*ls $HOME/messages";

Note: If the specified command generates several words or lines, a scrollable list of options is produced.

A shell command specified in **reply** is only executed once, when the form is loaded.

- **Specify the minimum and maximum allowable value:**

minimum = *n*

where *n* is the minimum entry if the field is decimal type, or minimum number of characters if the field is character type.

maximum = *n*

where *n* is the maximum entry if the field is decimal type.

For example:

directory = type int, minimum = 1;

width = *n*

where *n* is the maximum number of characters for a horizontally-scrolling, character type field.

Note: If you do not specify **width**, the maximum width is the same as the visible screen width defined in the :SCREEN section.

- **Specify the default value**

  default = *value*

  where *value* is a default value for the field on entry to the screen. Character fields need to be quoted, numeric ones do not. For example:

  directory = type char, default = "./", minimum = 1;

  An optional syntax for the default enables you to pass arguments from the invocation command line through as default field values:

  default = $(*n***)**

  where *n* is the nth argument on the command line. **$(1)** is the first argument following the screen name on the **ufill** command line, **$(2)** is the second, and so on.

  For example, if the name of the screen is called "find", it would be invoked with:

  ufill find

  If the name of the file to find is "invoice", the command line could read:

  ufill find invoice

  so "invoice" would be the first argument. You could pass this as a default value into one of the fields in the screen, for example:

  document = type char, default = $(1);

  where $(1) indicates the first argument from the command line, (in this example, invoice).

- **Specify a prompt:**

  prompt = "*string***"**

  where *string* is the prompt that will be displayed when the cursor is placed on the field. For example:

  detail = type char, reply = "Name only!Full details", default = 2, prompt = "Press SPACE bar for selections";

- **Specify the number of allowable decimal places (if it is a decimal field):**

  decimals = *n*

  where *n* is the number of decimal places you require. 0 specifies integer only; -1 specifies any number of decimal places. For example:

  decimals = 2

- **Specify the allowable character types (if it is a character field):**

  chars = "*x***"**

  where *x* can be one or more of the following:

Entry	Valid characters
A	A - Z, a - z, and any characters specified in either the UPPER or LOWER keyword of uniplex.sys
B	any Date separator character
C	any character less than space
D	0 - 9
E	any character not specified in the STOP list in uniplex.sys
F	! = > <
G	any valid time separator or suffix character: ; : . , - a p m A P M
H	any character > = space
L	a - z, and any characters specified in the LOWER keyword of uniplex.sys
M	. , ; : ! ? /
N	0-9, - or a Uniplex decimal point character
P	same as 'H'
Q	any character except a single quote
R	the tilde character (~)
S	space, tab, newline, carriage return and formfeed
T	same as 'A' but also includes the space character
U	A - Z, and any characters specified in the UPPER keyword of uniplex.sys
W	wildcard and pattern match characters: , * ? ^ [-]
X	safe UNIX filename characters (excluding /): a - z, A - Z, 0 - 9, and: _ . - #
Y	a - z
Z	A - Z

Note that several letters can be used to create wider ranges of accepatble characters, for example **"AD"** allows all alphanumeric characters.

Map User Input to Operating System Command Arguments

Specify:

:MAPS
map information

You use the :MAPS section to map user input to operating system command arguments.

You make entries in the :MAPS section using the syntax described below (this syntax is not Shell or C Shell):

o Assign a Value to a Variable:

set $(*variable***)** *value*

where *variable* is a field from the :SCREEN section or a command line variable, and *value* is the argument to pass to the command line. For example:

set $(DETAIL) "-print"

o Assign a Value to a Variable Using "If" Logic:

if $(*variable***)** *condition value1*
set $(*variable***)** *value2*
.
.
else
set $(*variable***)** *value3*
.
.
endif

where each *variable* is a field from the :SCREEN section or command line variables. The first value is user input, subsequent values are arguments to pass to the command line. The condition can be one of the logical operators: =, >, <. !=.

For example:

```
if $(detail) = "Name only"
set $(DETAIL) "-print"
else
set $(DETAIL) "-exec ls -al {} \;"
endif
```

- **Map One Value to Another:**

  map $(*variable***)** *condition old-value new-value*

  where the *variable* is a field from the :SCREEN section or a command line variable, *condition* is one of the logical operators, *old-value* is input from the user, *new-value* is an argument to pass to the command line.

  For example:

  ```
  map $(MODE) = "Yes" "-f"
  ```

- **Test One Variable Against Another:**

  test $(*variable1***)** *operator* **$(***variable2***) "***error message***"**

  where *variable1* and *variable2* are fields from the :SCREEN section or command line variables, *operator* is any logical operator except "less than" or "less than or equal to", *error message* is an alphanumeric string to be displayed on the screen if the result of the test is true.

  For example:

  ```
  test $(FIRST) > $(LAST) "First page to print is higher than last"
  ```

  This tests the value entered for the first and last pages to be printed of a file, and stops execution of the screen if the first page number is higher than the last.

