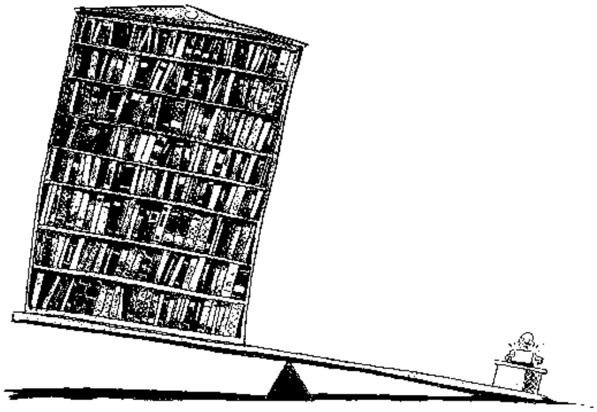
KE Texpress



Design Guide

KE Software Pty Ltd

Contents

Chapter 5 Query Form Design

Chapter 1

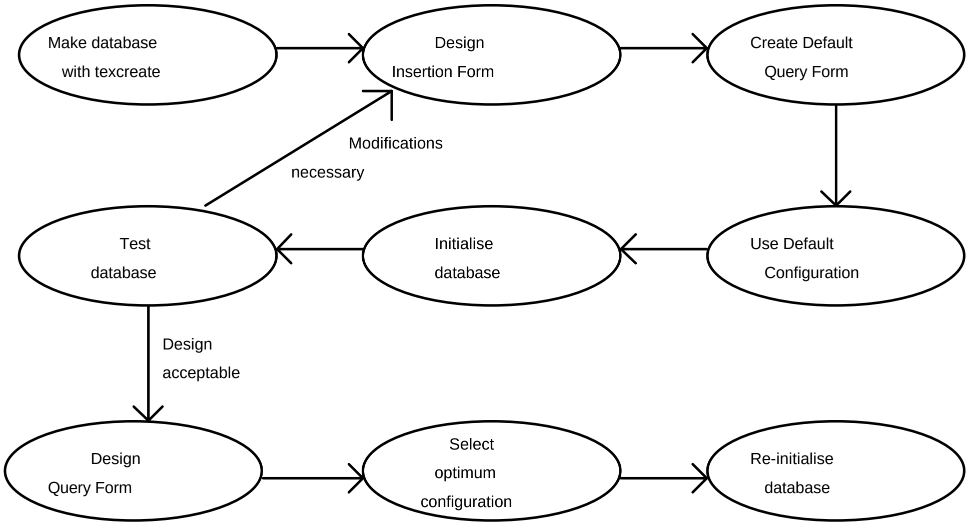
Introduction

Overview

This manual describes the KE Texpress database design facilities. The principal program is referred to as **Admin** and it provides access to all of the database design tools. Only the database administrator is permitted to use the design tools although other users can be granted access to certain programs invoked from the Administrator menu.

The major operations performed from are those of data definition, database configuration and initialization and the allocation of database access privileges. Numerous other operations are available but these are peripheral to the operation of the database and are provided only to present a more friendly working environment.

The following figure displays a flow chart outlining the major steps in the database design process. The important point to note is that reasonable testing of the database will be required before the design is finalized. During this design and testing process, the default Query form and database configuration should be used. Only when Insertion form design is complete, should detailed editing of the Query form be performed. Also at this stage an appropriate configuration (capacity of the database) can be specified.



Database design overview

Terminology

The KE Texpress system is a collection of programs each of which performs a specific task. For convenience, some of these programs have been given names which are referred to throughout this manual. The program names are:

Admin

This refers to Admin mode which is used for the design and maintenance of databases.

Texforms

This refers to the forms interface which is used for the collection and querying of data.

Texdesign

This program is used for the design of forms describing the type of data to be collected and queried.

KE Texpress uses a specific terminology to describe the various objects that are displayed and manipulated on the terminal screen. These terms and their definitions are as follows:

Database Administrator,DBA

The Database Administrator (hereafter, DBA) is the creator of a database. This user has total control over the database, deciding the format the database shall take and the types of queries that are permitted. The DBA also has the power to delegate database privileges to other users.

Form

A form is a screen image describing the type of data to be maintained and the actions to be performed on that data. It consists of a collection of headers and items located on the terminal screen in a format specified by theDBA.

Insertion form

The Insertion form is the form used by User for record data entry and display. If carefully designed, it contains sufficient, well chosen items to represent the data fully. TheDBA uses the forms editor (texdesign) to design the Insertion form.

Query form

The Query form is used to form the basis of a database query. It consists primarily of the subset of items of the Insertion form which have been indexed. Texdesign automatically generates the Query form from the Insertion form, after which theDBA can modify it to some extent.

Key form

A Key form is a form pre-defined in KE Texpress and specifically designed to permit database key and linked key access. Each Key form defines a particular Key type format which may be included in a database Insertion form.

Library form

A Library form is a predefined KE Texpress form that is made available to all DBAs for inclusion into Insertion forms. This provides a convenient facility for maintaining integrity and consistency across database designs.

Report form (also known asDisplay form)

A Report (or Display) form is a subset of the Insertion form which contains only the items from the Insertion form which are of interest for a particular label or report. It can contain additional reporting information and altered prompts and headers.

Record

A record is a collection of data which uniquely identifies one physical object. User accepts and maintains records for data querying and processing.

Header

A header is a string of printable characters which help describe the nature of the form. A header is not directly associated with any particular data in the record.

Item

Items are the basic form elements. They consist of an optional prompt followed by one or more fields.

Prompt

A prompt is a string of printable characters (similar to a header) which is associated with and identifies a field or collection of fields.

Field

A field is an area of the form where data is entered and displayed.

Terminal Capabilities

KE Texpress is a terminal intensive program and uses as many of the terminal's capabilities as possible. Thus some indication of those capabilities must be given to KE Texpress and this is done by setting the Unix environment variable **TERM**. The csh command to do this is:

```
setenv TERM  termtype
```

The sh equivalent is:

```
TERM=termtype
export TERM
```

Once *termtype* is defined in the KE Texpress terminal capabilities directory, this is sufficient for the system to access all of the relevant terminal characteristics.

Most terminal capabilities, other than cursor motion and a clear screen sequence, are not required and, if absent, have no other effect than to make screen manipulation slower and more cumbersome. One visually obvious capability, however, is highlighting. KE Texpress uses highlighting extensively to enable users to differentiate between user input and system prompts. Without this capability, no such visual distinction can be portrayed.

A terminal may support many forms of highlighting such as contrasting colours, half intensity, reverse video or even underlining. If possible, KE Texpress will make use of all of these attributes.

KE Texpress can be run on an enormous range of terminal types with all commands being accessible using KE Texpress's pull down menus. In order to bypass the pull down menus, keyboard accelerators are available for many KE Texpress commands. By default, KE Texpress is set up to use control characters for all of its keyboard equivalents, as control characters are supported by all terminal types. However, the system can be configured by the DBA or the system administrator at three levels (refer to the KE Texpress Interface Guide). The characters used to invoke any or all of the commands supported by KE Texpress can be altered to any set of characters including the function keys. All of text message used by KE Texpress (e.g. error messages, warnings, menu options, on-line help, etc.) can also be altered.

This document describes the default settings for commands and text message. When these settings are altered for a particular installation or database, the on-line help facility is automatically updated to reflect these changes.

Creating a Database

A program called **texcreate** is used to create a new KE Texpress database. It is invoked from the shell in the format:

```
texcreate dbname
```

where ***dbname*** is the name of the new database. The only restrictions on the database name are that it must not be more than eleven characters long and must be unique among databases on the current system. All programs and files needed for the new database are automatically created. The user issuing the command automatically becomes theDBA of the new database.

To avoid possible confusion, a database name which does not coincide with the name of a Unix command should be chosen.

Invocation

The KE TexpressAdmin program is invoked at the Unix command line with:

```
texadmin dbname
```

where *dbname* is the name of the database. (If invocation should fail then the shell environment variable, PATH, may need to be altered to include the directory in which the KE Texpress programs reside. See the system administrator for help or consult the Unix reference manuals.)

Chapter 2

General Utilities

Overview

This chapter describes a series of general features and functions which are used throughout KE Texpress. While these may be used for a variety of reasons, their operation is always the same.

Notation

Commands run within KE Texpress can be performed from pull-down menus, or via keyboard accelerators (characters). Often, a command can be performed in several ways. For uniformity and ease of use, only the pull-down menu option is described in the guide

The *Perform Query* command can be run by selecting the *Query* option on the *Function* pull-down menu in *Query* mode. The guide would describe it as:

```
    Perform Query ([Query] Function⇒ Query)
```

where the description of the command is followed by the pull-down menu option in brackets. The mode is contained in square brackets, e.g. [**Query**] and the pull-down menu name **Function** is followed by ⇒ and the command **Query** exactly as listed on the pull-down menu.

Alternatively, the Perform Query command can be performed from the keyboard, by holding down the **`Control`** key and typing **`Y`**. In the guide, the keystrokes are described as **`Ctrl+Y`**.

Many of the common commands may be invoked directly by a particular function key (provided the terminal supports function keys). Throughout KE Texpress (and the screen images displayed in this document), function keys are represented by the letter **F** followed by the function key number. Thus the command generated by pressing function key number one is represented as **F1**. The notation, F1 , is used in this document.

Other special symbols used in this document are:

`ESC`	The **escape** key.
↵	The **return** or **enter** key.
`DEL`	The **delete** or **interrupt** key (used to interrupt an operation).
`Space`	The **space bar**.
`Backspace`	The **backspace** key.
`Tab`	The **tab** key.
←	The **left arrow** key.
→	The **right arrow** key.
↑	The **up arrow** key.
↓	The **down arrow** key.

Front-End Menus

Front-end menus are usually displayed when the user first calls up the system. Some users, such as OPAC users, may be taken into a Query screen as soon as they login. Various front-end menus can be set up by the DBA, tailored to the privileges and functions of the users.

Menu options can be selected in one of two ways.

First, the user can move to the option by using the arrow keys ↓ and ↑, which will cause the option to be highlighted. Pressing ↵ will then select the highlighted option.

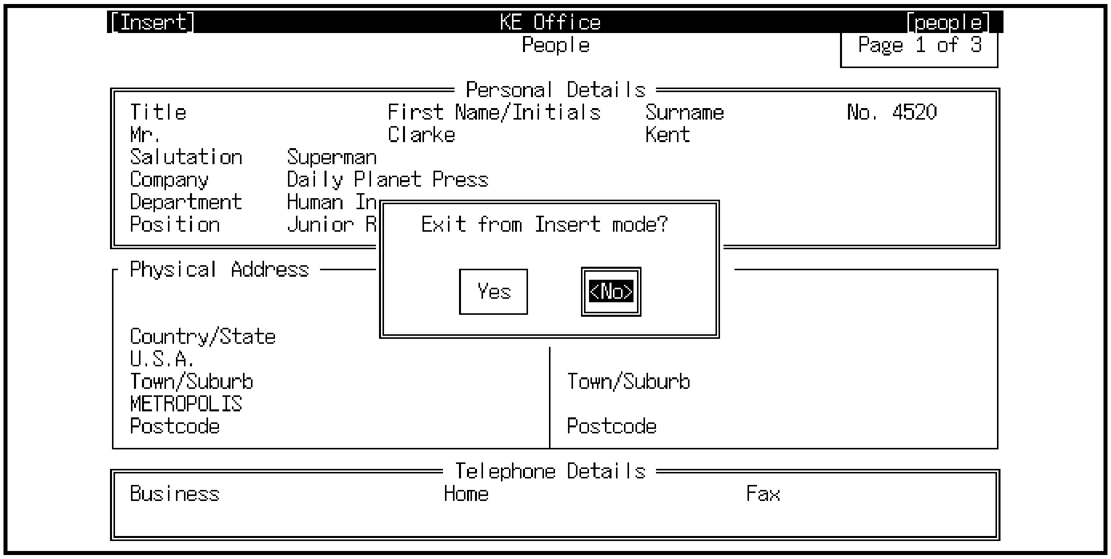
Second, the user can enter the letter or number associated with the menu. This immediately selects the optionwithout having to press↵.

To exit from the menu (always the last option on the menu), the user selects its associated letter or number, or alternatively, type **x**, **X**, **Ctrl+X** or F1 (if the terminal supports function keys.

If a menu option cannot be accessed or performed, the option will appear dimmed and it will not be possible to select it or move to it.

Screen Layout

The screen is divided into two regions: the **menu bar**, and below it the **form window** as shown below:



```
[Insert]                        KE Office                          [people]
                                  People                        Page 1 of 3
 ======================== Personal Details ============================
 Title                 First Name/Initials    Surname          No. 4520
 Mr.                   Clarke                 Kent
 Salutation    Superman
 Company       Daily Planet Press
 Department    Human In
 Position      Junior R   Exit from Insert mode?
 Physical Address
                            Yes      <No>
 Country/State
 U.S.A.
 Town/Suburb                          Town/Suburb
 METROPOLIS
 Postcode                             Postcode
 ======================== Telephone Details ===========================
 Business                 Home                     Fax
```

Screen layout

The menu bar occupies the top line of the screen. Usually it displays a mode indicator on the left, the command for pulling down menus or help, and the name of the application in the centre. When pull down menus are selected the menu bar displays the titles under which the available commands are grouped. During data entry the menu bar is also used to display helpful prompts.

The mode indicator displayed at the left edge of the menu bar indicates the current mode of operation, i.e. Insert, Query, Display and Edit. The database name is displayed in square brackets at the right edge of the menu bar.

The form window displays the various Insertion, Query and Report forms associated with the database. A database can have multiple query, and display or report forms.

A form may take up more space than is available on the computer screen, i.e. it may spill over onto several screens. At the top right hand corner of the screen is often a message, e.g. `"Page 1 of 3"` . This informs the user that the screen currently displayed is the first of 3 screens. Prompts, status messages and error messages are displayed in a pop up message box. This message box whose size varies according to the particular message, pops up in the centre of the screen, overlaying the form window. When the user has responded, the message box disappears.

Pull Down Menus

All KE Texpress commands can be invoked by selection from a pull down menu. These menus are normally not displayed and emerge from the menu bar only when requested. The menu bar displays the titles under which the available commands are grouped. When pull down menus are selected the group of commands under the first title is displayed and the first command on that menu is highlighted. Selection of a command involves first moving to its menu (using ← or →) and then moving to the option (using ↑ or ↓). Once the option is highlighted, the user calls up the command by pressing ↵. Note that keyboard accelerators cannot be used to involve commands while pull-down menus are exposed.

Some menus may display commands that cannot be selected, which will appear dimmed. This can occur when commands that are normally available are invalid under a particular set of circumstances. For example, the command to move to the next record when viewing the results of a query (**Move ⇒ Forward a record**) is invalid if only 1 record has been retrieved by the query.

The pull-down menu operations are described in the following paragraphs. Keyboard accelerators are displayed in parentheses, separated by a comma if there is more than 1.

The pull-down menu commands are:

Select pull-down menus (=)

Typing this key displays the first pull-down menu. The first available command on that menu is highlighted. If pull-down menus had previously been accessed in the current mode of operation then the menu containing the last selected command is displayed and that command is highlighted.

Select command (↵)

Typing the ↵ key removes the menu from the screen and performs the command currently highlighted.

Move down (↓, j)

Typing either of these keys moves the menu cursor down the currently displayed menu to the next command which may be selected.

Move up (↑, k)

Typing either of these keys moves the menu cursor up the current menu to the previous command which may be selected.

Move left (←, h)

Typing either of these keys moves the menu cursor to the left adjacent pull-down menu, displaying its commands

Move right (→, l)

Typing either of these keys moves the menu cursor to the right adjacent pull-down menu, displaying its commands.

Move to menu via title

It is possible to move directly to a pull-down menu by typing the first letter in the menu's title. Either the lower or upper case letter may be entered. If a particular letter refers to multiple titles then the pull-down menu associated with the first of these titles to the right of the current pull-down menu is displayed.

Help (?, F2)

Typing either of these keys displays a pop-up message box oulining the operation of the currently highlighted command. The last line of the message box displays a list of keyboard accelerators which can be used to invoke the command directly. Also displayed on the last line of the message box is the name of the KE Texpress help file which contains the currently displayed text.

Exit (Ctrl+X, esc)

Typing either of these keys clears the pull-down menus from the screen without performing any command on the menus.

Keyboard Accelerators

Most KE Texpress commands have keyboard equivalents. This means that the command can be invoked directly from the keyboard (rather than selection from a pull down menu) providing fast access to heavily used commands. Typically the keyboard accelerators are control characters, however the major commands may also be invoked using function keys (provided your terminal type supports functions keys and KE Texpress has an appropriate description for your terminal).

When a pull down menu is displayed, the keyboard equivalent (if any) is shown to the right of the command name. Note that whilst any pull down menu is displayed keyboard accelerators cannot be used. A command may have many keyboard accelerators, however pull down menus display only one. A list of all the keyboard accelerators for a particular command can be obtained by selecting help when the command is highlighted on the pull down menu.