Enter the Command Line

Specify:

:COMMANDS
command line(s)

where *command line(s)* are the operating system shell commands, to be executed. You enter each command line, as you would from an operating system prompt, using field names to indicate the points at which user input will be entered. For example:

find $(directory) -name '$(document)' $(DETAIL) $(OUTPUT)

You can also use the following keywords to control the execution of the :COMMANDS section:

Keyword	Description
clear	clears the screen prior to running the command.
pause	prompts with "Hit RETURN to continue" message after executing the operating system command.
hold	after executing the command, the user is returned to the screen.
reset	is used with hold to reset screen fields to the default values.
paste	is used to put data in clipboard 9.
write	is used to write a text string to a file, ufill variable or shell variable, using the following syntax: **write** {*filename*¦*ufill variable*¦*shell variable*} *text* where, either *filename*, *ufill variable* or *shell variable* is the target for the write, and *text* is the string you want to write.
append	is used to append a text string to a file, ufill variable or shell variable, using the following syntax: **append** {*filename*¦*ufill variable*¦*shell variable*} *text* where, either *filename*, *ufill variable* or *shell variable* is the target for the append, and *text* is the string you want to append.

Note: Regardless of where they occur in the :COMMANDS section, these keywords are processed or noted before any shell commands are executed. For example, the following will NOT do what you might expect:

```
:COMMANDS
cp /etc/group /tmp/file
APPEND /tmp/file "This line will be overridden by the CP above"
```

Enter Screen Help

Specify:

:HELP
help text

where *help text* is the text to be displayed if the user enters the help command.

Building a Screen

You build screens as follows:

1 Select the Build a Screen option.

2 Select the screen name.

Uniplex builds the screen. If the build is successful Uniplex reports the correct compilation on screen.

If the build is not successful, Uniplex reports the error lines on the screen, and copies the report to the Build Log.

The Build Log always contains the details of the last unsuccessful build operation. To examine the build log:

- Select the View Build Log option.

Uniplex displays the error report. You can correct errors by selecting the Edit Screen option from the menu. Then rebuild the screen.

3 Test the screen, by selecting the Run Screen option from the menu.

Uniplex displays your screen. Test the screen thoroughly before installing it on the system.

Configuring a Screen into the Menu System

You invoke a screen using the syntax:

ufill *screen_name* [*arg1*].. [*arg2*].. [*arg3*]..

arg1 can be referred to within the screen as **$(1)**, *arg2* as **$(2)**, and so on.

For example, if the source screen name was "find", the syntax would be:

ufill find

You can invoke the screen from within the Uniplex menu system by editing the Uniplex menu file, uniplex.menu.

For example, you might add a menu action such as:

1 ! 'ufill find <FILE>','Enter name of file',,C

Note: **ufill** invokes system-defined screens, whereas **ufill -l** will invoke a user-defined screen.

For details on the menu-action syntax, see the chapter Configuring Menus in the Uniplex Technical Guide.

Chapter 2

Formfill

THIS PAGE INTENTIONALLY LEFT BLANK

Accessing Formfill

To access Formfill:

o Select the Formfill option from the Uniplex Main Menu.

Uniplex displays the Formfill menu as follows:

```
                        F O R M F I L L

USE A FORMFILL                  CREATE A FORMFILL

1-Enter Data                    5-Create a Formfill
2-Print Data and Formfill       6-Edit a Formfill
3-Print Data only               7-Build a Screen and Print Forms
4-Print Form only               8-View Build Log

                                A-Formfill Alteration and Admin.

                                H-Help
                                Q-Quit
```

Creating a Formfill

Creating a formfill is a two stage process. First, you create a document containing the text you want to appear on the form, together with special field markers to indicate the fields on the form. Then you build the document. When you do this Uniplex creates two files. One file defines the appearance and operation of the form as it will appear on the screen. The other defines the format of the form as it will appear when you print it out. You can also optionally edit these two files to modify the appearance and operation of the form. See the section Modifying a Formfill to find out how to do this.

When creating a formfill, you follow these steps:

- Create a Formfill

  You enter a freehand design of the form containing the text you want to appear on the form, together with field markers to indicate the fields on the form. See the later section Create a Formfill.

- Edit a Formfill (optional)

  You can edit your freehand design to make changes, for example to enhance the readability of the form. See the later section Edit a Formfill.

- Build the Screen and Print Forms

  You build the formfill. When you do this Uniplex creates a screen form, defining the form you complete on the screen, and a print form defining the way the form will be printed out. See the later section Build the Screen and Print Forms.

Create a Formfill

You use the Uniplex Word Processor to create a freehand design of the form. The advantage of this is that you can use:

- Full word-processing editing commands.
- Boxes and effects to create a more attractive and readable form.
- Formatting commands to provide centering, justification and indentation in the text.

To create a formfill:

1 Select the Create a Formfill option.

Uniplex places you in an empty document.

2 Enter the text you want to appear in the form.

For each piece of variable information, such as a name, date or money, you enter blank spaces or underscores enclosed with the field markers, [ ].

Note: Each screen should be a maximum of 18 lines long.

For example:

```
                    Purchase Order Form

      Date: [          ]        Raised by [          ]

        Qty              Description              Price

  [               ] [                 ] [               ]
  [               ] [                 ] [               ]
  [               ] [                 ] [               ]
  [               ] [                 ] [               ]
  [               ] [                 ] [               ]
  [               ] [                 ] [               ]

    Signature:

    Dept. Head:
```

3 Press **Esc e** to save the formfill.

Uniplex prompts for a name for the formfill.

4 Enter a name for the formfill.

Edit a Formfill

You can edit a freehand design you have made, as follows:

1 Select the Edit a Formfill option.

Uniplex displays a list of the available formfills.

2 Select the formfill you require.

Uniplex displays the formfill.

3 Edit the formfill. The full word-processing commands are available.

4 Press **Esc e** to save the formfill.

Build the Screen and Print Forms

To turn the formfill design into a form, you do the following:

1 Select the Build a Screen and Print Forms option.

Uniplex displays a list of the available formfills.

2 Select the formfill you require.

Uniplex builds the formfill. Uniplex creates two files defining the appearance and operation of the form as it will appear on screen, and when printed out.

If there are any errors in the formfill, Uniplex reports them on the screen. Subsequently, you can use the View Build Log option to examine the error report. Otherwise, the form is ready for use. See the Formfill chapter of the Advanced Office System User Guide to find out how to use the form.

Note: If you subsequently decide to modify the appearance or operation of the form, you modify the formfill. See the section Modifying a Formfill for details of how to do this.

Modifying a Formfill

When you build a formfill, Uniplex creates a screen form and a print form. These define the appearance and operation of the form. The screen and print forms contain default formatting instructions. You can modify the forms to change these defaults. For example, you can change the title of the screen form, or change the formatting instructions on the print form. To modify a formfill, you use the Formfill Alteration and Admin menu option:

o Select the Formfill Alteration and Admin option.

Uniplex displays the Formfill Alteration and Admin menu:

```
ALTER SCREEN FORMFILLS   GENERAL

1 - Edit a Screen Formfill     D - Delete a Formfill
2 - Build a Screen Formfill    C - Copy a Formfill
3 - View Build Log             R - Rename a Formfill
                               L - List a Formfill

ALTER PRINT FORM         TESTING

4 - Edit a Print Formfill      E - Enter Data
                               P - Print Data and Formfill

                               H - Help
                               Q - Quit
```

Modify a Screen Formfill

You can modify formats and control the data entered into a screen formfill. To do this:

1 Select the Edit a Screen Formfill option.

Uniplex displays a list of the available formfills.

2 Select the formfill you require.

Uniplex displays the formfill.

You can modify the screen formfill by adding a title, and adding prompts and validations.

The screen formfill uses the same syntax and commands as the Screen Builder. See the previous chapter for details on how to:

- Add a title.
- Add prompts and validations.
- Enter field definitions.
- Specify allowable entries in a field.

Modify the screen formfill using the guidelines given in these sections.

3 Press **Esc e** to save the formfill.

4 Select the Build a Screen Formfill option.

5 Select the required formfill.

Uniplex builds the modified screen formfill.

Modify a Print Formfill

You can modify formats and control the data entered into a print formfill. To do this:

1 Select the Edit a Print Formfill option.

Uniplex displays a list of the available formfills.

2 Select the formfill you require.

Uniplex displays the formfill.

The print formfill contains all the fixed text, boxes and effects you drew freehand. It also contains labels for the fields. By default there are no special formatting controls associated with these labels. You can edit the formfill and provide these formatting controls as follows:

3 Move the cursor to the :LABELS section, and add the required formatting controls.

Press **Esc e** to save the formfill.

The remainder of this section describes the formatting controls you can use.

Include Format Information

The print formfill contains a :SCREEN definition, describing the form and labeling each of the field markers in the form. The :LABELS section names each of the labels used for the field markers. You can add formatting information to each of the labels to control the way the finished form is printed.

You must terminate each of the label definitions with a semi-colon.

The labels are defined using the following syntax:

labelname**,** option**,**...option**;**

The following *options* are available:

strip	Uniplex strips any trailing spaces after the variable, if it is not as long as the maximum field length.
left	Uniplex left justifies the variable within the field, if it is shorter than the field length.
right	Uniplex right justifies the variable within the field, if it is shorter than the field length.
center	Uniplex centers the variable within the field.
truncate	Uniplex truncates the variable if it is longer than the field length.
extend	Uniplex extends the length of the field if the variable is longer than the declared length.
float	Uniplex allows the position of the field to move if previous formatting on the line has changed the line length.
fixed	Uniplex places the variable in the field exactly as indicated by the formfill.
delete	Uniplex deletes the line, if there is no variable from the form.

Note: You cannot specify left and strip together.

For example:

labelname, right, float;

The default formatting options are:

- left
- extend
- fixed

In addition to these formatting options, you can include a *picture* of the format as follows:

format = *pattern*

This should be the last argument on the label line.

Note: The decimal separator (DECTAB) and the thousand and field separators (FIELDSEPS) can be configured in uniplex.sys. Depending on the configuration, the format shown in the following table may be different on your system. For details on uniplex.sys, see the Uniplex Technical Guide.

X Specifies an exact match of the characters. For example:

Format:	Input:	Output:
format = XX/XX/XX	031287	03/12/87

U Changes lower case to upper case. For example:

Format:	Input:	Output:
format = UXXX	fred	Fred

L Changes upper case to lower case. For example:

Format:	Input:	Output:
format = LXXXX	Fred	fred

A Matches alpha characters (a to z and A to Z). For example:

Format:	Input:	Output:
format = AA9	B7	B 7

9 Matches numeric characters. For example:

Format:	Input:	Output:
format = 999.99	23.4	023.40

Z Suppresses leading and trailing zeroes. For example:

Format:	Input:	Output:
format = ZZ9.9Z	23.4	23.4

*	Substitute leading zeroes with *. For example:		
	Format: format = ****.*	Input: 1.1	Output: ***1.1
,	Separate with a comma. For example:		
	Format: format = 9,999,999.99	Input: 123466977	Output: 1,234,669.77
/	Separate with a slash. For example:		
	Format: format = 9/999/999/99	Input: 123466977	Output: 1/234/669/77
0	Separate with a zero. For example:		
	Format: format = 99909	Input: 1234	Output: 12304
B	Separate with a blank space. For example:		
	Format: format = 99B09	Input: 1234	Output: 123 04
$	Format with a $ sign. For example:		
	Format: format = $ZZ,ZZZ,ZZ9.99	Input: 1000	Output: $1,000.00
-	Display minus sign if variable is a negative number. For example:		
	Format: format = -ZZ9.9	Input: -10.3	Output: -10.3
+	Display either + or -, depending on the variable. For example:		
	Format: format = +ZZ9.9	Input: 10.3 -10.3	Output: +10.3 -10.3
DB	Format for debit, displays DB if variable is a negative number. For example:		
	Format: format = ZZZZ9.99DB	Input: -100.3	Output: 100.3DB

CR Format for credit, displays CR if variable is a negative number. For example:

Format:	Input:	Output:
format = ZZZZ.99CR	-100.3	100.3CR

( ) Displays the variable within parentheses if it is a negative number. For example:

Format:	Input:	Output:
format = (ZZZ99.9)	-100.3	(100.3)

% Displays the percentage sign. For example:

Format:	Input:	Output:
format = Z9%	10	10%

Note: Uniplex truncates the variable if the variable is larger than the format you specify.

If you do not use any of the formatting options that indicate whether a number is positive or negative, there is no way of distinguishing a negative number. All negative numbers are shown as positive in this case.

Common Formats

This section provides examples of using the formatting options in some of the most common types of fields:

- **Text within Paragraphs. For example, names.**

  If you want a paragraph of text to include names, you do not want trailing spaces after the name; you want the text to appear normally spaced. In this case, you could define the label as:

  labelname, strip, extend, float;

- **Numbers within Paragraphs. For example, percentages.**

  If you want a paragraph of text to include percentages, you do not want trailing spaces after the percentages; you want the text to appear normally spaced. In this case, you could define the label as:

  labelname, strip, extend, float;

And, in the text include the percent sign. For example:

[ ]%

- **Text within Columns. For example, lists of names.**

  If you want to produce one or more columns containing text, you want the text within the columns left or right justified at the same point. In addition you do not want the columns to disturb each other's format. In this case, you could define the label as follows:

  labelname, left, truncate, fixed;

- **Numerics within Columns. For example, lists of salaries.**

  If you want to produce one or more columns containing numbers, you want the numbers within the columns left or right justified at the same point. In addition, you do not want the columns to disturb each other's format. Also, if the numbers are decimal, you want the decimal point positioned at the same point. In this case, you could define the label "labelname" as follows:

  labelname, right, fixed, format = ZZ9.9Z;

- **Money Values. For example, dollars and cents.**

  If you are including money values in your report, you want them displayed with the right money sign and the correct number of decimal places. In this case, you could define the label as follows:

  labelname right, fixed, format = $ZZ,ZZ9.99;

Note: For more details on the formatting options, see the Report Writer chapter in the Advanced Office System User Guide.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Customized Database Forms

THIS PAGE INTENTIONALLY LEFT BLANK

Accessing Customized Forms

You can create a customized form in one of two ways:

- Use the standard form as a basis for the customized form. See the section Creating a Customized Form using the Standard Form.
- Create the customized form without using the standard form. See the section Creating a Customized Form Without using a Standard Form.

To access the Customized Forms menu:

1 Select the Database Forms option from the Uniplex Main Menu.

2 From the Database Management menu, select the Build Customized Forms option.

Uniplex displays the Customized Forms menu:

```
                    C U S T O M I Z E D   F O R M S

BUILDING FORMS      GENERAL             UTILITIES

1 - Create Std Form 7 - Delete a Form   P - Printing
2 - Create New Form 8 - Copy a Form     T - List Tables
3 - Edit a Form     9 - Rename a Form   L - List Forms
4 - Build a Form                        V - View Table Schema
5 - Run a Form
                                        S - Select a Database
6 - View Build Log
                                        H - Help
                                        Q - Quit
```

The design of a new form is important. Before beginning to create a form, identify:

- The information you want to access
- Who is going to use this form, and how often
- Whether you are going to place any restrictions on access to the information

Give the user of the form as much assistance as possible by following these guidelines:

- Keep the layout of the form clear and uncluttered
- Do not have too many fields on a form
- Make the progress through the form easy
- Include prompts on how to use each field

Creating a Customized Form using the Standard Form

Follow these steps to create a customized form using the standard form:

1 Create the Standard Form. See the section Create a Standard Form.

2 Change the title of the form, if required. See the section Change the Title of the Form.

3 Change the screen layout of the form, if required. See the section Change the Screen Layout.

4 Set display only fields, the allowable entries in a field, and prompts to display with a field, if required. See the section Change Field Access and Use.

5 Specify access to this table via this form. See the section Specify Access in the General section.

6 Build the form. See the section Build a Form in the General section.

Create a Standard Form

Create a standard form as follows:

1 From the Uniplex Main Menu, select the Database Forms option.

2 Select the Build Customized Forms option.

3 Select the Select a Database option. Select the Database for which you want to create a form.

4 Select the Create Std Form option. Select the table for which you want to create the standard form.

Uniplex creates the standard form (see the next section for an example of a standard form) and prompts you to press RETURN to continue.

5 If you want to create another standard form, select the required table. If you do not want to create another form, press **Esc q** to leave this form.