On-Line Help

On-Line, context-sensitive help is always available in KE Texpress. It can be selected by typing**?** or **F2.**

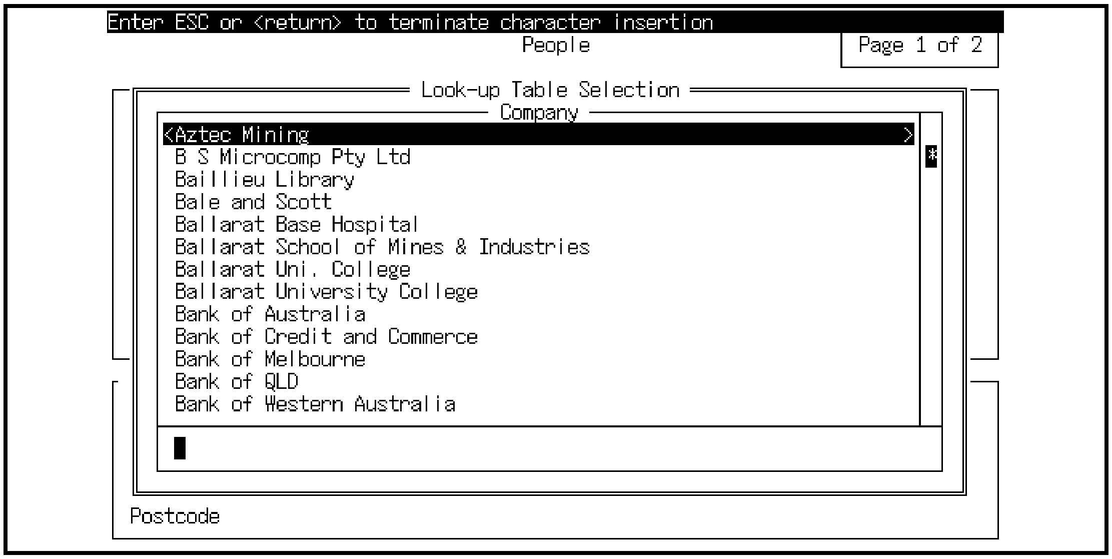
When help is selected, one or more screens of text describing the current mode of operation or the highlighted function on the pull-down menu, are displayed.

Typing ↵ causes the next screen to be displayed. Typing ↵ on the last help screen returns the user to KE Texpress.

To exit from help, type**F1** or **Ctrl+X.**

LookUp Table

A lookUp Table is a list of all entries previously entered into the system at that field. Choosing an entry from this table saves the user typing it into the field, which also eliminates typographical errors. When used during record insertions or edits, LookUp Tables also keep the format of entered information consistent across users. See also Chapter 3.



```
Enter ESC or <return> to terminate character insertion
                          People                          Page 1 of 2
               Look-up Table Selection
                       Company
<Aztec Mining                                                    >
 B S Microcomp Pty Ltd                                            *
 Baillieu Library
 Bale and Scott
 Ballarat Base Hospital
 Ballarat School of Mines & Industries
 Ballarat Uni. College
 Ballarat University College
 Bank of Australia
 Bank of Credit and Commerce
 Bank of Melbourne
 Bank of QLD
 Bank of Western Australia

Postcode
```

LookUp Table

Entering Information into Fields

As soon as a character is typed, any previous information is replaaced by the new entry and a line of underscores displays, outlining the width of the field. To finish data-entry, press↵ **ESC**

Move left (Edit⇒ Move left)

Move the cursor left one character position.

Move right (Edit⇒ Move right)

Move the cursor right one character position.

Move to start of field (Edit⇒ Move to first)

Moves the cursor to the first character in the current field

Move to End of field (Edit⇒ Move to last)

Moves the cursor to the lastcharacetr in the current field.

Insert (Edit⇒ Insert)

Insert characters before the cursor, the widith of the field permitting. Underscore will display, indicating the total width of the field.

Append (Edit⇒ Append)

Append characters after the cursor, the width of the field permitting. Underscores will display, indicating the total width of the field.

Delete (Edit⇒ Delete)

Delete the character currently under the cursor. All characters to the right of the cursor are drawn left by one character position.

Overstrike (Edit⇒ Overstrike)

Overstrike the character under the cursor with the next character typed

Copy text (Edit⇒ Copy text)

Copy part of the entire field into the Copy buffer. When this command is invoked, the menu bar changes.

The arrow keys can also be used to position the cursor. Note that it is not possible to move left past the position at which the Ctrl C command was invoked. When ESC is typed, all of the information between the position at which the command was invoked and the final marked position inclusive is copied into the Copy buffer without affecting the original information. This command is linked with the Extract text command.

Remove text (Edit⇒ Clear text)

Typing the ESC character with no preceding characters clears the field.

Undo (Edit⇒ Undo)

The undo command is used to remove the effect of the previous command, up to when ↵ was entered.

Extract text (Edit⇒ Extract text)

This command pastes, or extracts text from the copy buffer, if any, into the current field. The width of the field data type permitting and the field's

Rather, all field editing commands performed in succession are considered to be one alteration and hence are undone as one command.

The undo command, itself, can be undone, producing a toggle effect. At the completion of this command, the cursor is placed at the start of the Text buffer.

Other characters

All characters typed which are not valid result in the terminal bell being rung and are otherwise are ignored.

Terminate Field data-entry.↵ or ESC

The entered information is accepted as the field data, and the cursor jumps to the next field.

Backspace _

The backspace character to erases the previous character typed.

Erase all @

This character is used to erase all input typed into this field in this session.

Remove special effect \

The backslash character is used to take away the special effect (if any) from the next character entered. Hence the input sequence:

```
\@      produces       @
\\      produces       \
\X      produces       x       (where 'x' is any
character.
```

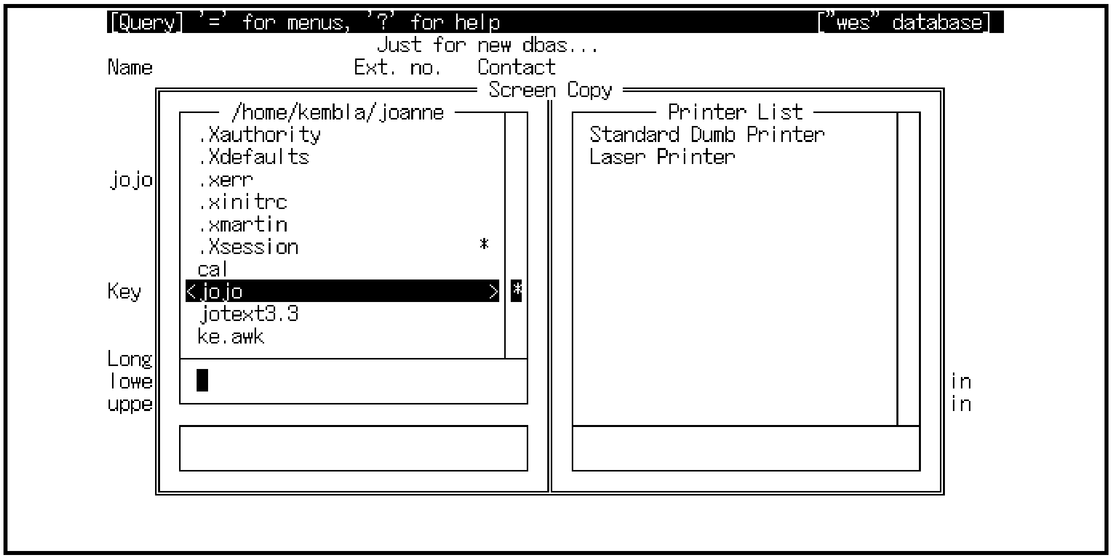
If a **?** is required in the first character position of a field, it must be preceded by a \ or help is given.

Printing or Copying Information

Users can:

- Copy all information on the screen (Screen Copy), invoked by Ctrl+G
- Copy the current record (Show ⇒ Copy this match). This is useful if the record spills over 1 screen.
- Copy all retrieved records (Show ⇒ Copy all match).

Information can be copied into a file or sent directly to a printer. When any of these copy facilities are invoked, selection boxes of file names and printers will display as shown in the following figure:



```
[Query] '=' for menus, '?' for help                      ["wes" database]
                     Just for new dbas...
Name                Ext. no.   Contact
                         Screen Copy
         /home/kembla/joanne              Printer List
        .Xauthority                    Standard Dumb Printer
        .Xdefaults                     Laser Printer
jojo    .xerr
        .xinitrc
        .xmartin
        .Xsession             *
        cal
Key    <jojo                  > *
        jotext3.3
        ke.awk
Long
lowe                                                          in
uppe                                                          in
```

The file selection box on the left consists of 3 sections. The top section displays all files in the current directory, including system files (starting with a. e.g. .cshrc). Please note that systems files should NEVER be selected for copies. Below the file section is a blank area where users can enter the leading character(s) of existing files, which will make the cursor jump to the first file starting with the character(s). The bottom section is another blank area, where users can enter the name of a new file.

The printer list box on the right displays all the valid printers. Users can scroll down the list by using the arrow keys ↓ or ↑, or by entering the leading character(s) of the printer in the blank section below the printer list.

Initially, the cursor will be inside the File box, in the New File Section. Here, users can enter a new file name, and when ↵ is entered, the file will be created with the information copied into it. The cursor can be moved to the Existing File Section, or the printer box by pressing the TAB key as often as necessary. Once the cursor is in the appropriate section, users can use the ↓ or ↑ key to move through the list, or enter the leading character(s) to make the cursor jump directly to the first item beginning with the character(s). Pressing Ctrl N or Ctrl P will scroll the list down or up.

To select a file or printer, move the cursor to it and press ↵. If any existing file is chosen, you will be prompted to append the information to the end of the file, overwrite the file with the information, or exit to about the copy.

Screen Copy

A copy of the current screen image can be obtained at any time by typing Ctrl G. The screen image can be saved or printed using the facility described in the previous section. By default the complete screen image is copied (including the menu bar).

If the database option, **screencopy=short**, is set the information in the menu bar is not copied.

If the database option, **screencopy=block**, is set the user is requested to mark two points on the screen. All information bounded by these two points is then copied. During the marking procedure the message window displays:

```
(Mark top left corner)
or:
(Mark bottom right corner)
```

as appropriate. The available commands are:

Mark corner (Function⇒ Mark)

Mark the current cursor position as a corner of the block.

Exit (Function⇒ Exit)

Exit from the marking procedure and abort the screen copy command.

Move left(Move⇒ Move left)

Move the cursor left one character.

Move right (Function⇒ Move right)

Move the cursor right one character position.

Move up (Function⇒ Move up)

Move the cursor up one line.

Move down (Function⇒Move down)

Move the cursor down one line.

Each of the cursor movement commands may be preceded by a repeat count, indicating that the command should be performed multiple times. Also, if the upper case equivalent of the command (eg. **H** for left) is entered the cursor moves to the next tab stop in the appropriate direction. This provides a faster method of movement around the screen. If an attempt is made to move the cursor outside the screen bounds the terminal bell sounds.

Screen Redraw

Screen redrawing can be performed at any time by typing **Ctrl+R**. This is useful if the screen happens to become jumbled, eg. if information is sent from another terminal to warn if impending database closure, or if the telephone line sends garbage over the modem.

Suspend or Invoke Shell

This is generated by the character, **Ctrl+Z**. On Unix systems with job control facilities this signal suspends execution and returns control to the Unix command interpreter or shell. Execution recommences. When the user enters **fg**.

For Unix systems without job control, this command causes a Unix shell to be executed. The shell to be executed is selected in the following manner:

(1) If the Unix environment variable, SHELL, is set then this program is used. For example, to execute the command interpreter, /bin/csh, the environment variable, SHELL, must be set to that value. This can be achieved from **csh** using the command:

```
setenv SHELL /bin/csh
```

or from **sh** using the commands:

```
SHELL=/bin/csh
export SHELL
```

(2) If the SHELL environment variable is not set then the user's login shell in the password file is executed.

(3) If the password file has no shell entry for the current user then **/bin/sh** is executed.

The termination of this shell results in a return to the exact context prior to the command being executed.

On systems with job control, the **Ctrl+Z** command takes effect immediately. On systems without job control, this command does not take effect immediately but rather is performed when the program is waiting for character input.

If usage of the suspend command is considered unsafe, it will be ignored

Interrupt

Typing the Delete or Interupt key, interrupts the current operation and returns the user directly to the current menu.

If usage of this interrupt facility is considered unsafe, the command will be ignored.

Text Perusal

Where large chunks of text have been split up into screen-sized records in KE Texpress, there is a text perusal facility, which allows the user to view the text consecutively, one screen at a time

The user needs to first retrieve one or more KE Texpress records, and then while within Display mode, invoke this perusal facility. **`([Display] Function ⇒ Peruse records)`**

The commands are as follows:

([Perusing] Move⇒ Forward a record) or Move⇒ Move from down)

The next page of text is displayed. Moving forward from the last page of text terminates the Text perusal procedure.

([Perusing] Move⇒ Backward a page) or Move⇒ Move form up)

The previous page of text is displayed.

([Perusing] Move⇒ Copy text) or Move⇒ Copy a record)

Make a local copy of the text for the user's own convenience. The information may be copied into a Unix file or printed.

([Perusing] Function⇒ Exit)

Terminate the display of text and return the user to the KE Texpress record.

Database Privileges

KE Texpress uses a network of privileges which allows a DBA to control access to data on several levels:

- Record
- Item
- Operation

To achieve this, KE Texpress uses the notion of privilege levels. The maximum number of privilege levels permitted is nine. However, the actual number of levels for a given database can be fewer and is determined at the time of database configuration (refer to Database Configuration in the Design Guide). For a database with **n** privilege levels, the levels are numbered **0** to **n - 1**. These numbers represent ranks in a hierarchical structure where level 0 is more privileged than level 1, which, in turn, is more privileged than level 2, and so on.

The DBA assigns each new database user an insertion and display privilege level which determines access to information whenever the database is accessed.

Record Privileges

At the time of insertion, each new record is assigned a privilege level. This implies that only users with a display privilege level higher than or equivalent to the record's privilege level can retrieve and view that record when querying. In other words, for insufficiently privileged users, this record is invisible.

Item Privileges

When the database is designed (refer to Database Design in the Design Guide), each object in both the Insertion and the Query Forms is assigned a privilege level. Only items, headers and boxes with a privilege level lower than or equivalent to the user's privilege level appear on the screen whenever a form is displayed. Thus selected data within a record can be protected while the user is still allowed access to some of the information in the record.

A user's insertion privilege level determines which objects can be accessed while inserting or editing a record. A user's display privilege level determines which objects can be viewed while displaying matching records. Therefore, a user may be privileged to view certain information but not sufficiently privileged to edit that information.

Operation Privileges

There exists for each user another set of privileges based on the operations that the user is permitted to perform. These privileges are not hierarchical in structure, but are simply granted or denied. The user is permitted to perform the operation only if the privilege is granted.

The user level operations controlled by these privileges include:

- Record insertion (I)
- Record edit (E)
- Record deletion (D)
- Change record privilege level (P)
- Query (Y)
- Destructive edit (U)
- Validation override (V)
- Videodisc access (X)
- Temporary record creation (T)
- Temporary file summary (F)
- Temporary record retrieval (R)
- Report access (C)
- Report create / delete (K)
- Unix facilities (N)
- Set user options (S)

A Look-up table insertion privilege also exists and can be granted or denied for each individual Look-up table in the database.

In addition to this set of privileges, there are administrator privileges which control access to the maintenance operations of the database.

Finally, there is the DBA privilege which cannot be granted to other users but is solely that of the DBA. All privileges, other than the DBA privilege, can be either granted or denied to all registered users of the database, and are controlled by the DBA.

Privileges in General

The existence of privilege levels is generally hidden from the user.

Privilege levels determine the operations and data available to the user. Pull-down menus reflect the users' privileges and hence users know only about operations which they can perform and data (fields or whole records) which they can view. Databases can be designed to have only one privilege level, which gives every user access to all the data.

Users with "Change record privilege" turned on their user account, can after a record's privilege level. To do this, the user must retrieve the record(s) and then in display mode enter the new privilege level **`([Display] Edit`** ⇒ **`Adjust Privilege)`**. This option is only available if the database has been configured to have more than one privilege level.

Chapter 3

The Administrator Mode

Overview

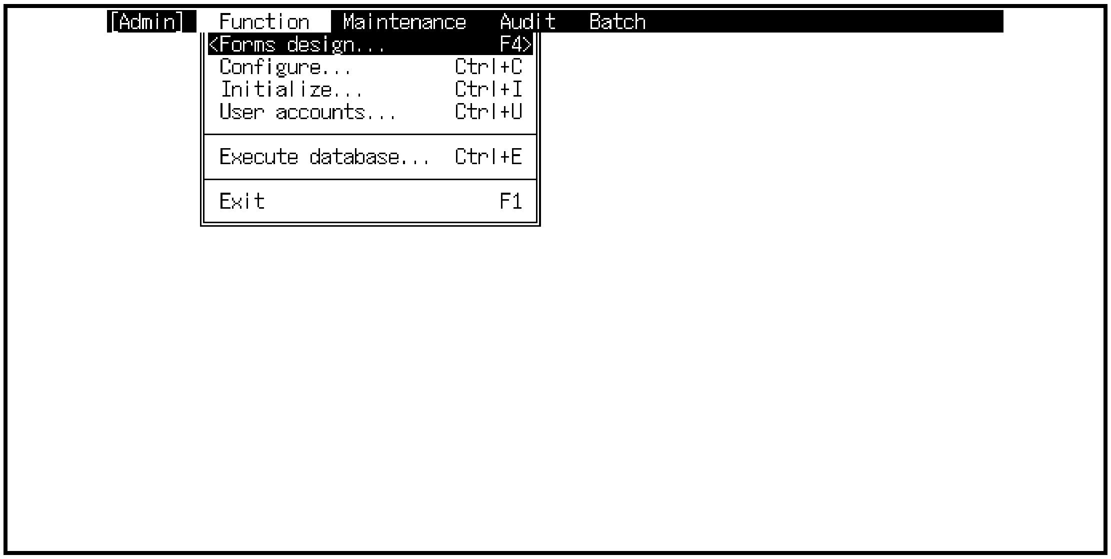
The person who creates a database (via the **texcreate** command) is the DBA and the only person permitted to use the database design facilities in **Admin** mode for that database. Maintenance operations are also performed in Admin mode.

The screen displays the Admin mode menu bar. Each of the titles relates to a pull down menu containing a group of administrator commands. Commands are selected via the pull down menus or by directly typing the keyboard equivalent for the command.

The commands under each menu bar title are briefly discussed in this chapter. For more detailed information refer to the following chapters and the KE Texpress Maintenance and Utilities Guides.

Function Menu

The **Function** pull down menu contains the primary commands required to build a KE Texpress database. Its structure is as displayed in the following figure.



Admin Function menu

Forms Design ([Admin] Function⇒ Forms design)

Each database is defined by two types of form; the Insertion and Query forms. The Insertion form is used for data capture as well as displaying retrieved records. Query forms are used to specify the information for a query. Both types of form are designed using the forms editor (texdesign). See Insertion Form Design (Chapter 4) and Query Form design Chapter 5).

Configure ([Admin] Function⇒ Configure)

Once the database forms are designed it is necessary to define the configuration of the database. The configuration defines the internal characteristics used by KE Texpress to efficiently implement indexing and space allocation. Refer to Database Configuration (Chapter 6) for further details

Initialise ([Admin] Function⇒ Initialise)

Database initialisation must be performed after forms design and database configuration. Initialisation creates the files needed by KE Texpress to handle the general usage of the database. Refer to the Initialisation section in Chapter 6 for further details.

User Accounts ([Admin] Function⇒ User Accounts)

This option allows the DBA to nominate the users who may access the database and to specify the operations these users are permitted to perform. Refer to the User Registration form section in Chapter 6 for a detailed discussion.

Execute Database ([Admin] Function⇒ Execute Database)

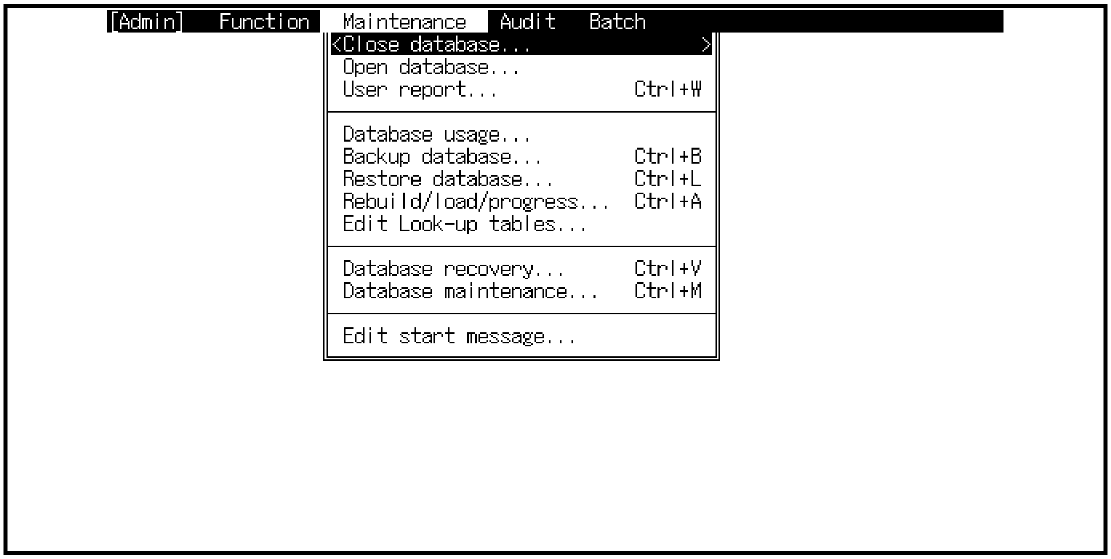
This option allows the DBA to invoke **texforms** from within Admin mode. When **texforms** is exited, the user returns to Admin mode. There is no difference in invoking **texforms** from Admin mode or directly from the shell. It is provided on the Function menu simply for convenience.

Exit ([Admin] Function⇒ Exit)

Selecting this option terminates the Administration menu and returns the user to the Unix shell.

Maintenance Menu

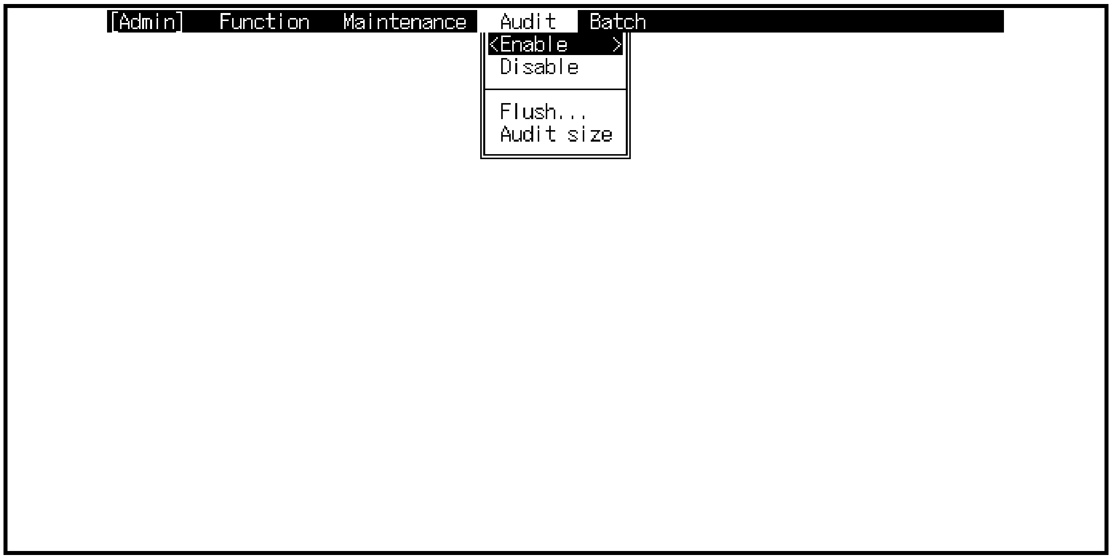
From this menu it is possible to execute numerous programs which assist with the general maintenance and day to day running of the database. Refer to the KE Texpress Maintenance Guide for more information. This menu is shown in the following figure:



Admin Mode Maintenance Pull-down Menu

Audit Menu

The Audit menu is used simply to enable or disable the audit trace for the database. This audit trace is used by the **texaudit** program to report on database activity. An option to flush or empty the audit file is also available on the Audit menu. Refer to the KE Texpress Utilities Guide for more information on auditing databases. The menu is shown in the following figure:

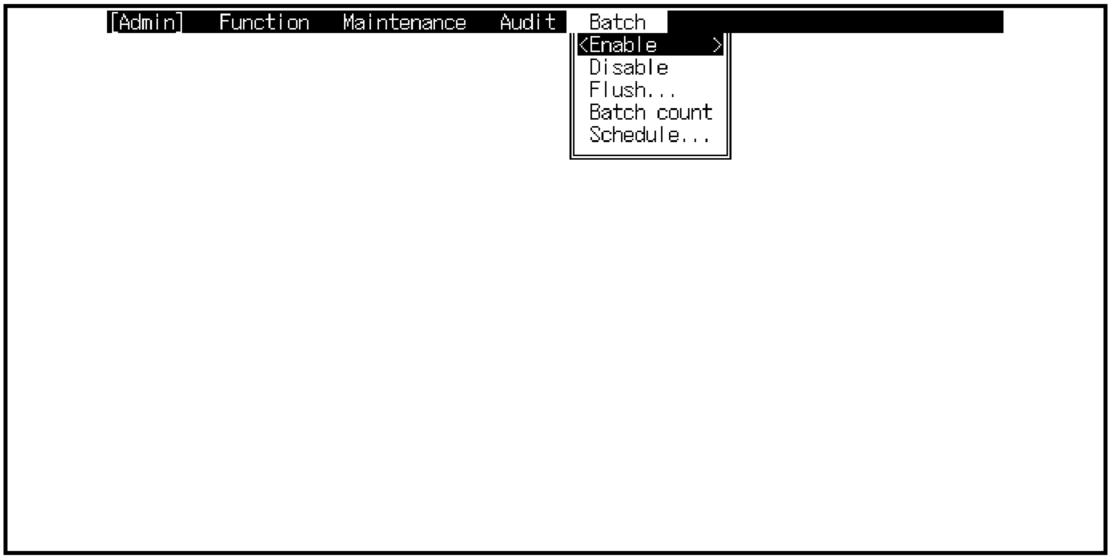


Admin Mode Audit Pull-down Menu

Batch Menu

This menu covers options for fast interactive insertions. If enabled, this mechanism delays most of the time consuming work of indexing until sometime later, when it can be completed in batch mode. This is recommended when high volumes of record insertions are carried out. Users can still access the inserted records extremely quickly.

Refer to the KE Texpress Maintenance Guide for more information. The menu is shown in following figure:



Admin Mode Batch Pull-down Menu

Chapter 4

Insertion Form Design

Overview

The forms editor **texdesign** provides an interactive form design environment. It produces form specifications that are used during the data insertion and query routines of **texforms**, the forms interface. A simple command structure is used to create and manipulate form components as well as specifying the attributes of these components. The facilities available in this program provide the DBA with total flexibility in the design of a database.

Texdesign is invoked from the Function menu in Admin mode. The following message is displayed:

```
Commencing forms editor ...
```

followed by:

```
Reading [Insertion form] . . .
```

The Insertion form is displayed and editing may commence. (To edit the Query form, the user can jump to it from the Insertion form via a Pull-down menu option. See the Misc section in this chapter)

When the Insertion form is first displayed, the cursor is positioned at the top left hand corner of the form window.

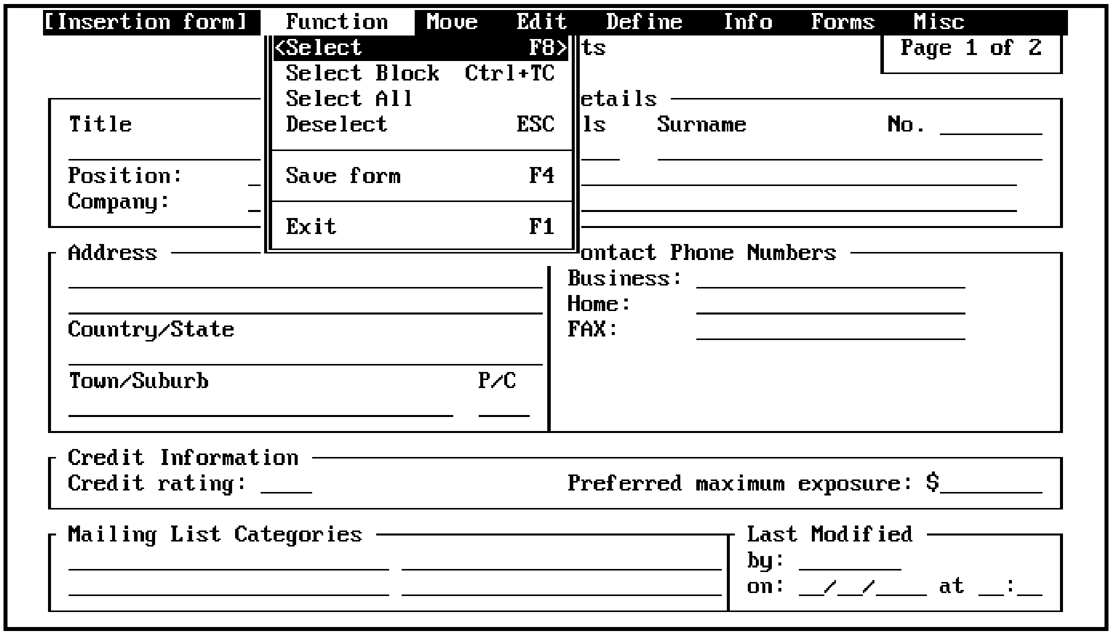
Form Layout

A form object is any prompt, field, item, header or box, or a group of items, headers or boxes, which are manipulated by the various commands.

Database forms vary in size. A form may be considerably larger than the screen size. In this case the screen is a window, which allows a portion of the form to be viewed at a time. The form window may be scrolled horizontally or vertically, to provide access to other parts of the form. If the cursor is not already at the form boundary, the form scrools automatically if an attempt is made to move the cursor outside the current window.

KE Texpress maintains internally a map of the layout of the form objects. When a command manipulates objects, the map is consulted to ensure that manipulation does not result in an overlap of objects. An exception to the prevention of clashing objects is that boxes may overlay, or be overlapped by any other object. Also, an object may overlap other objects during the course of a movement to a non-clashing location.

Function Menu



```
[Insertion form]  Function   Move   Edit   Define   Info   Forms   Misc
                 <Select           F8>ts                              Page 1 of 2
                  Select Block  Ctrl+TC
┌────────────────  Select All         etails ─────────────────────────────────┐
│ Title            Deselect       ESC ls    Surname           No. ________  │
│ ________________                   ___   _____________________________    │
│ Position:     _   Save form       F4 ____________________________________  │
│ Company:      _                      ____________________________________  │
└────────────────   Exit            F1                                        ┘
┌ Address ────────                    ontact Phone Numbers ─────────────────┐
│ _____________________________________ │ Business: _____________________   │
│ _____________________________________ │ Home:     _____________________   │
│ Country/State                         │ FAX:      _____________________   │
│ _____________________________________ │                                   │
│ Town/Suburb                       P/C │                                   │
│ ______________________________   ____ │                                   │
└───────────────────────────────────────┴───────────────────────────────────┘
┌ Credit Information ───────────────────────────────────────────────────────┐
│ Credit rating: ____                  Preferred maximum exposure: $________│
└───────────────────────────────────────────────────────────────────────────┘
┌ Mailing List Categories ──────────────────────────┬ Last Modified ────────┐
│ _________________________ _________________________│ by: ________          │
│ _________________________ _________________________│ on: __/__/____ at __:__│
```

[Insertion form] Function Menu

Selection and Deselection

The selection commands are used for adding objects to, and removing objects from, the current group of selected objects. This group of selected objects, termed "the current selection", can be manipulated by other forms editing commands. When the cursor is located over some non-blank portion of a header, box or item, it is deemed to be "marking" that object. The selection commands do not operate on parts of items, only on whole items. For individual manipulation of prompts and fields, refer to the appropriate forms editing command.

Select object ([Insert]Function⇒Select)

Select the marked object if it is currently not selected, and de-select it if it is currently selected. Selecting an object (item, header or box) adds it to the current group of selected objects, whereas de-selecting an object in the current selection removes it from the current selection.

Select block ([Insert]Function⇒Select block)

Select a block of objects by specifying an enclosing rectangle. The current cursor position is taken as the top left corner of the block and the user is requested to select the position of the lower right corner. All objects completely enclosed within the designated block are added to the current selection. Objects within the block which were already selected will remain so.

Select all objects ([Insert]Function⇨Select All)

Include all form objects as part of the current selection.

Deselect object ([Insert]Function⇨Deselect)

Discard the current selection.

Exit texdesign

Save form ([Insertion form]Function⇒Save)

The save command writes the form to file, saving the changes made during the form editing session. The form is automatically validated before it is saved, and while this is happening, the following message is displayed:

```
Validating [Insertion form]  .  .  .
```

If the Insertion form is invalid, a message reports this, and requests confirmation of the intention to save:

```
[Insertion form] is invalid. Really save? [y/n]
```

The No reply aborts the save and is merely for convenience, to ensure that an invalid form is not saved accidentally. However, when developing the design of a form, it is wise to save quite frequently and there is no harm in saving an invalid form at this stage.

After the validation, while the form is being written, the following message is displayed:

```
Writing [Insertion form]  .  .  .
```

In an uninitialised database, when the Insertion form is saved, the Query form may be invalidated. The following message is displayed:

```
Adjusting Query form   .  .  .
```

The adjustment reported is internal only. The appearance of the Query form does not change.

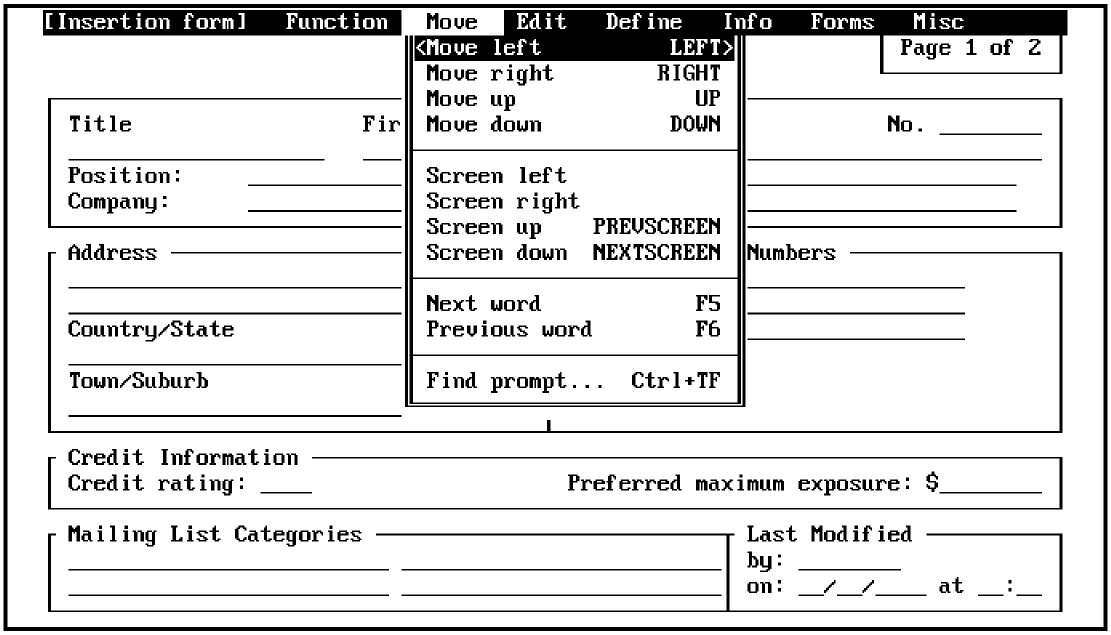
Exit form ([Insertion Form] Function⇒ Exit)

Exit from texforms. If the form has not been modified since the last save, texforms immediately terminates, and control is returned to Admin mode. If the form has been modified, the system first prompts for confirmation:

```
[Insertion form] [Modified]. Really exit? [y/n]
```

Choosing yes terminatestexforms. Choosing no aborts the exit command.