6 Select the Edit a Form option to edit the standard form.

Uniplex displays the forms available.

7 Select the form you want to edit.

8 Edit the form description, according to the following sections.

You can use the full word processing facilities to edit the form. Note that you can change the ruler width if it is not wide enough for your needs. See the Word Processor chapter in the Uniplex II Plus User Guide for details.

Note: You can set the ruler width beyond the screen width, however, Uniplex displays a warning message when you compile the formfill.

9 Press **Esc e** when you have made all the changes you require.

Standard Form Template

This section shows the standard form, and explains each section of it.

:TITLE	The title section.
-- TABLE INQUIRY 'table1' --	The title Uniplex displays when a user accesses this form. See the section Change the Title of the Form.
:SCREEN	The screen section.
name :[L0_____] address1 :[L1_____] address2 :[L2_____] car_reg :[L3_____] salary :[L4_____]	The screen layout as seen by the user. The user does not see the label definitions 0 to 4, these are used in the :LABELS section.
:LABELS	The labels section.
table = table1	The name of the table this form accesses. Do not change this entry.
L0 = name; L1 = address1; L2 = address2; L3 = car_reg; L4 = salary;	Associates each label name with its corresponding column in the table. See the section Change Field Access and Use.
:END	The end section. Do not change the end section.

Change the Title of the Form

The standard form created by Uniplex contains the title for the form in the :TITLE section. As created, this contains:

-- TABLE INQUIRY 'table_name' --

Uniplex displays this title when a user accesses this form. You can change the title to a phrase or name that describes the function of this form. For example Update Product Prices.

To change the title:

1 Move to the line after :TITLE.

2 Overwrite the existing title with your new title.

For example:

```
:TITLE
Update Product Prices
```

Change the Screen Layout

The standard form created by Uniplex contains the screen layout for the form in the :SCREEN section. As created, this contains a name and field definition for each column in the table. Each field definition has a field of the correct length for the column, enclosed with square brackets, and contains a label name. The label names begin with the letter L and are numbered sequentially. Uniplex labels single character fields with a single letter (a - z).

If the form is made up of more than one screen, Uniplex creates a separate screen section for each screen.

For example:

```
:SCREEN
name                :[L0_________]
address1            :[L1_________________]
address2            :[L2_________________]
car_reg             :[L3_________________]
salary              :[L4_________]
```

You can change the screen layout as you require. For example:

- Change the column names to descriptions of the content of a field.
- Remove columns you do not want accessed with this form.
- Change the layout. For example, center the fields or add a surrounding box.
- Add any text you want displayed with the form.

For example, you could change the form above to the following:

```
:SCREEN
```

```
Use this form to enter the name, address and, if applicable,
car registration of new employees.

Enter last name:                [L0____________]
Enter address:                  [L1____________]
                                [L2____________]

Enter car registration:         [L3____________]
```

If you change a label name, you must change its corresponding entry in the :LABELS section (see below). Therefore, do not change the label names (L0, L1 and so on) unless absolutely necessary.

You can remove field definitions. If you do, remember to remove the corresponding entry in the :LABELS section (see below).

Change Field Access and Use

The standard form created by Uniplex contains the name of the table this form accesses and the label definitions for the form in the :LABELS section. As created, this contains each label name and its corresponding column name. For example:

```
:LABELS
table = table1
L0 = name;
L1 = address1;
L2 = address2;
L3 = car_reg;
L4 = salary;
```

If the field is numeric, Uniplex shows the number of decimal places as follows:

decimals *n* where *n* is the number of decimal places

decimals -1 any number of decimal places

For example:

L4 = salary, decimals 2;

The table entry identifies the table accessed by this form. Do not change this entry.

You can change the :LABELS section to:

- Specify whether a user can change the entries in a particular column, or whether the field is for display only. (You can further restrict access to tables in the :GLOBAL section. See the section Specify Access in the General section.)
- Specify the allowable entries that can be made in a field. For example, a range of values allowed.
- Specify that a character field can have certain display attributes.
- Specify prompts that Uniplex displays when a user accesses a field.
- Restrict the number of decimal places a user can enter into a field.

In addition, you must:

- Remove any label definitions for field definitions you have removed from the :SCREEN section.

To set a field to display only:

1 Move to the label definition for the field you want to set.

2 Add the following to the end of the label definition:

display only

For example:

L4 = salary, display only;

To specify on options clause for a character field:

1 Move to the label definition for the field you want to set.

2 Add the following to the end of the label definition:

options = *display_attribute*

where *display_attribute* is one of the those listed in the section **Enter the Field Definitions** of Chapter 1. For example:

L5 = size, options=autoexpand, reply="small¦medium¦large";

To specify the allowable entries in a field:

1 Move to the label definition for the field you want to set.

2 Specify the allowable entries in one of the following ways:

- **Specify a list of allowable entries (if it is a character field):**

  Enter:

  reply = "*list*"

  Where *list* is a list of valid entries, separated by an exclamation mark (!). For example:

  L1 = color, reply = "red!blue!green";

  To include a long reply list, start the list on a new line, after the = sign. For example:

  L1 = color, reply =
  "red!blue!green!yellow!orange!black!pink";

- **Specify the maximum and minimum allowable value (if it is a numeric field):**

  Enter:

  maximum = *n*
  minimum = *n*

Where *n* is a value. For example:

L2 = cost, minimum = 10;
L3 = amount, maximum = 20;

You can combine these commands to specify a range. For example:

L4 = price, minimum = 3, maximum = 10;

- **Specify a list of allowable entries, using a UNIX command (if it is a character field):**

  Enter:

  reply = **"****unix_command*"

  Where *unix_command* provides the entries for the field. For example:

  L1 = color, reply = "*cat /usr/UAP/colors";

  Where /usr/UAP/colors is a file containing the colors available.