Move Pull-down Menu



[Insertion form] Move Menu

There are various methods of cursor movement available. In all of these methods the form window automatically scrolls when required. If an attempt is made to move the cursor outside the bounds of the form window, the terminal bell sounds.

The smallest step of cursor movement is one character position or one line. For larger steps, tab movement of the cursor can be used. By default, tab stops are set at each eight character positions of the form window width and each six lines of the form window length. These values may be changed if required. Tab movement is selected by entering the upper case equivalent of the appropriate command (eg. **H** for tab left, instead of **Ctrl+H**). A third method of cursor movement allows direct movement between words in the form. A word is defined as any sequence of non-blank characters.

Cursor Movement Commands

Move left ([Insertion form] Move⇒ Move left)

Move the cursor one character position to the left. Typing H moves the cursor left to the previous tab stop.

Move right ([Insertion form] Move⇒ Move right)

Move the cursor one character position to the right. Typing L moves the cursor right to the next tab stop.

Move up ([Insertion form] Move⇒ Move up)

Move the cursor one line up. Typing K moves the cursor up to the previous tab stop.

Move down ([Insertion form] Move⇒ Move down)

Move the cursor one line down. Typing J moves the cursor down to the next tab stop.

Screen Movement Commands

The following commands scroll the form window and are available if the form size is larger than the screen size. By default the horizontal scroll distance is set to 80 characters and the vertical scroll distance is set to 23 lines. These values may be changed if customisation is required.

Screen left ([Insertion form] Move⇒ Screen left)

Scroll the form window left. If the window is already at the extreme left of the form, the terminal bell sounds.

Screen right ([Insertion form] Move⇒ Screen right)

Scroll the form window right. If the window is already at the extreme right of the form, the terminal bell sounds.

Screen up ([Insertion form] Move⇒ Screen up)

Scroll the form window up. If the window is already at the top of the form, the terminal bell sounds.

Screen down ([Insertion form] Move⇒ Screen down)

Scroll the form window down. If the window is already at the bottom of the form, the terminal bell sounds.

Word Movement Commands

The following commands permit direct movement between words in the form. A word is defined as any sequence of non-blank characters. Boxes are ignored by these word movement commands.

Next word ([Insertion form] Move⇒ Next word)

Move forward to the next word. This operates on the form in a left to right, top to bottom manner with wrap-around.

Previous word ([Insertion form] Move⇒ Previous word)

Move backward to the previous word. This operates in the form in a right to left, bottom to top manner with wrap-around.

Note that when hot-key accelerators rather than the pull-down menu options are used for cursor movement, the commands may be preceded by a number indicating a repeat count. For example, **`l`** moves the cursor 1 character position to the right, but if **`7l`** is entered, the cursor moves seven character positions to the right. Similarly, **`3J`** moves the cursor down three tabs. If the repetition count for a command would position the cursor outside the form bounds the cursor is moved only the maximum allowable distance in the appropriate direction.

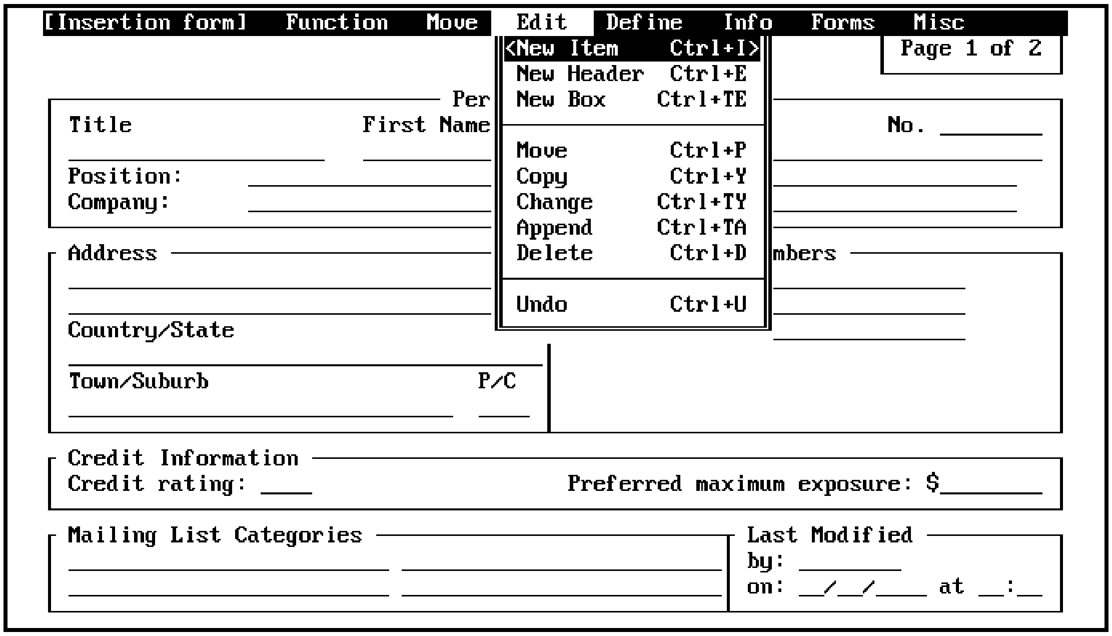
Jump to Prompt ([Insertion form] Move⇒ Find Prompt)

A selection box will pop up, requesting the user to enter the leading letter(s) of the prompt. When the user enters ↵, the cursor jumps to the next item prompt which begins with those letters. Searching for a prompt commences from the current item or at the item nearest the current cursor position. The form window scrolls if the prompt is located on a different page. If no matching prompt can be found, the terminal bell sounds.

If however the **finditem=ids** option is set, the same command will search for the next item whose internal id begins with those leading letters. (See the Database Options - General section in Chapter 7 of the User Guide).

Edit Menu

This menu contains the primary forms editing commands.



```
[Insertion form]  Function  Move   Edit   Define   Info   Forms   Misc
                                 <New Item    Ctrl+I>    Page 1 of 2
                                  New Header  Ctrl+E
                           Per    New Box    Ctrl+TE
 Title             First Name                              No. ________
 __________________ _________                       ___________________
 Position:    _______________     Move        Ctrl+P ___________________
 Company:     _______________     Copy        Ctrl+Y ___________________
                                  Change     Ctrl+TY ___________________
                                  Append     Ctrl+TA
 Address ________________________ Delete      Ctrl+D mbers
 ________________________________                    ______________
 ________________________________ Undo        Ctrl+U ______________
 Country/State                                       ______________
 ____________________________________
 Town/Suburb                   P/C
 _____________________________ ____

 Credit Information
 Credit rating: ____               Preferred maximum exposure: $________

 Mailing List Categories                         Last Modified
 ________________________ ________________________  by: ________
 ________________________ ________________________  on: __/__/____ at __:__
```

[Insertion form] Edit Menu

Inserting Items, Headers and Boxes

Before using any of these commands, the cursor must be positioned to define the location for the new object. Note that in general it is not possible for form objects to overlap (see "Form Layout"). However, item prompts and headers may be broken into several sections on the one form line. During the insertion of an item prompt or a header, input data is not permitted to overlap an existing object. Instead the cursor will automatically jump to the next available character position on the form line.

Insert Item ([Insertion form] Edit⇨ New Item)

The left edge of the menu bar displays the message:

```
Enter prompt
```

and character input is accepted, beginning at the current cursor position, to create the text of the prompt until either **ESC** or ↵ is entered. If the item is to have no prompt, **ESC** or ↵ is entered without typing any characters. If a null prompt has been defined, the current cursor position defines the start of the first field. If the item prompt is non-null, the right edge of the menu bar displays the message:

```
Select field position
```

The cursor should be moved to the starting position of the field and the position selected **([Insertion form] Function ⇒ Select position** If the item is set up without a prompt, the current cursor location is taken to be the field position. A selection box pops up with the message:

```
Enter field length:
```

and an integer field length up to 255 may be entered. The underscores representing the field are automatically displayed on the screen. If the user is unsure of the required length, s(he) can omit specifying the field length (by pressing ↵ without entering a number), and the cursor will be positioned at the start of the field, with the left edge of the menu bar displaying a message:

```
Enter field underscores
```

The user can then press the **SPACE** bar, which will cause underscores representing the length of the field to display on the screen, until ↵ is entered. A message box will pop up prompting :

```
More fields? [y/n]
```

Choosing yes causes the field entering procedure to repeat. Note that there is no limit to the number of fields set up for an item.

Insert header ([Insertion form] Edit⇒ New header)

The menu bar displays the message:

```
Enter header
```

Any character typed by the user will be accepted as header text and displayed on the form. The header string should be terminated by **ESC** or ↵. If no characters are entered, the header is not inserted into the form.

Insert box ([Insertion form] Edit⇒ New Box)

The current cursor location is taken to be the top left corner of the box. The menu bar will display a message requesting the user to select the position of the bottom right corner of the new box. Next, the user is asked :

```
Single or double line box? [s/d]
```

Choosing **s** selects a single line box, while **d** selects a double line box and the new box is then drawn on the form.

Some terminals may not be capable of displaying double lines. In this case the best approximation is displayed (typically single lines). If zero height or width is specified, boxes will display as simple horizontal or vertical lines .

Move Objects ([Insertion form] Edit⇒ Move)

The move command is used to relocate objects within the form. It operates on the currently selected object(s). If nothing is currently selected, the object which the cursor is currently marking may be moved. If that object is a prompt or a field, then just that prompt or field will be moved. To move a complete item, the complete item must first be selected.

The object(s) can be dragged to a new form position using the cursor movement commands. The final position is marked by entering **`Ctrl+C or F8`** During a move operation, it is possible to move objects across other objects on the way to a new location within the form. However, objects are not allowed to clash when the final position is specified. If, when **`Ctrl+C`** **or F8** is pressed, a clash is detected, a message will display :

```
Clash at present position
```

The object(s) must be relocated to a non-clashing location before pressing **`Ctrl+C`** **or F8** again.

If there is no room within the form for moved object(s), replace them in their original position and then change the form boundaries. (See the Form Bounds section in this chapter). If an attempt is made to move any part of the object(s) outside the form boundaries, the terminal bell sounds.

Copy Objects ([Insertion Form] Edit⇒ Copy)

The copy command is used to duplicate objects within the form. Copying is most useful when several complex items of a similar nature are needed on the same form. Only one item need be created explicitly and the others can simply be copied from the original item. This command operates on the current selection.

If nothing is currently selected the object which the cursor is currently marking is copied. If that object is a field then just that fieldwill be copied. As items may have only one prompt it is not possible to copy a prompt. To copy a complete item, the complete item must first be selected.

The copy command is similar to the move command.The original object(s) remain stationary and duplicate object(s) emerge in the position(s) of the original object(s) and can then be relocated. As with the move command, if a clash is detected when the end position is marked, a message will display :

```
Clash at present position
```

The object(s) must be relocated.

It is possible to take a copy of some object(s) and then find that there is not room for the duplicate in the form. If this situation arises the copy command can be aborted by typing the **DEL** key.

Copying preserves the attributes of the object, except in two situations. Copying items which constitute a partial or complete multi-level hierarchy does not result in a new hierarchy being formed for the new copy. All multi-level hierarchies must be created explicitly (see the Hierarchies section in this chapter). Also if a Key item is copied, the new item will not be defined as a key item. This is because a form may have only one Key item.

Change Objects ([Insertion form] Edit⇒ Change)

The change command is used to alter an existing form object. Each of the objects in the current selection may be changed in turn.

If nothing is currently selected, the object which the cursor is currently marking may be changed. If that object is a prompt or a field, then just that prompt or field will be changed. If a complete item is to be changed, the complete item must first be selected.

Note that integer field types have a length limit of 15 characters and real field types have a length limit of 25 characters.

When the command is chosen, the object is cleared from the form and a new position may be selected (**`Ctrl+C or F8`**). If the same form location is required the location selection command **(`Ctrl+C` or F8)** may be entered immediately. The procedure then followed is similar to creating a new object of the same type.

It is not permissible to delete an object by changing it to a null object. If this is attempted, an error message is displayed, the original object is restored and the command is terminated. To remove an object from the form, use the delete command. (See the Delete Object section in this chapter).

Append to Item ([Insertion form] Edit⇒ Append)

The append command is used to add a prompt and/or fields to existing items. Each of the items in the current selection may be appended to in turn.

If nothing is currently selected and the cursor is currently marking either the prompt or a field of an item then the append operation will apply to that item.

For items which do not have a prompt, a prompt may be appended. Following this, new fields may be appended in turn to the item. The standard procedure for inserting a new item is followed.

Delete Object ([Insertion form] Edit⇒ Delete)

The delete command is used to remove objects from the form. It operates on the current selection. If all objects are currently selected, confirmation is requested before the delete operation proceeds.

If nothing is currently selected, the object which the cursor is currently marking is deleted. If that object is a prompt or a field then just that prompt or field is deleted. To delete a complete item, the complete item must first be selected.

There are a few restrictions on the type of items that may be deleted. It is not possible to delete an item in a multi-level hierarchy. The hierarchical relationship must first be destroyed.

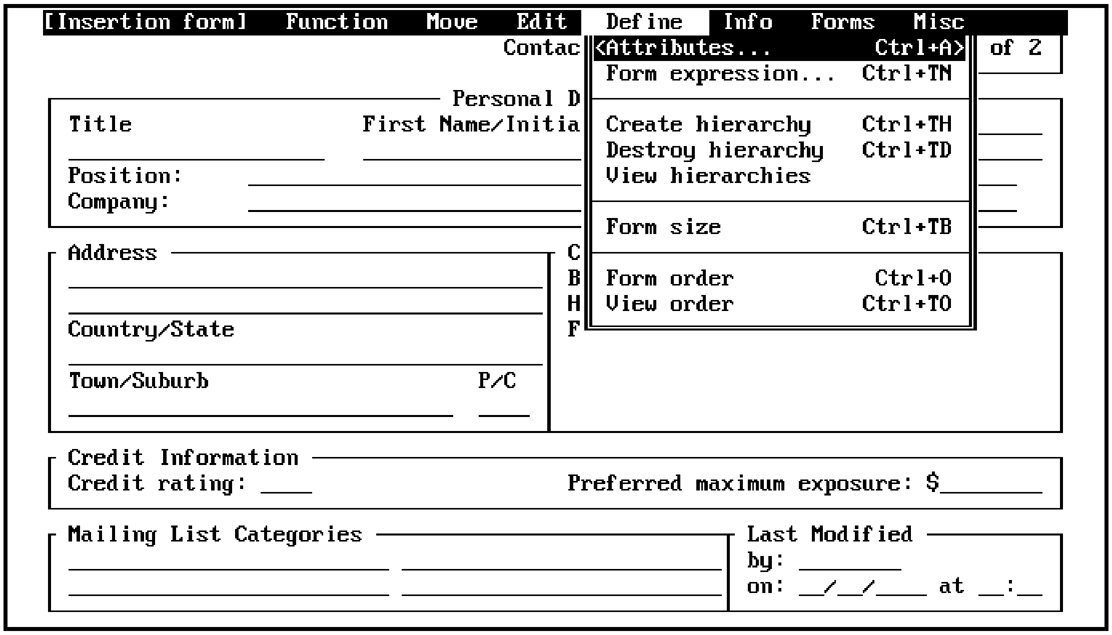
If the database has been initialized, or the data is saved for automatic re-entry, the delete command alters slightly. If editing is restricted, delete operations which could affect the data are disabled. Otherwise each non-trivial delete operation requests confirmation before proceeding.

Undo a Change ([Insertion form] Edit⇒ Undo)

This command can be used to undo the last command executed. Only commands which act on form objects may be undone. This includes all the commands on the Edit menu.

No message is generated; the form is just silently returned to the state prior to the execution of the previous command. For example, if the previous command had moved an item to a new position within the form, then the undo command would place the item back at its original location. The undo command itself may be undone, producing a toggle effect.

The Insertion Form Define Menu



```
[Insertion form]   Function   Move   Edit    Define   Info   Forms   Misc
                             Contac <Attributes...         Ctrl+A>   of 2
                                     Form expression...   Ctrl+TN
                   Personal D
 Title          First Name/Initia    Create hierarchy     Ctrl+TH  _____
 ____________   ________________     Destroy hierarchy    Ctrl+TD  _____
 Position:      ________________     View hierarchies              ___
 Company:       ________________                                   ___
                                     Form size            Ctrl+TB
 Address                         C
 _____________________________   B   Form order            Ctrl+O
 _____________________________   H   View order           Ctrl+TO
 Country/State                   F
 _____________________________
 Town/Suburb               P/C
 ______________________    ___

 Credit Information
 Credit rating: ____             Preferred maximum exposure: $_______

 Mailing List Categories                          Last Modified
                                                  by: _______
 ___________________  ___________________         on: __/__/____ at __:__
 ___________________  ___________________
```

[Insertion form] Define Menu

Attributes

Items, fields, headers and boxes have numerous attributes which determine how they are treated by the KE Texpress. For example, attributes can be set to indicate:

- An item is to be indexed, enabling fast searching of its values.
- Only certain data types are to be accepted by a field.
- A field value is to be extracted from another database.
- A field value is to be calculated from fields within the current database.

Attributes should be chosen to maintain data quality and database efficiency. In order to define attributes correctly, the DBA should be familiar with the type of data expected, as well as the database query requirements.

Attributes fall into two categories, boolean attributes and text attributes. Boolean attributes are either true or false (ie. set or reset). Text attributes take textual values. Certain sets of attributes are mutually exclusive and some mutually inclusive, ie: they cannot/must be set at the same time, and these consistencies are automatically maintained by the system.

Define Attributes ([Insertion form] Define⇒ Attributes)

This command calls up attributes screens for editing the attributes of the object which the cursor is marking. When attribute editing commences, the cursor is placed at the first attribute which may be edited. The attribute path may then be traversed and editing performed as desired. Due to the attribute mutual exclusion/inclusion properties, the traversal path of the cursor among the attributes may change as attributes are altered.

The item attributes screen consists of several pages. The first page contains the general attributes for the particular item, and subsequent pages contain the attributes of each individual field within the item.

The following commands are used to define the attributes:

Set/ResetAttributes ([Item/Field*n* attributes] Function⇒ Toggle)

Toggle the current attribute's value. To toggle an attribute means to set/reset its current value. Therefore if the current attribute is set (turned on), it is reset (turned off), and vice-versa. The toggle command is used solely for boolean attributes. For easy identification, all attributes which may be toggled have a prompt ending with a question mark (?). A set boolean attribute is indicated by the **#** character.

Edit Field Expression ([Field*n* attributes] Function⇒ Edit expression)

This command is available only while editing field attributes and while the cursor is located at one of the assignment, validation or branch expression attributes. It allows an expression to be edited using a Unix text editor. Exiting from the editor returns control to attribute editing.

The editor invoked by this command is determined by the Unix shell environment variable, EDITOR, is used to invoke an editor. If EDITOR is not set, an editor is chosen from the editor settings in the KE Texpress text file for the current database (typically this will be *vi*).

Edit Also Search Items ([Item attributes] Function⇒ Edit item list)

This command is available only when editing item attributes, and while the cursor is at the Also search attribute. It is used to compile a list of items which are also to be searched whenever the current item is used in a query. When the command is run, a selection box pops up containing the Ids of all items in the form. The Id of the current item and Ids already in the Also search list for the current item are not available for selection. Selection is made by moving through the list to locate each required Id, selecting it with **`Ctrl+C or F8`**, and when all Ids required have been selected, typing ↵. All selected Ids are then added to the Also search list.

Copy Field Expression ([Field*n* attributes] Function⇒ Copy expression)

The user is prompted to enter the name of a Unix file from which the expression is read. Any existing expression is clobbered.

Copy Field Attributes ([Field*n* attributes] Function⇒ Copy field)

This command is available only while editing field attributes. It allows the attributes of another field within the current item to be copied to the current field. A selection box will pop up prompting:

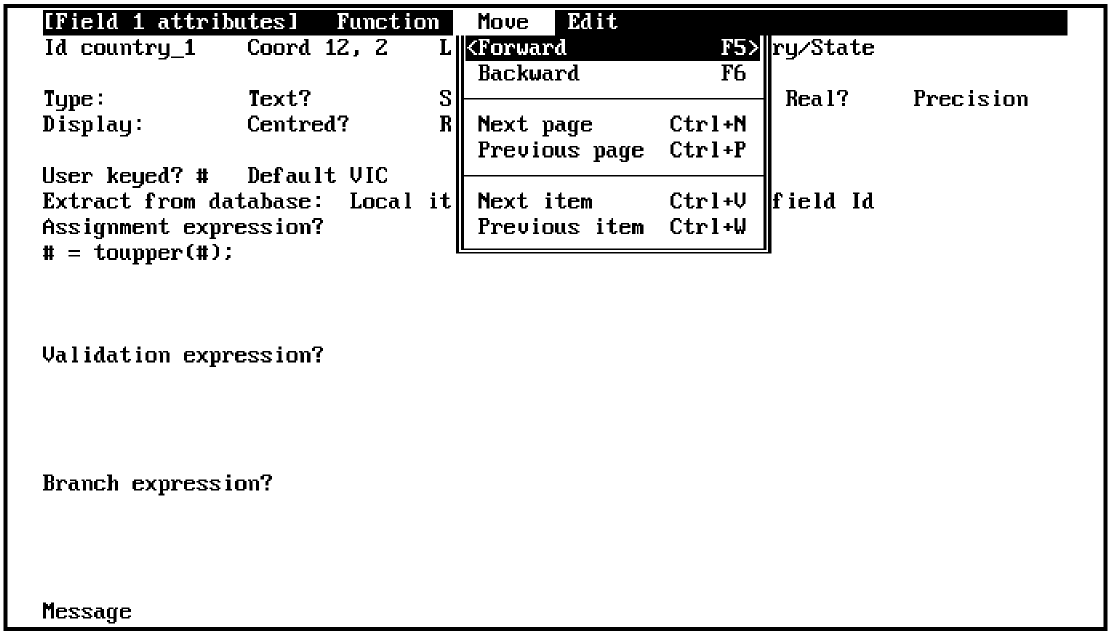
```
Copy attributes of field number:
```

The user must then enter the appropriate field number followed by ↵. To abort the copy, just enter ↵.

Exit Attributes Screen ([Item/Field*n* attributes] Function⇒ Exit)

Exit from attribute editing and return to the Insertion Form.

Moving around the Attributes screen



```
[Field 1 attributes]   Function   Move   Edit
Id country_1     Coord 12, 2    L <Forward          F5>  ry/State
                                   Backward          F6
Type:            Text?          S                         Real?     Precision
Display:         Centred?       R  Next page      Ctrl+N
                                   Previous page  Ctrl+P

User keyed? #    Default VIC
Extract from database:  Local it   Next item      Ctrl+V  field Id
Assignment expression?             Previous item  Ctrl+W
# = toupper(#);



Validation expression?




Branch expression?




Message
```

[Field 1 Attributes] Move Menu

Next Attribute ([Item/Field*n* attributes] Function⇒ Forward)

Move forward to the next attribute which may be edited. This command cycles between attributes on the current page.

Previous Attribute ([Item/Field*n* attributes] Function⇒ Backward)

Move backward to the next attribute which may be edited. This command cycles between attributes on the current page.

Next Attribute Page ([Item/Field*n* attributes] Function⇒Next page)

Move to the first attribute which may be edited on the next page. Disabled for header and box attribute editing.

Previous Attribute Page ([Item/Field*n* attributes] Function⇒Previous page)

Move to the first attribute which may be edited on the previous page. Disabled for header and box attribute editing.

Next Item ([Item/Field*n* attributes] Function⇒ Next item)

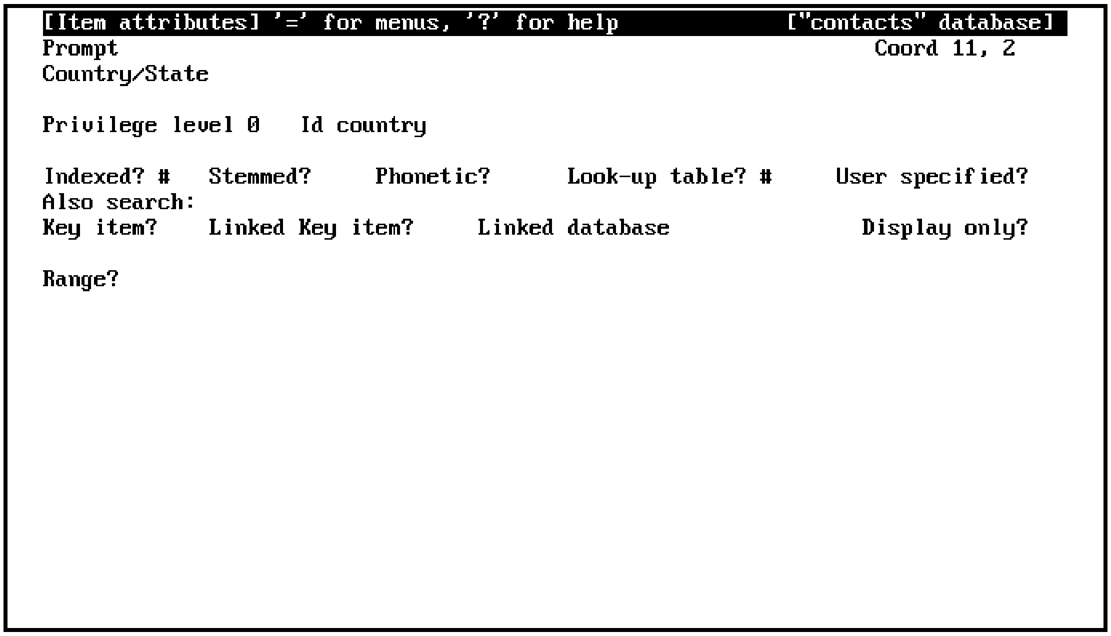
Commence attribute editing on the next item, header or box in the form order.

Previous Item ([Item/Field*n* attributes] Function⇒ Previous item)

Commence attribute editing on the previous item, header or box in the form order.

Item Attributes

The item attribute page consists of several fields as displayed in the following figure :



```
[Item attributes] '=' for menus, '?' for help          ["contacts" database]
Prompt                                                        Coord 11, 2
Country/State

Privilege level 0   Id country

Indexed? #   Stemmed?     Phonetic?      Look-up table? #     User specified?
Also search:
Key item?    Linked Key item?     Linked database              Display only?

Range?
```

[Item attributes]

Prompt

The item prompt may be modified using Text edit mode commands. Note that it is not possible to leave this attribute field if the altered prompt is identical to another prompt on the same form. Also it is not permissible to delete the prompt.

Privilege level

To be able to view an item when using KE Texpress, a user must have a privilege level higher than or equal to (numerically less than or equal) the item privilege level. For example, an item with a privilege level of 2 can be viewed only by users with a privilege level of 0, 1 or 2. The maximum number of privilege levels is nine. An integer value may entered, ranging from zero, up to but not including, the number of privilege levels specified in the database configuration. The item privilege level applies to the entire item, namely the prompt and all fields.

Note that for multi-level hierarchy items, it is possible to edit only the privilege level of the first item in each hierarchy. Any change to this item's privilege level is propagated down the hierarchy to all other items.

Id

The item Id is a system identifier associated with an item. This identifier allows the item to be referenced by other items/fields. Each item must have an Id and it must be unique in the database. The item Id can also be used in automatic database loading.

If possible an item Id is generated automatically from the item's prompt. The default Id so derived contains the alphanumeric characters of the prompt, truncated if necessary to a length of 8. If the default Id so formed is not unique, or does not begin with an alphabetic character, a unique Id of the form "id*n*" where *n* is the next available integer) is assigned to the item.

If an item Id is entered explicitly, it must begin with an alphabetic character. Form validation detects any invalid or non-unique item identifiers.

KE Texpress library

If the item is one of the KE Texpress Library types, this attribute is displayed along with the the name of the particular Library. It is a dispaly only field and cannot be edited.

Automatic

Automatic is a boolean attribute which can be set only for certain KE Texpress Library types (refer to Appendix C). During data entry, **texforms** automatically generates values for items which have this attribute set.

Indexed

The Indexed attribute specifies that during a record insertion, the fields of the current item should be used to create an internal descriptor for the record. This descriptor provides the main mechanism by which records are retrieved. Indexing should be turned on only for items which are likely to be specified in a query and which (in general) have a large set of values.

Stemmed

This attribute may be set only on items whose fields are of Text type (see Field Attributes). If stemmed is set, then index support is provided for stemmed values. It should be set only on items where stemmed queries are likely to be specified. If this attribute is turned on, the Phonetic attribute cannot be turned on and vice-versa.

Phonetic

This attribute may be set only on items whose fields are of Text type (see Field Attributes). If Phonetic is set then index support is provided for phonetic (sounds-like) queries. It should be set only on items where phonetic queries are likely to be specified. If this attribute is turned on, the Stemmed attribute cannot be turned on and vice-versa.

Look-up table

The Look-up table attribute indicates that a table of all unique values entered for the item is to be maintained. This is used to define dynamically the vocabulary of an item. The order of entries within the Look-up table is dependent upon the Type attribute (see Field Attributes). Text and String fields are sorted alphabetically. Integer and Real fields are sorted numerically.

For multi-item hierarchies the Look-up table attribute is set automatically by the hierarchy creation command. Note that KE Texpress Libraries are not permitted to have Look-up tables.

User specified

This attribute may on rare occasions be used in conjunction with the Look-up table attribute. If set, it indicates that the order of entries in the table is to be specified by the user. If so, at record insertion time, for any new Look-up table entry, KE Texpress will request the user to nominate the entry's position relative to positions of the other Look-up table entries.

When a Lookup table is consulted, values in the table are displayed in order of their position in the table. In the great majority of cases, the convenient order is alphabetic for Text and String fields, and numeric for Integer and Real fields. On rare occasions there may be an unusual requirement for a user specified order. Note that rebuilding a user specified Lookup table destroys the user specified order.

Also search

This attribute is used to specify a list of other items on the form that will be searched whenever the current item is used in a query. The search of all selected items is performed using the Type attribute (see Field Attributes) and the indexing characteristics, of the current item. The required list of items is compiled using the Edit item list command on the Functions menu. (Refer to the Define Item Attributes - Edit Also Search Items section found earlier in this chapter).

Key type

This is displayed if the item is a Key type. (See the Forms Menu - Key Forms section in this chapter).

Key item

The Key item attribute indicates to KE Texpress that the content of this item uniquely identifies the record. In order for this attribute to be set, the item must be set up as one of the Key Types. (See the Forms Menu - Key Forms section in this chapter).

Linked Key item

Turning on the Linked Key item attribute indicates that this item is used to link to another database. This attribute may be set only if the item is a Key type. It must also be set up with the same Key type as the Key item in the other database. During record insertion, when a value is entered into a Linked Key item, KE Texpress searches the linked database for a record with that value as its Key value. If such a record is found, data is extracted from the record and copied into designated fields within the current record (this is described further in Field Attributes).

Note that after the extraction, if the record in the linked database, from which the data was extracted is edited, the data in the current record will not not reflect this change, unless the current record is re-edited and the Linked Key value is re-entered.

Linked database

The Linked database attribute indicates the name of the linked database to which the Linked Key item value applies.

Display only

If the Display only attribute is set the linked information is not stored with the current record but rather is extracted every time the record is displayed. This enables the current database to be updated dynamically by alterations to the information in the linked database, which will act as a Master list.

Range

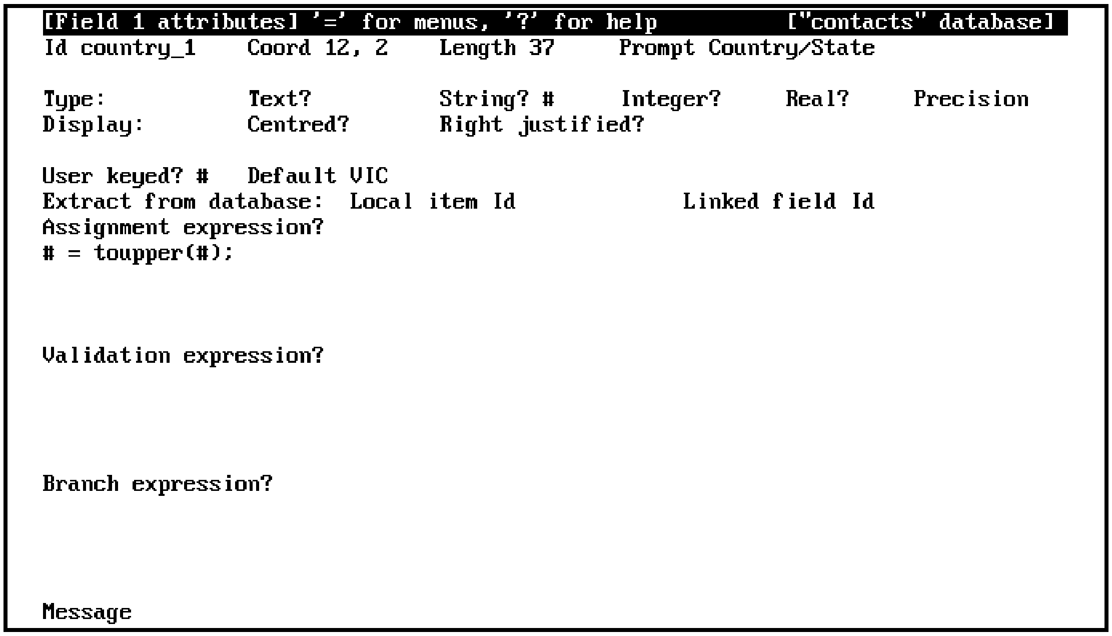
Setting the Range attribute indicates that the item is to be used for range queries. A range query is one in which the user specifies lower and upper bounds for the field value. All records that have a field value within the specified range are retrieved. Generally, a range item is allowed to have only a single field of numeric type. The exceptions to this rule are some of the KE Texpress library types which are specially designed to permit ranges (refer to Appendix C).

The area the of item attribute page directly below the Range attribute is for entering the actual ranges associated with the item. The maximum number of range intervals permitted is 36. The method of insertion is straightforward; all of the usual edit commands are once again used. The specified values indicate the intervals over which range queries are to be expected for the item. These range intervals enable KE Texpress to perform efficient range queries. The range intervals are automatically ordered numerically as they are entered.

If the item being ranged is a numeric type (Integer or Real), each range value must conform to the field type. If the item being ranged is a KE Texpress library, since the value of the entity that the item represents is specified by the ocmbination of values in all fields of the item, each range interval must specify the values of all fields in the item.