  Note: A UNIX command specified in **reply** is only executed once, when the form is loaded.

- **Specify the number of allowable decimal places (if it is a decimal field):**

  Enter:

  decimals *n*

  Where *n* is the number of decimal places you want. Or -1 for any number of decimal places. For example:

  cost = cost, decimals 2;

- **Specify allowable character types (if it is a character field):**

  Enter:

  chars = **"***x*"

Where *x* can be one or more of the following:

Entry	Valid characters
A	A - Z, a - z, and any characters specified in either the UPPER or LOWER keyword of uniplex.sys
B	any Date separator character
C	any character less than space
D	0 - 9
E	any character not specified in the STOP list in uniplex.sys
F	! = > <
G	any valid time separator or suffix character: ; : . , - a p m A P M
H	any character > = space
L	a - z, and any characters specified in the LOWER keyword of uniplex.sys
M	. , ; : ! ? /
N	0-9, - or a Uniplex decimal point character
P	same as 'H'
Q	any character except a single quote
R	the tilde character (~)
S	space, tab, newline, carriage return and formfeed
T	same as 'A' but also includes the space character
U	A - Z, and any characters specified in the UPPER keyword of uniplex.sys
W	wildcard and pattern match characters: , * ? ^ [-]
X	safe UNIX filename characters (excluding /): a - z, A - Z, 0 - 9, and: _ . - #
Y	a - z
Z	A - Z

To include a prompt when a user accesses a field:

1 Move to the label definition for the field you want to set.

2 Add the following to the label definition:

prompt = "*prompt***"**

Where *prompt* is the prompt you want displayed when this field is accessed. For example:

L5 = id, prompt = "Enter the product ID";

Creating a Customized Form Without using a Standard Form

To create a customized form without using a standard form, follow these steps:

1 Design the form. See the section Customized Form Template.

2 Create the form. See the section Create a New Form.

3 Enter the title of the form. See the section Enter the Title of the Form.

4 Enter the layout of the form. See the section Enter the Layout of the Form.

5 Enter the Field Definitions. See the section Enter the Field Definitions.

6 Create prompts for a field, if required. See the section Display Prompts.

7 Join tables, if required. See the section Joining Tables.

8 Specify Access to the form, if required. See the section Specify Access in the General section.

9 Create help for the form, if required. See the section Add Help to a Form in the General section.

10 Add comments to the form, if required. See the section Add Comments to a Form in the General section.

11 Specify the end of the form. See the section Enter the End Section in the General section.

12 Build the form. See the section Build a Form in the General section.

See the next section, Customized Form Template, for how a form description is made up of sections. Then, follow the detailed descriptions of how to create these sections.

Customized Form Template

:TITLE *title name*	Identifies the title of the form
:SCREEN *Screen layout*	Start of screen layout
field definition **[***labelreference***]**	field display and definition
:LABELS	label definitions
table = *tablename*	reference to the database table
labelreference = *columnname*;	Associates the field definition with a table column. Optional extra criteria for accessing fields are:
display only,	display only field
options = *display_attribute,*	display attribute
reply ="*string*",	set of allowable entries
maximum = *n*,	maximum value of entry
minimum = *n*,	minimum value of entry
default = *value*,	default value in field
decimals = *n*,	no. of decimal places
prompt = "*text*",	prompt to display for field
:REFER	identify another table to access, for reference only
table = *tablename*	table name
search on *columnname operator labelname*	joins two tables by identifying the common column
:GLOBAL *restrictions*	the access restrictions to place on the form
:C or ***** *comment*	comments on the form, not displayed with the form
:HELP *help text*	help text to display with the form
:END	

Create a New Form

When you have planned your form on paper and you are familiar with the form description concepts detailed above, you can create the form on the system.

To create a form on the system, do the following:

1 Select the Build Customized Forms option from the Database Management menu.

2 Select the Select a Database option and then select the database for which you want to create a form.

3 Select the Create New Form option.

4 Enter the name for the form and press RETURN.

Uniplex places you in the Word Processor.

5 Enter the form description, according to the following sections.

The full word processing facilities are available to you. You can change the ruler if it is not wide enough. See the Word Processor chapter in the Uniplex II Plus User Guide for details.

6 Press **Esc e** when you have finished entering the form.

Enter the Title of the Form

Specify:

:TITLE
title name

where *title name* is the name of the form. You can use up to 30 characters. Uniplex displays the title on line 4 of the screen, centered and highlighted.

The :TITLE command must be the first command in a customized form.

Enter the Layout of the Form

The command you use to define the screen layout of the form is:

:SCREEN
screen layout

where *screen layout* is the form as it will be displayed to the user. The screen layout also contains field definitions, see next section. You use the Word Processor to type in the layout of the screen. All the normal effects are available to you apart from rulers or dot commands.

The screen layout you include here is the exact template Uniplex displays when users access this form.

You can create as many screen pages as are required. Each screen should be no longer than 16 lines. Separate each screen with a new :SCREEN command.

Enter the Field Definitions

Just as printed forms contain blank spaces to fill in, a database form contains blank spaces to fill in. The blank spaces on a database form are called fields.