Field Attributes

Each field has the attributes, as shown in the following figure.



```
[Field 1 attributes] '=' for menus, '?' for help         ["contacts" database]
Id country_1    Coord 12, 2    Length 37     Prompt Country/State

Type:           Text?          String? #     Integer?     Real?     Precision
Display:        Centred?       Right justified?

User keyed? #   Default VIC
Extract from database:  Local item Id            Linked field Id
Assignment expression?
# = toupper(#);



Validation expression?




Branch expression?




Message
```

[Field 1 attributes]

Id

The field Id consists of the associated item Id, concatenated with the underscore character ('_') and the field number. Field numbers are assigned automatically, commencing at one. By default,refering and item's ID refers to its first field.

Coord

Displays the fieldcoordinates in line, column format.

Length

Displays the field length.

Prompt

Displays the item prompt (if applicable).

Type

The attributes on this line define the type of the field :

Text

A field of Text type accepts all printable characters. When searching for record matches, the data is interpreted as words and matching is performed on a word basis. Spacing, punctuation and character case is ignored. Also noise words (such as 'a', 'the', 'but' etc.) are ignored. Stemmed and Phonetic queries are also permitted. For the default noise word list refer to the Text Queries - Noise Words section in the KE Texpress User Guide.

String

A field of String type accepts all printable characters.When searching for record matches, the entire data in the field is treated as one value which must be matched exactly. Spacing and punctuation are significant. Character case is ignored.

Integer

The Integer field accepts an optional '+' or '-' sign followed by one or more digits. This attribute may not be set on fields greater than 15 characters in length.

Real

The Real field type accepts an optional '+' or '-' sign followed by one or more digits, followed by a decimal point '.', and one or more digits. This attribute may not be set on fields greater than 25 characters in length.

Precision

The Real field type may have an optional precision value specified, indicating the number of decimal places to be displayed. A precision value of 0 indicates that arbitrary precision is to be used for the field.

Display

These attributes define the display orientation of field values. By default all values are displayed left justified within the field. Also, whenever a record is edited, all field values are displayed left justified.

Centred

Field values are displayed centred within the field.

Right justified

Field values are displayed right justified within the field.

User keyed

The User keyed attribute indicates that the input for the field is to come from the user keyboard. If this is not set, the cursor can never move to the field during record insertion. User keyed is not set for fields which are filled automatically, calculated from other fields, or extracted from a linked database.

Default

A default value may be specified for the field. When each record insertion commences, KE Texpress traverses the form and automatically assigns all default values. This feature can save time if large numbers of insertions have the same value for a particular field. If the User keyed attribute is set for a field with a Default value, the value may be overridden from the terminal keyboard. Note that the default values must conform to the field type, and altering the field type results in the field default value being deleted.

Extract from database

This consists of the two attributes Local item Id and Linked field Id. If they are filled, then they are used to determine the field's data via a linked database access.

Local item Id

The Local item Id is the Id identifier of the Linked Key Item in the Insertion form of the <u>current</u> database. It is used to reference the linked database.

Linked field Id

The Linked field Id is the field for indicating the field in the linked database from which the data is to be extracted.

The linked field Id is the field identifier for the field in the linked database, from which the database is to be extracted and linked into the current field. During record insertion, if the User keyed attributed is set for the current field, and its associated Local Id item is not filled, then the cursor will visit the current field, and data may be entered into this field from the keyboard. If however, the associated Local Id item is filled, then data in the current field is replaced by data extracted from the field specified by the Linked field Id in the linked database, and KE Texpress will not permit the cursor to travel to the current field.

Assignment expression

It is possible to associate with a field an expression, i.e. a small program, which specifies some instructions to generate a value for the current field itself, or for other fields. Assignment expressions are evaluated when a record insertion begins, or whenever data is altered in a field referred to in the assignment expression.

([Field *n* attributes] Function⇒ Edit expression)

The command calls up the designated Unix text editor for entering, viewing or modifying the expression.

Validation expression

Regardless of the source of the data in a field (e.g. direct entry, assignment, batch load), all data entered into a record must pass validation before the record can be stored. It is possible to associate with a field an expression, i.e. a small program, which specifies some instructions to determine the validity of the data in the field.

Validation expressions are evaluated when the command to save a record is run, before the data is actually saved. If no validation expression is specified for a field, then data in the field is always valid.

A validation expression uses the reserved variable **val** to indicate the validity of the data. The data is valid if the variable val is asigned a true (non-zero) value, but invalid if the variable is assigned a fasle (zero) value. An example of a validation expression is:

```
if (# > 110)
    val = 0;
```

([Field *n* attributes] Function⇒ Edit expression)

The command calls up the designated Unix text editor for entering, viewing or modifying the expression.

Branch expression

The form order determines the traversal sequence through the fields on the form (See the Form Order section in this chapter). It is possible to associate with a field an expression, i.e. a small program which specifies some instructions to vary the cursor traversal sequence through the form. Branch expressions are evaluated when there is movement from one field to another, for example, when ⇒ key is entered.

([Field *n* attributes] Function⇒ Edit expression)

The command calls up the designated Unix text editor for entering, viewing or modifying the expression.

A branch expression uses the inbuilt function **addr()** to assign the address of the required field to the reserved variables **next** and **prev**, to indicate the destination of the cursor after the current item during forward or backward traversal of the form. The branch expression may state absolutely that the cursor is to visit a particular field after the current one, or the destination of the cursor may be altered dynamically depending on the content of the current field or some other field. An example of a dynamic alteration in the traversal sequence is as follows:

```
if (# == "y")
    next = addr(name);
```

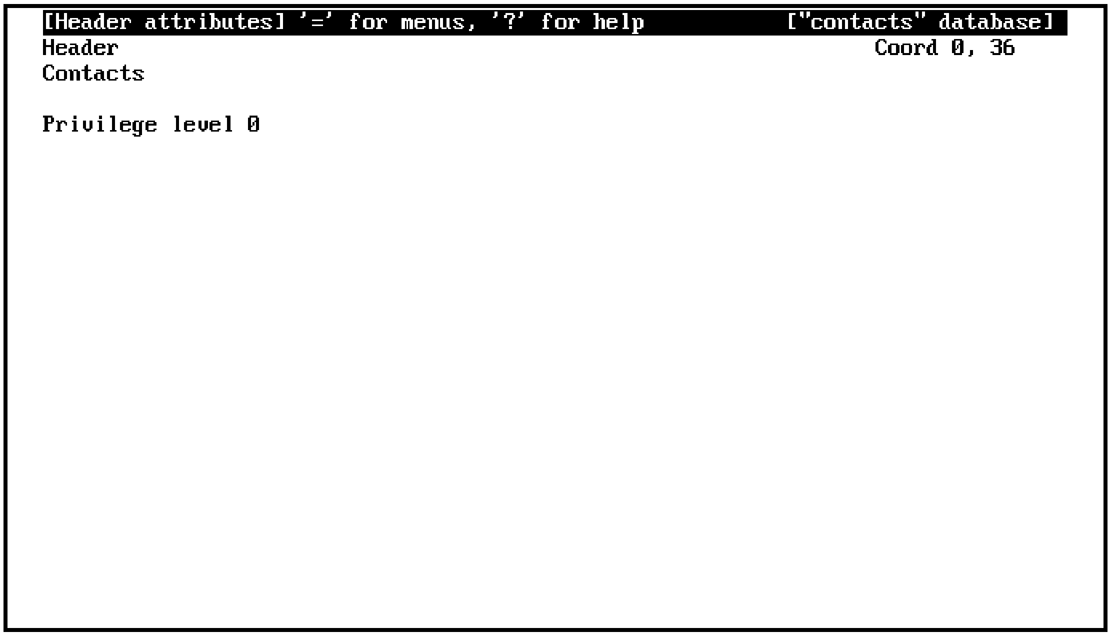
(If the field value is **y**, then traversing forward from this field would send the cursor to the first field of the **name** item.)

Message

This attribute accepts text, which is displayed on the menu bar whenever data is being entered into the field. This provides a simple, flexible, user prompt mechanism. Note that is not possible to insert text into this attribute if the User keyed attribute is not set.

Header Attributes

The header attribute page is structured as in the following figure.



```
[Header attributes] '=' for menus, '?' for help          ["contacts" database]
Header                                                          Coord 0, 36
Contacts

Privilege level 0
```

[Header attributes]

Header

The header text may be modified using the usual edit commands. Note that it is not possible for the cursor to leave this attribute if the altered text clashes with another form object. Also it is not permissible to delete the text.

Privilege level

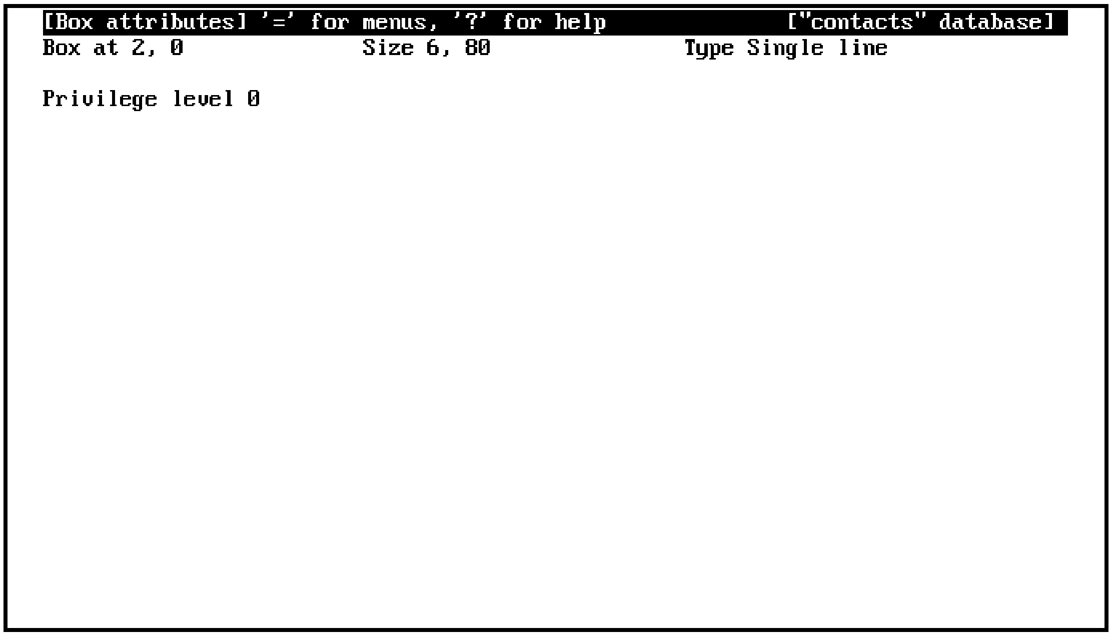
To be able to view a header when using KE Texpress, a user must have a privilege level higher than or equal to (numerically less than or equal) the header privilege level. For example, a header with a privilege level of 2 can be viewed only by users with a privilege level of 0, 1 or 2. The maximum number of privilege levels is nine. The privilege level has an integer value, ranging from zero up to (but not including) the number of privilege levels specified in the database configuration.

Coord

Displays the headercoordinates in line, column format.

Box Attributes

The box attribute page is structured as in the following figure.



```
[Box attributes] '=' for menus, '?' for help          ["contacts" database]
Box at 2, 0              Size 6, 80               Type Single line

Privilege level 0
```

[Box attributes]

Box at

Coordinates of the top left corner of the box in line, column format.

Size

Size of the box in lines, columns.

Type

The type of the box, single or double line.

Privilege level

Similar to item and header privilege levels.

Expression Grammar

KE Texpress's expression grammar permits powerful expressions to be specified for field assignment, validation and branching. The expressions syntax closely follows that of the C programming language. Block structure, temporary variables, **if-else** constructs, **while** loops and **for** loops are supported. Most of the standard C language operators are provided.

The expression grammar also draws on some of the concepts of the Unix facility **awk.** Explicit declaration of variables is not required and variables can be both string and numeric simultaneously. The string value is the default representation and a numeric value is calculated from the string. Strings which are not in a correct number format assume the value zero. An assignment to an identifier which does not match any item or field Ids creates a new variable of that name. Note that temporary variables do not maintain values across separate invocations of an expression.

An extensive set of inbuilt functions which can used in any expression is also provided. This set of functions simplifies the development of more complicated expressions.

For further assignment, validation and branch expression examples refer to Appendix F. For a complete listing of the expression grammar refer to Appendix G. For a list of the in-built constants and functions refer to Appendix H.

Display Mode Form Expression ([Insertion form] Define⇒Form expression)

Associated with the form may be an expression which can be executed whenever matching records are displayed. This expression can perform varied operations. The syntax of the expression is the same as that used for field assignment, validation and branch expressions.

This command calls up the designated Unix text editor, and the current form expression may be edited and saved.

Hierarchies

Hierarchical relationships between items are permitted by KE Texpress. KE Texpress maintains Look-up tables for all hierarchical items and remembers the allowable combinations of values in the hierarchical relationships. These combinations are set up as data is entered into the database (refer to the KE Texpress User Guide).

There is no limit to the number of items in a hierarchy. However some restrictions apply to the types of items allowed. The restrictions are:

- For multi-item hierarchies, only the item at the lowest level of the hierarchy may have more than one field.
- An item can not be involved in more than one hierarchy.
- KE Texpress Library items are not permitted in hierarchies.

Create hierarchy ([Insertion form] Define⇨Create hierarchy)

The menu bar displays the message:

```
Select items in hierarchy
```

and items may be selected in order of the hierarchy. The top level hierarchy item should be selected first, followed by the lower levels in turn. Selecting a null (blank) area on the form indicates that all items in the new hierarchy have been selected. Items in a multi-level hierarchy must be at the same privilege level as the first item in the hierarchy. Privilege levels are automatically adjusted if this is not the case.

Delete hierarchy ([Insertion form] Define⇨Destroy hierarchy)

The hierarchy encompassing the item which the cursor is marking is destroyed. If the marked item is not in a hierarchy, an error message is displayed. Destroying hierarchies does not actually remove the items from the form, it simply removes the hierarchical relationship. The Look-up table attribute remains set for each item.

View hierarchies ([Insertion form] Define⇨ View hierarchies)

This command provides a visual indication of the hierarchical relationships existing in the form. Each item in each hierarchy is successively highlighted. After each complete hierarchy is displayed, the message:

```
Hierarchy n
```

is displayed. Type any key to move on to the next hierarchy. The value ***n*** is the number of the hierarchy in the form order. This number has no meaning outside the scope of this command.

Form Bounds

When a form is created, it is set to the maximum size that fits neatly into the form window (typically 23 lines by 80 columns). In many cases this is sufficient for the form. If not, the following command may be used to alter this default form size.

Define Form Boundaries ([Insertion form] Define⇒ Form bounds)

Change the form size. A sequence of dots is displayed indicating the rightmost and bottom boundaries of the form if the form size in the respective dimension is less than the size of the terminal screen. There is no limit to the size of a form.

The cursor is positioned at the bottom rightmost corner of the form. The simple cursor movement commands are then used to relocate the form boundaries, so altering the form size. Alternatively, the following commands can be used:

Move boundary left ([Form size] Function⇒ Move left)

Move the right form boundary left, decreasing the width of the form. It is not possible to move the boundary past the edge of the right most form object.

Move boundary right ([Form size] Function⇒ Move right)

Move the right form boundary right, expanding the width of the form.

Move boundary up ([Form size] Function⇒ Move up)

Move the bottom form boundary up, decreasing the length of the form. It is not possible to move the boundary past the lowest form object.

Move boundary down ([Form size] Function⇒ Move down)

Move the bottom form boundary down, increasing the length of the form.

Display Coordinates ([Form size] Function⇒ Form coordinates)

This command displays the current form size, form view and cursor coordinates. The display has the format:

Size (*lines*, *cols*), View (*y*, *x* : *my*, *mx*), Cursor (*y*, *x*)

Size lists the number of lines and columns in the form. View indicates the form window coordinates for the current screen and cursor gives are the current form line and column position of the cursor.

Exit Form Boundaries ([Form size] Function⇒ Exit)

Return to form editing. The adjusted form size becomes the current form size.

Form Order

The form order is the order in which items are visited during data insertion and query. By default, the order is determined by the sequence in which new items are added; new items are always placed at the end of the form order. However the order in which items are to be visited may be changed any time. Headers and boxes play no part in the form order. They are for display purposes only.

Define Form order ([Insertion form] Define⇒ Form order)

Specify (or re-specify) the form traversal order. The menu bar displays:

```
Select items in sequence
```

The user should select each item in the order desired. Selecting a blank area of the form signifies the end of the ordering procedure. Confirmation that form ordering is complete is then requested. All items need not be selected; unselected items remain in the same relative order.

Each item is highlighted as it is selected. Re-selecting selected items or trying to mark headers or boxes results in an error message. Upon termination of the selection procedure, or when all items are selected, a message will display:

```
Re-ordering form . . .
```

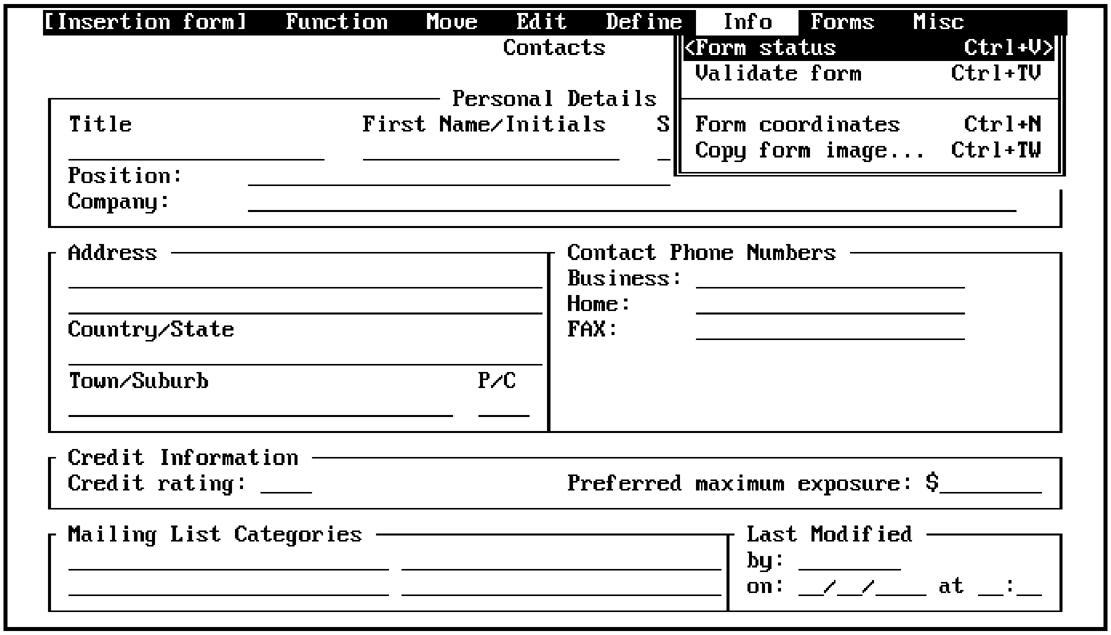
The form order will then be adjusted.

In order to ensure consistent behaviour when data is entered into multi-level hierarchies, it is recommended that hierarchy items be ordered consecutively in parent to child sequence. When hierarchies are created, the form order is silently adjusted to ensure that this relationship holds. However this default order may be altered. This is the only command which alters the form order. All other commands (including undo) preserve the order of the form.

View Form order ([Insertion form] Define⇒ View order)

This command allows the user to view the current form traversal order. The cursor is positioned in turn at each item, and the item is highlighted. After a brief pause, the next item in the form order is displayed.

Info Menu



```
[Insertion form]  Function  Move  Edit  Define  Info  Forms  Misc
                                     Contacts  <Form status          Ctrl+V>
                                               Validate form         Ctrl+TV
                         Personal Details
 Title               First Name/Initials    S  Form coordinates      Ctrl+N
 ________________    ____________________   _  Copy form image...    Ctrl+TW
 Position:     _______________________________
 Company:      ____________________________________________________________

 Address                                   Contact Phone Numbers
 _____________________________________     Business: _____________________
 _____________________________________     Home:     _____________________
 Country/State                             FAX:      _____________________
 _____________________________________
 Town/Suburb                      P/C
 ______________________________   ____

 Credit Information
 Credit rating: ____                        Preferred maximum exposure: $________

 Mailing List Categories                                Last Modified
 _________________________   _________________________  by: ________
 _________________________   _________________________  on: __/__/____ at __:__
```

[Insertion form]Info Menu

Display Form Status ([Insertion form]nfo ⇒ Form status)

Display the form name (if applicable), form type (Insertion, Query, Key, Library or Report) and the current status of the form.

The form status flags are:

[Read only]

The form is read only and editing can not be performed.

[Modified]

The form has been modified since last being written to file.

[Init]

The database has beeninitialized.

[Rest]

The command set has been temporarily restricted to ensure that the changes are not performed which would require a database rebuild.

[Auto]

An automatic rebuild is required before access to the User menu will be permitted.

Validation

Form validation is used to check the consistency of the form. All item and field linked database attributes are verified for consistency. Field assignment, validation and branch validation expressions are also checked to verify that they conform to the grammar.

A complete list of form validation error messages along with explanations is provided in Appendix I.

Validate form ([Insertion form]nfo ⇒ Validate form)

The message:

```
Validating [Insertion form] . . .
```

is displayed. A validation report is generated and displayed on the screen a page at a time (using the Text perusal facility). The validation report lists items in form traversal order and provides a verbose description of all form objects. Any detected errors are displayed at the start of the report. It is possible to have a copy of the report written to a file or sent to the printer.

After the validation report has been viewed, a message is displayed indicating the validity of the form.

Display Form Coordinates ([Insertion form]Info ⇒ Form coordinates)

This command displays the current form size, form view and cursor coordinates. The display has the format:

```
Size (lines, cols), View (y, x : my, mx), Cursor (y, x)
```

Size lists the number of lines and columns in the entire form. However the order in which items are to be visited may be changed any time. View indicates the form window coordinates for the current screen. Cursor gives are the current form line and column position of the cursor.

Privilege Level

For databases configured for multiple privilege levels, it is possible to view the form as it would be displayed for each distinct privilege level. This allows simple visual verification of privilege level scope. This command is disabled if the database has only one privilege level. Privilege level scope is hierarchical. For example, level 0 encompasses all levels 0 to n; level 1 encompasses all levels 1 to n, and so on.

View at Privilege level ([Insertion form]Info ⇒ View at level)

A selection box prompts for the desired privilege level for viewing the form to be entered:

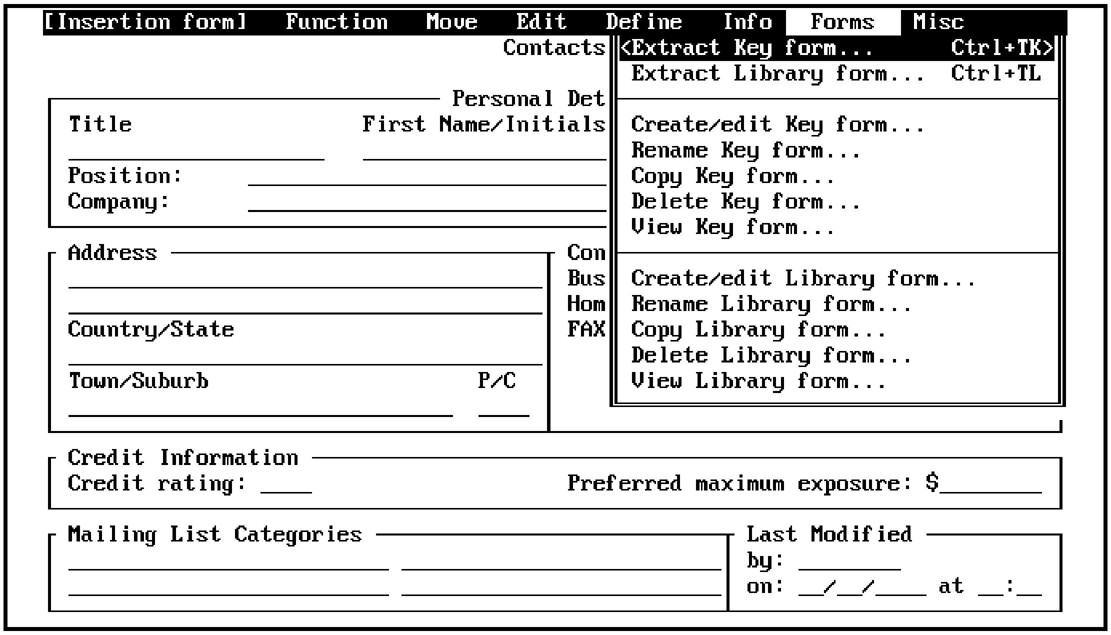
```
Viewing at privilege level  c. Change to [0 -  n] :
```

where ***c*** represents the current privilege level and ***n*** the maximum permissible privilege level as specified in the database configuration. Only those objects which are encompassed by that privilege level are displayed. If required, the form may be scrolled in order to view other areas of it. If the privilege level specified is anything other than level 0, **read only** forms editing is invoked.

Copy Form Image ([Insertion form]Info ⇒ Copy form image)

A copy of the complete form can be sent to a printer or copied to a Unix file.

Forms Menu



```
[Insertion form]  Function  Move  Edit  Define  Info  Forms  Misc
                              Contacts <Extract Key form...      Ctrl+TK>
                                        Extract Library form...  Ctrl+TL
                          Personal Det
 Title          First Name/Initials     Create/edit Key form...
                                        Rename Key form...
 Position:                              Copy Key form...
 Company:                               Delete Key form...
                                        View Key form...
 Address                        Con
                                Bus     Create/edit Library form...
                                Hom     Rename Library form...
 Country/State                  FAX     Copy Library form...
                                        Delete Library form...
 Town/Suburb              P/C           View Library form...

 Credit Information
 Credit rating: ____              Preferred maximum exposure: $________

 Mailing List Categories                          Last Modified
                                                  by: ________
                                                  on: __/__/____ at __:__
```

[Insertion form] Forms Menu

Key forms

A Key form is a form specifically pre-designed to permit database key and linked key access. Each Key form contains just one item and defines one particular Key type. Once a Key type is extracted into an Insertion form, its attributes may be edited to specify that it is the Key item for the database, or a linked Key to another database, or both.

An Insertion form may contain any number of items which are extracted Key types, but only one of them may be the Key item. If an Insertion form contains a Key item, then each database record must have a unique value in that item. KE Texpress enforces uniqueness for Key item values. For certain Key types, KE Texpress can generate Key item values, either automatically as each insertion commences, or, when users run a special command at any stage during insertion of a record.

KE Texpress provides a number of Key forms, which may not be edited or deleted. In addition, database administrators may design further Key forms, which may subsequently be edited or deleted only by the administrator who created them. For a description of the Key types provided in the KE Texpress release, refer to Appendix B in the KE Texpress Design Guide or the KE Texpress User Guide. Because all Key forms reside in the one directory and are available to all DBAs during form design, each Key form must have a name unique among Key forms.

The Forms pull-down Menu in Insertion form mode contains options which allow a database administrator to manipulate the various Key types.

Extract key form ([Insertion form] Forms⇒ Extract key form

This command lists the key forms, in a selection box, from which the user can select one by entering ↵. This extracts the key form into the current cursor position on the Insertion form. Note that the entire area in the Key Form is extracted into the Insertion form.

Modify key form ([Insertion form] Forms⇒ Create/edit Key form)

Create a new Key form, or edit an existing Key form. A selection box is displayed, headed:

```
Create/Edit Key Form
```

and listing all the existing Key forms. The Key Forms provided as part of the KE Texpress release, and those owned by other users may not be edited. All other key forms may be selected from the list, (enter TAB to move the cursor to the selection list), or a new name can be entered into the lower box (enter TAB to flip between the upper and lower boxes). If the name entered is that of a Key type not available for editing, the following message is displayed:

Can't create " *keyname*" Key form

Otherwise editing commences, either on an existing Key form, or on a new (blank) Key form.

Each Key form must contain only one item, which may consist of any number of fields. Whereas in Insertion form editing, all fields of the one item must be of the same type, in Key form editing it is possible for different fields of the one item to have different types.

It is not possible to set the Key item attribute, or the Linked Key item attribute during Key form editing. These attributes may be set only in Insertion form editing, after the designed Key type item has been extracted into the Insertion form.

When a previously designed Key form is extracted into the Insertion form, it occupies the area determined by its layout in Key form design. It is important therefore that the Key form item be located at the top left hand corner of the form, and the form bounds should be flush with it, so that the Key form occupies the minimum space possible.

The commands possible in Key form editing is very similar to those used in Insertion form editing. Some commands available in Insertion form editing are disabled as they are inappropriate for Key form editing.

An additional menu option is provided for Key form editing. If the Key form item has fields of integer type, a key item incrementation order may be defined. This incrementation order is used by KE Texpress to determine the next available key, which may be generated automatically. The Define Menu in Key form editing provides an additional option for the specification of this order.

Define key order ([Key form] Define⇒ Key order)

Define the Key item incrementation order. A message on the Menu bar prompts the user:

```
Select fields in  incrementation order
```

Each incrementing field should be selected in turn, from the first incrementing to the last incrementing. If an attempt is made to select a non integer field, an error message is displayed:

```
Only integer fields can be incremented
```

The administrator indicates that all required fields have been marked by marking a blank area. A question is displayed:

```
All fields selected?
```

A Yes response terminates the selection procedure.

When Key form editing has been completed, the Key form is saved, and control returns to Insertion form editing.

Rename key form ([Insertion form] Forms⇒Rename Key form)

Change the name of a Key form. A selection box is displayed, headed:

```
Rename Key Form
```

and listing all the existing Key forms. The Key form whose name is to be changed is selected from this list. Another selection box is displayed, headed:

```
Enter new Key Form name
```

The new name may be entered. It must be unique among Key form names. If it is not, a message reminds the administrator of this before aborting the rename procedure:

"*newname*" Key form already exists

Otherwise, confirmation of a successful name change is displayed:

"*oldname*" Key form renamed " *newname*"

Copy key form ([Insertion form] Forms⇒Copy Key form)

Copy an existing Key form. A selection box is displayed, headed:

```
Copy Key Form
```

and listing all the existing Key forms. The Key form to be copied is selected from this list. Another selection box is displayed, headed:

```
Enter new Key Form name
```

The new name may be entered. It must be unique among Key form names. If it is not, a message reminds the administrator of this before aborting the rename procedure:

"*copyname*" Key form already exists

Otherwise, confirmation of a successful copy is displayed:

"*keyname*" Key form copied to " *copyname*"

If a new Key form is to be designed, it is often desirable to begin with a copy of an existing form. Fields may then be added or deleted as required, and attributes adjusted as required.

Delete Key form([Insertion form] Forms⇨Delete Key form)

Delete a Key form. A selection box is displayed, headed:

```
Delete Key Form
```

and listing all the Key forms. The Key form to be deleted is selected from this list. The pre-designed Key forms provided with the Texpress release, and all other Key forms not owned by the current administrator, are unavailable for deletion. When an available Key form is selected, a question box requests confirmation of the intention to delete:

Delete " *keyname*" Key form?

After a Yes response, confirmation of the deletion is displayed:

"*keyname*" Key form deleted

View Key form([Insertion form] Forms⇨View Key form)

View a Key form. A selection box is displayed, headed:

```
View Key Form
```

and listing all the Key forms. The Key form to be viewed is selected from this list. The Key form selected need not be owned by the Administrator. Viewing is performed using texdesign in read only mode. All cursor commands are provided, but no actual editing is permitted.

Library forms

A Library form is a form containing a set of items, headers and boxes with associated attributes, pre-designed to provide a frequently used object, which can be extracted by DBAs during Insertion form design. Library forms remove the need for repeated creation of commonly used objects every time they are needed in an Insertion form, and help to promote integrity and consistency across database designs.

The KE Texpress release provides a number of Library forms, which may not be edited, copied, renamed or deleted. In addition, database administrators may design further Library forms, which may subsequently be edited or deleted only by the administrator who created them. For a description of the Libraries provided in the KE Texpress release, refer to Appendix C in the KE Texpress Design Guide or the KE Texpress User Guide. Because all the Library forms are in the one directory, and are available to all DBAs during form design, each must have a name that is unique among Library forms. The Library forms provided with the KE Texpress release may have fields of different types. This property is not available to other Library forms.

The Forms Menu in Insertion form mode contains options which allow a database administrator to manipulate the various Libraries.

Extract library form ([Insertion form] Forms⇒ Extract library form)

This calls up a selection box, listing all the library forms, from which the user can select one by entering ↵. This extracts the entire key form into the current cursor position on the Insertion form.

Modify library form ([Insertion form] Forms⇒Create/edit Library form)

Create a new Library form, or edit an existing Library form. A selection box is displayed, headed:

```
Create/Edit Library Form
```

and listing all the existing Library forms. Neither the Library Forms provided as part of the KE Texpress release, nor those owned by someone other than the current administrator, are available for editing. An available name may be selected from the list, or a new name may be entered manually. If the name entered manually is that of a Library form not available for editing, the following message is displayed:

```
Can't create " libname" Library form
```

Otherwise editing commences, either on an existing Library form, or on a new empty Library form.

The command set in Library form editing is very similar to that used in Insertion form editing. Some commands available in Insertion form editing are disabled, as they are inappropriate for Library form editing.

When a previously designed Library form is extracted into the Insertion form, it occupies the area determined by its form area in Library form design. It is important, therefore, in Library form design, that the Library form object should be located at the top left hand corner of the form, and that the form bounds be closed around it, so that the form occupies the minimum space possible. When extracted, the object can then be placed neatly and compactly at the current cursor position. Because all Library forms reside in a common directory and are available to all DBAs during form design, each Library form must have a unique name.

When Library form editing has been completed, the Library form is saved, and control returns to Insertion form editing.

Rename ([Insertion form] Forms⇒Rename Library form)

Change the name of a Library form. A selection box is displayed, headed:

```
Rename Library Form
```

and listing all the existing Library forms. The Library forms provided with the KE Texpress release are not available for renaming. However other Library forms may be renamed. The Library form whose name is to be changed may be selected from the available names on this list. Another selection box is then displayed, headed:

```
Enter new Library Form name
```

The new name may be entered. The name must be unique among Library form names. If it is not, a message reminds the administrator of this before aborting the rename procedure:

```
"newname" Library form already exists
```

Otherwise, confirmation of a successful name change is displayed:

```
"oldname" Library form renamed " newname"
```

Copy ([Insertion form] Forms⇒ Copy Library form)

Copy an existing Library form. A selection box is displayed, headed:

```
Copy Library Form
```

and listing all the existing Library forms. The Library forms provided with the KE Texpress release are not available for copying. However other Library forms may be copied. The Library form to be copied may be selected from the available names on this list. Another selection box is then displayed, headed:

```
Enter new Library Form name
```

The new name may be entered. The name must be unique among Library form names. If it is not, a message reminds the administrator of this before aborting the rename procedure:

"*copyname*" Library form already exists

Otherwise, confirmation of a successful name change is displayed:

"*libname*" Library form copied to " *copyname*"

If a new Library form is to be designed, it is often desirable to begin with a copy of an existing form. Fields may then be added or deleted as required, and attributes adjusted as required.