When you are creating a form, you design how fields are displayed and specify what information they can contain.

You indicate to Uniplex that you are creating a field definition by using the following syntax within the screen layout:

Screen Information **[**labelreference spaces**]**

Note: You do not have to associate input fields with screen information. They can be anywhere on the screen.

The following table explains each part of this syntax:

Screen Information	You must identify the field to the user of the form. Normally you do this by including information on the screen that indicates the field purpose.
[ ]	The right and left square brackets indicate to Uniplex the beginning and end of a field definition.
labelreference	The "labelreference" is the name for this field. Later in the form description, you associate the "labelreference" with a column in a table. "labelreferences" can contain any alphanumeric characters, but must only be one word.

spaces	You define the length of a field by creating it with the required length. Thus the "labelreference" makes up part of the length. If you want the field to be any longer, include spaces, periods or underscores to make up the length.

For example:

Product Identifier: [productid_____]

This will appear on the screen as:

Product Identifier: [_____]

Label Definitions

You define the tables and columns you want the form to access in the :LABELS section of the form description. To indicate the beginning of a labels section, use the command:

:LABELS

Position the :LABELS section after the Screen Layout section. You must have a :LABELS section for each table you want to access.

The first entry in the :LABELS section must be:

table = *tablename*

where *tablename* is the name of the table you want to access.

In the next part of the :LABELS section, you associate the label references used in the field definitions of your form with columns in the table. Use the syntax:

labelname **=** columnname**;**

where:

labelname	is the name of the label reference in the field definition
columnname	is the name of the column in the table (the column name used here must be an exact match of the column name in the table)

For example:

productid = prodid;

You can also specify:

- The access the user can have to the table. See the section Specify Access in the General section.
- The default entries in the fields.
- The allowable entries in the fields. See the section Specify the Allowable Entries in a Field.
- A prompt to display when accessing the form. See the section Display Prompts.

Specify a Default Entry in a Field

You can specify a default entry in any of the fields in your customized form. When you run that form any default entries appear on the screen. You can change this entry if necessary. A default entry must follow the specified character types in each field.

To specify a default entry:

1 Move to the :LABELS definition for the field you want the default entry to appear in.

2 Enter the default entry you require using the format:

default="*default_entry***"**

where *default_entry* is the entry you want to appear in the field by default.

For example, if you have a field marked Department and specified lowercase characters you could type the following default entry:

default="administration"

You can also set the default from a UNIX command:

default="!*command***"**

To specify today's date in a date type field, enter:

default = $(TODAY)

This command enters the current date in the Uniplex date format setup on your system.

Note: You can also initialize a date field using **reply** (see below).

Specify a Display Attribute for a Character Field

You can specify an options clause for a character field. Specify:

options = *display_attribute*

where *display_attribute* is one of the those listed in the section **Enter the Field Definitions** of Chapter 1.

For example:

id = weather, options = autoexpand, reply = "cold!temperate!hot";

Specify the Allowable Entries in a Field

You can limit the entries the user is allowed to make within a field. This helps the user, since it reduces the chance of invalid entries. In addition, if you are allowing the user to update the database, it reduces the chances of invalid information being entered into the database.

You can limit the entries a user can make in the following ways:

- **By specifying a list of allowable entries (for character and date fields):**

  For a character field, specify:

  reply = "*string***"**

  where *string* is a list of all valid entries. Each of these entries is separated from the next by an exclamation mark (!).

  For example:

  id = productid, reply = "type1!type2!type3";

To include a long reply list, start the list on a new line, after the = sign. For example:

L1 = color, reply = "red!blue!green!yellow!orange!black!pink";

You can also make a list of valid entries from a UNIX command. For example, to include the list of file names in the current directory, enter:

L1 = filename, reply = "*ls";

The reply string can be up to 255 characters in length. The list of valid reply entries will be available to the user as a scrolling field.

You can use **reply** against a date field to initialize the field. For example, you can enter:

reply = $(TODAY)

To use a command to supply the date, the format must match that expected by Uniplex. For example:

reply = "*udate +%d/%m/%y"

- **If it is a numeric field, you can set the maximum and minimum allowable value**

  To specify the maximum value:

  maximum = *n*

  where *n* is the maximum value allowed.

  For example:

  cost = cost, maximum = 10;

  To specify the minimum value:

  minimum = *n*

  where *n* is the minimum value allowed.

  For example:

cost = cost, minimum = 3;

You can combine these commands to specify a range with which the entry must lie. For example:

cost = cost, minimum = 3, maximum = 10;

- **In a decimal number field, you can set the number of allowed decimal places**

  To specify the required number of decimal places:

  decimals *n*

  where *n* is the number of decimal places you want. Or -1 for any number of decimal places. For example:

  cost = cost, decimals 2;

- **Specifying Allowable Character Types**

  Specify:

  chars = "*x***"**

  x can be one or more characters used to specify the type of character-input allowed. For a list of valid entries, refer to the section Creating a Customized Form using the Standard Form.

Display Prompts

Specify:

prompt="*text***"**

where *text* is a message Uniplex displays when the user moves to this field. For example:

id = prodid, type char, prompt="Enter the product ID";

Joining Tables

The Uniplex database is a relational database. This means you can create direct relationships between tables when you access the database using forms. You use a technique called *joining* to connect data from different tables, as if they were part of the same table.

To join two tables, one column in each table must contain the same data. This column is called the join column and is used to create a temporary link between the tables.

1 Create the :LABELS section for the first table you want to access.

2 Create a :LABELS section for the subsequent table you want to access. If the subsequent table is to be *read only*, use :REFER instead, see below.

3 Include a search condition in the table entry in the :LABELS section for the second table:

table = *table2* **search on** *columnname operator labelname*;

Where *table2* is the name of the additional table you want to access, where *columnname* is the name of the column common to both the tables, *labelname* is the label reference in the form, and where *operator* is an operator to join the tables.

Operator can be one or a combination of the following:

=	equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
NOT	not
LIKE	like

Use these operators as follows:

= Use the equals operator to join two tables on their common column, accessing the entries as specified by the entries the user makes in the other fields of the form.

For example, if you have two tables in a database containing stock details and both contain a stock number field, you can join these tables with these fields using the equals operator.

> Use the greater than operator to join two tables on their common column, accessing the entries greater than the value the user specifies in the field on the form.

< Use the less than operator to join two tables on their common column, accessing the entries less than the value the user specifies on the form.

>= Use the greater than or equals to operator to join two tables on their common column, accessing entries greater than or equal to the value the user specifies in the field.

<= Use the less than or equals to operator together to join two tables on their common column, accessing entries less than or equal to the value the user specifies in the field.

NOT= Use the NOT operator with the = operator to join two tables on their common column, accessing numeric entries not including the entry the user makes in the field.

LIKE Use the LIKE operator to join two tables on their common field, accessing entries like the entry the user makes in the field. You can use the wildcard * to specify one or more characters, and the wildcard ? to specify a single character. If you do not use wildcards, Uniplex finds all the entries *containing* the entry the user makes in the field.

In addition, you can use the following keywords in the :LABELS section of the first table specified for a join:

validate Does not allow users of the form to add additional information to a record, if that record does not exist in the second table.

reference Allows users of the form to add additional information to a record, even if that record does not exist in the second table. This is the default.

General

This section contains information on:

- The optional sections you can add to your customized forms.
- How to build and run a customized form.

Specify Access

You can specify how much access the user can have to the database from the form.

You can specify restrictions that are made:

o Throughout the form

Specify:

:GLOBAL
restrictions

where *restrictions* can be one or more of the following

no edit means Uniplex will not allow the user to edit any entries accessed in the database

no delete means Uniplex will not allow the user to delete any entries it accesses from the database

no create means Uniplex will not allow the user to add any entries to the database

You can position the Global section anywhere in the form description. It is recommended that you position it after the form title.

o Throughout a table

There are two ways of restricting access to a table:

- Use the :REFER command instead of the :LABELS command when identifying a table you want to access. Uniplex will not allow the user to edit, delete or add entries to this table.
- Use the keywords: no edit, no delete or no add within the Labels section.

o On a field

You can restrict access to a particular field by specifying:

display only

in the label definition of a field. For example:

firstname = name, display only;

This means Uniplex will not allow the user to access this field, but will display entries from the database in it.

Add Comments to a Form

You can add comments to a form description that are ignored by Uniplex and are intended to document the form. Specify:

:C *comment*
or
***** *comment*

where *comment* is the comment you want to make. You can include comments anywhere in a form description. Position the :C or * in the first column.

Add Help to a Form

You can create help for a form. Specify:

:HELP
help text

where *help text* is any amount of text you enter that will help the user use the form.

You can display help when using a customized form as follows:

1 Press **Esc h**.

Uniplex displays the Help menu.

2 Use the menus to find the help for Customized Forms.

Enter the End Section

Optionally you can indicate the end of a form. Specify:

:END

to indicate the end of a form description.

Syntax Rules

Remember the following syntax rules when you create a form:

- You must include the :TITLE, :SCREEN and :LABELS sections.
- Each section header (for example :LABELS) must start in column 1 on a new line.
- If there is more than one screen page to a form, each new page must begin with a :SCREEN command.
- Each label definition in the :LABELS section must end with a semi-colon(;).

Build a Form

When you have created a form, you must build it before it can be used on the system. This could be known as compiling the form.

To compile the form, do the following:

1. Select the Build Customized Forms option from the Database menu.
2. Select the database that the form accesses, with the Select a Database option.
3. Select the Build a Form option.
4. Enter the name of the form and press RETURN.

   Uniplex builds the form. If there are any errors in the form, Uniplex displays them on your screen when it has finished building the form. In addition, Uniplex writes any errors to a Build Log. Use the View Build Log option to redisplay any errors.
5. If there are any errors, use the Edit Form option to edit the form and correct the errors. Then, compile the form again with the Build a Form option.

Edit a Form

If you create a form and subsequently want to edit it, use the Edit Form option as follows:

1 Select the Build Customized Forms option from the Database Management menu.

2 Select the database that the form uses using the Select a Database option.

3 Select the Edit a Form option.

4 Enter the name of the form and press RETURN.

Uniplex places you in the Word Processor.

5 Edit the form description.

6 Press **Esc e** when you have finished editing the form.

Use a Customized Form

When you have created and built a customized form, you can use it as follows:

1 Select the Build Customized Forms option from the Database Management menu.

2 Select the Select a Database option and then select the database you require.

3 Select the Run a Form option from the Customized Forms menu.

4 Select the form you want to run.

Uniplex displays the form. See the chapter Database Forms in the Uniplex II Plus User Guide, for details of how to use the database forms.

THIS PAGE INTENTIONALLY LEFT BLANK