Delete ([Insertion form] Forms⇒ Delete Library form)

Delete a Library form. A selection box is displayed, headed:

```
Delete Library Form
```

and listing all the Library forms. The Library form to be deleted is selected from this list. The pre-designed Library forms provided with the Texpress release, and all other Library forms not owned by the current administrator, are unavailable for deletion. When an available Library form is selected, a question box requests confirmation of the intention to delete:

Delete " *libname*" Library form?

After a Yes response, confirmation of the deletion is displayed:

"*libname*" Library form deleted

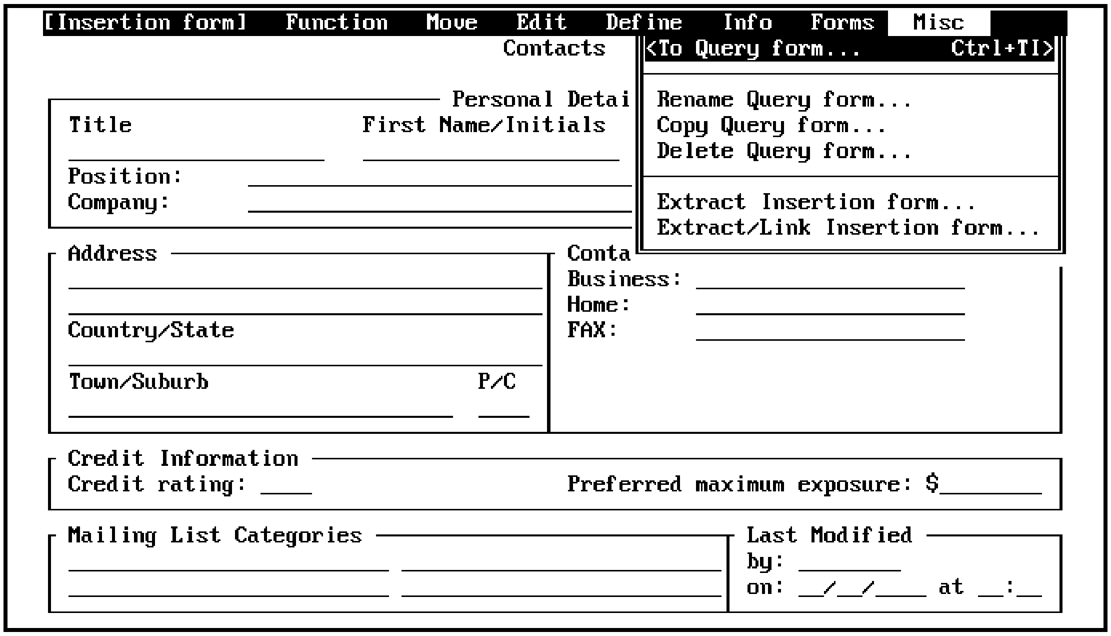
View ([Insertion form] Forms⇒ View Library form)

View a Library form. A selection box is displayed, headed:

```
View Library Form
```

and listing all the Library forms. The Library form to be viewed is selected from this list. The Library form selected need not be owned by the Administrator. Viewing is performed using texdesign in read only mode. All cursor commands are provided, but no actual editing is permitted.

Misc Menu



```
[Insertion form]  Function  Move  Edit  Define  Info  Forms  Misc
                                    Contacts  <To Query form...      Ctrl+TI>

                         Personal Detai  Rename Query form...
 Title               First Name/Initials  Copy Query form...
 ___________________  ___________________  Delete Query form...
 Position:   ____________________________
 Company:    ____________________________  Extract Insertion form...
                                           Extract/Link Insertion form...
 Address ________________________________  Conta
 ________________________________________  Business:  ____________________
 ________________________________________  Home:      ____________________
 Country/State                             FAX:       ____________________
 ________________________________________
 Town/Suburb                        P/C
 ______________________________     ____

 Credit Information ______________________________________________________
 Credit rating: ____                  Preferred maximum exposure: $________

 Mailing List Categories _________________________  Last Modified ________
 _________________________  _________________________  by: ________
 _________________________  _________________________  on: __/__/____ at __:__
```

[Insertion form]Misc Pull-down Menu

Query form editing

Access to Query form editing is available only from Insertion form editing. There may be several Query forms associated with a database. One of these is the default Query form, and the others are referred to by name. The Misc Menu in Insertion form editing provides various commands to manipulate Query forms.

Switch to Query form ([Insertion form]Misc ⇒To Query form. . .)

Switch to Query form editing. If the Insertion form has been modified since the last save, a message box displays the form status and requests confirmation of the intended switch to Query form editing:

```
[Insertion form]  [Modified].  Really switch?
```

A No response aborts the switch. If the response is Yes, all changes made since the last save are discarded. If the Insertion form is invalid, it is not possible to switch to Query form editing, and a message advises of this:

```
Insertion form must be valid before creating a Query form
```

If the Insertion form is valid, the switch may proceed. A selection box is displayed, listing any existing named Query forms and the default Query form, whether or not that already exists. The Query form to be edited may be selected from this list, or the name of a new Query form may be entered manually. Since all Query forms but the default one are in the same directory, they must be named uniquely.

If the default Query form does not exist, then whether the default Query form is selected from the list, or a name is entered manually, the default Query form is automatically created. If the default Query form already exists, and a new name is entered manually, a Query form of that name is automatically created. When a Query form, whether the default one or a new named one, is automatically created, it contains certain objects from the Insertion form, laid out as in the Insertion form. These objects, extracted from the Insertion form by default, consist of all headers and boxes, the Key item if it exists, and all indexed items. Query form editing may then commence.

If the Query form selected already exists, whether it is the default one or a named one, a message box prompts for confirmation that the existing form is to be used:

```
Read [Query form]?
```

If the response is Yes, the existing Query form is read from file, and editing may commence. If the response is No, a second message box prompts for confirmation that the required Query form is to be re-created:

```
Create [Query form]?
```

A No response aborts the switch. If the response is Yes, the required Query form is created anew.

Refer to Chapter 5 for detailed information about Query form editing.

Rename ([Insertion form]Misc ⇒Rename Query form. . .)

Change the name of a Query form. A selection box is displayed, headed:

```
Rename Query Form
```

and listing all the existing Query forms. The Query form whose name is to be changed is selected from this list. Another selection box is displayed, headed:

```
Enter new Query form name
```

The new name may be entered. It must be unique among Query form names. If it is not, a message reminds the administrator of this before aborting the rename procedure:

```
"newname" Query form already exists
```

Otherwise, confirmation of a successful name change is displayed:

```
"oldname" Query form renamed " newname"
```

Copy ([Insertion form]Misc ⇒Copy Query form. . .)

Copy an existing Query form. A selection box is displayed, headed:

```
Copy Query Form
```

and listing all the existing Query forms. The Query form to be copied is selected from this list. Another selection box is displayed, headed:

```
Enter new Query Form name
```

The new name may be entered. It must be unique among Query form names. If it is not, a message reminds the administrator of this before aborting the copy procedure:

```
"copyname" Query form already exists
```

Otherwise, confirmation of a successful copy is displayed:

```
"queryname" Query form copied to " copyname"
```

Delete ([Insertion form]Misc ⇒Delete Query form. . .)

Delete a Query form. A selection box is displayed, headed:

```
Delete Query Form
```

and listing all the Query forms. The Query form to be deleted is selected from this list. A message box requests confirmation of the intention to delete:

```
Delete " queryname" Query form?
```

After a Yes response, confirmation of the deletion is displayed:

```
"queryname" Query form deleted
```

Extract Insertion form ([Insertion form]Misc ⇒Extract Insertion form...)

It is possible to extract, into the current Insertion form, a copy of the Insertion form of another database, provided the administrator has access to that other Insertion form. A message box will display, prompting:

```
Enter database name:
```

Once the name is entered, KE Texpress checks that the administrator has access to the Insertion form of the other database via a user account in that database. If not, a message to this effect is displayed:

```
You are not permitted to use the " dbname" database
```

If, while the other Insertion form is being extracted, a clash is detected, a message warning of this is displayed:

```
Clash at present position
```

and the extraction is aborted. Otherwise, the extraction proceeds. Only those objects which the administrator is privileged to view in the other database are extracted. All extracted objects are extracted relative to the current cursor position.

Extract/Link Insertion form ([Insertion form]Misc ⇒Extract/Link Insertion form. . .)

Extract the Insertion form of another database into the current Insertion form, assuming that the extraction is to set up a link to the other database.

If the other database has no Key item, the effect of this command is exactly the same as that of the Extract Insertion Form command.

If the other database does have a Key item, the item extracted from that Key item automatically has its attributes changed so that it becomes a Linked Key item, the linked database being the one from which the extraction was made. Each other field extracted also has its attributes changed automatically, so that the Local item Id is set to that of the newly created Linked Key item, and the Linked field Id is the field in the other database from which the current field was extracted.

This provides a quick and easy way to set up a link to another database. Any fields extracted from the other Insertion form but not required in the current form can then be deleted.

Messages displayed are the same as those for the Extract Insertion form command.

Interrupt a Command

Generally, for each user of Unix, a particular key is set as an interrupt key. This is frequently the **DELETE** key, but may be something else such as **Ctrl+C**. To determine what has been set as the interrupt key, the command **stty**, or **stty -a** may be used. Amongst the output will be the setting for **intr**, e.g. **intr=DEL**.

Once the interrupt key is set, and interrupt may be generated by typing that key. During non-critical stages of Form development, interrupts are permitted. A message will display:

```
Interrupt
```

The current command will be aborted and the form restored to its previous state. If interrupts are disabled, then pressing the interrupt key has no effect and the current command continues.

Form Development Procedure

In order to ensure time is not wasted when designing and modifying the forms associated with a new database, a standard development method is recommended. The DBA should have a good idea of the Insertion form layout before commencing development. This knowledge should include all of the various item types and the associated field types and lengths. The different types of indexing should be considered, as well as ranges and the available libraries. If the database requires a Key item then it is much simpler to insert a Key item during the initial form development rather than deciding one is necessary after initialization and data entry.

As each item is inserted it is recommended that its attributes be edited immediately. The designer should also keep a note of each item's identifier and characteristics to simplify item references. Copies of the form validation report can be generated at any time.

The form design does not have to be completed in one session. A partially completed form may be written to file and re-edited at a later date. The Query form should not be designed until the Insertion form is finalized.

Edit Initialised Database Forms

The database Insertion and Query forms may be edited even after the database is initialized and data entered. All form operations are permitted, although some alterations affect the relationship between forms and data, so that the automatic rebuilding facility may be required. These operations are:.

- Changing the indexing type for an existing indexed item.
- The alteration or creation of hierarchies.
- Creating a Key item or altering the existing Key item.
- Altering any range items.

These operations are referred to as major changes.

For more information on modifying an initialized database and performing the database restructuring (if necessary), refer to Chapter 9.

Chapter 5

Query Form Design

Overview

To design the Query form the DBA must first create the default Query form (see Chapter 4, To Query form). This default Query form may then be modified as required by changing or deleting headers, prompts, fields or boxes, and by inserting lower and/or upper range items. Objects may also be repositioned in the form.

Query form editing is very similar to Insertion form editing. Most of the Insertion form commands are available. However the mechanism of some commands is slightly different.

Switch Forms

All KE Texpress databases contain a Query form, called the default Query form. In addition to this default Query form, other Query forms may be designed and used. The following commands are used to switch between editing of Query forms and the Insertion form.

Switch to Insertion form ([Query form]Misc ⇒ To Insertion form)

The Insertion form is read and editing of that form may commence. If the Query form has been modified the form status and the prompt:

```
[Query form] [Modified]. Really switch? [y/n]
```

is first displayed. A `n` response aborts the command whereas a `y` response results in all changes made since the last write command being discarded.

Switch to Other Query form ([Query form]Misc ⇒ To Query form)

This command takes the user to another Query form. If the current Query form has been modified, the form status is displayed and confirmation is first requested. A selection box displaying all the query forms for the database will then pop up. To select an existing query form, the user need only move the cursor to the upper box using the **TAB** key, move to the form using the arrow keys, and press ↵. Otherwise, to create a new query form, with the cursor sitting in the lower box (use the **TAB** key to move to it from the upper box), enter a unique name.

If the Query form already exists, the user is prompted:

```
Read "name" [Query form]? [y/n]
```

Choosing yes reads in the Query form. Choosing no causes another prompt to display:

```
Create "name" [Query form]? [y/n]
```

Choosing yes here results in a new Query form being generated and editing of the new form may commence. Choosing no aborts the command.

Delete Query form
([Query form]Misc ⇒ Delete Query form)

A selection box pops up displaying all the query forms. To select the form, the user need only move the cursor to it using the arrow keys and press ↵. The default Query form or a named Query form currently being edited cannot be deleted. To abort the command, just exit (**Ctrl+X** or **F1**.

Query Form Edit menu

Extract Item([Query form] Edit⇨ Extract Item)

When a Query form is created, only the Key item and those items which have the Indexed or Range attribute set are copied across from the Insertion form. Any other non indexed items required may be copied across to a Query form. Alternatively an item previously deleted from the Query form may be re-installed.

Before running this command, first place the cursor at the desired location of the extracted item. A selection box will then display the Ids of all the items on the insertion form. To select the item, move the cursor to it using the arrow keys and press ↵. The default Query form or a named Query form currently being edited cannot be deleted. To abort the command, just exit (**`Ctrl+X`** or **`F1`**). The extracted item will be placed on the Query form at the current cursor position. If the item already exists in the Query form or insufficient space is available an error message will display.

Insert Header ([Query form] Edit⇨ New Header)

This command is used to insert headers into the form and is identical in procedure to inserting headers in the Insertion form. (See Edit Menu - Inserting Items, Headers and Boxes in Chapter 4 of this guide for details).

Insert Box ([Query form] Edit⇨ New Box)

This command is used to insert boxes into the form and is identical in procedure to inserting boxes in the Insertion form. (See Edit Menu - Inserting Items, Headers and Boxes in Chapter 4 of this guide for details).

Lower and Upper Range Items

Lower and upper range items may be associated with a Key item or item with the range attribute set. These items are used by KE Texpress for the specification of lower and upper query bounds.

To provide for range queries on a Key item it must have both a lower and upper range item created. Standard range items need not have both lower and upper ranges specified.

Insert Lower range([Query form] Edit⇨ Lower range)

The cursor should be marking a Key item, or item which has the range attribute set. A message will display:

```
Position lower range item
```

When the user moves the cursor, the lower range item will emerge from the item. The lower range item can then be moved to the desired location. It is recommended that the prompt be changed to reflect that the item is of lower range type, e.g. "Minimum" or "From".

Insert Upper range([Query form] Edit⇨ Upper range)

The cursor should be marking a Key item or item which has the range attribute set. A message will display :

```
Position upper range item
```

When the user moves the cursor, the upper range item will emerge from the item. The upper range item can then be moved to the desired location. It is recommended that the prompt be changed to reflect that the item is of upper range type, e.g. "Maximum" or "Up to".

For easy Query form traversal, range items should be ordered in a consistent manner. By default new lower and/or upper range items immediately follow the original item in the form order. This ordering is the most convenient for data querying within KE Texpress. However, this default order need not be retained.

Move Object ([Query form] Edit⇨ Move)

This command is used to move objects around on the form and is identical in procedure to moving objects on the Insertion form. (See Edit Menu - Move Objects in Chapter 4 of this guide for details).

Copy Object ([Query form] Edit⇨ Copy)

This command is used to copy objects on the form and is identical in procedure to copying objects on the Insertion form. (See Edit Menu - Copy Objects in Chapter 4 of this guide for details).

Change Object ([Query form] Edit⇨ Change)

This command is used to change objects on the form and is identical in procedure to changing objects on the Insertion form. (See Edit Menu - Change Objects in Chapter 4 of this guide for details).

Append to Item ([Query form] Edit⇒ Append)

This command is used to append fields to items on the form and is identical in procedure to appending to items on the Insertion form. (See Edit Menu - Append to item in Chapter 4 of this guide for details).

Delete Object ([Query form] Edit⇒ Delete)

This command is used to delete objects around on the form and is identical in procedure to deleting objects on the Insertion form. (See Edit Menu - Delete Object in Chapter 4 of this guide for details).

Query Form Define menu

Edit Attributes ([Query form] Define⇒ Attributes)

In order to maintain consistency between Insertion and Query Forms, certain attributes may not be edited on Query forms. The forms editor automatically adjusts the attribute traversal path to ensure consistency.

Linked database attributes can be designated on Query forms. Also fields may have assignment, validation and branch expressions. (See the Attributes sections in Chapter 4 of this guide.)

View Hierarchies ([Query form] Define⇒ View Hierarchies)

This command displays all the set up hierarchies and their components. (See Hierarchies - View in Chapter 4 of this guide for details).

Form Size ([Query form] Define⇒ Form Size)

This command is used to view and change the bounds or size of the form. (See Form Bounds - Define in Chapter 4 of this guide for details).

Form Order ([Query form] Define⇒ Form Order)

This command is used to define the order of the items on the form, i.e. the order in which they are visited. (See Form Order - Define in Chapter 4 of this guide for details).

View Order ([Query form] Define⇒ View Order)

This command is used to view the order in which items are visited on the form. (See Form Order - View in Chapter 4 of this guide for details).

Chapter 6

Configuration & Initialisation

Overview

This chapter discusses in detail the procedure required for database configuration andinitialization.

Once the database forms are designed, it is necessary to specify the configuration of the database. This defines the internal characteristics used by KE Texpress to implement efficiently the partial match retrieval method upon which the system is based.

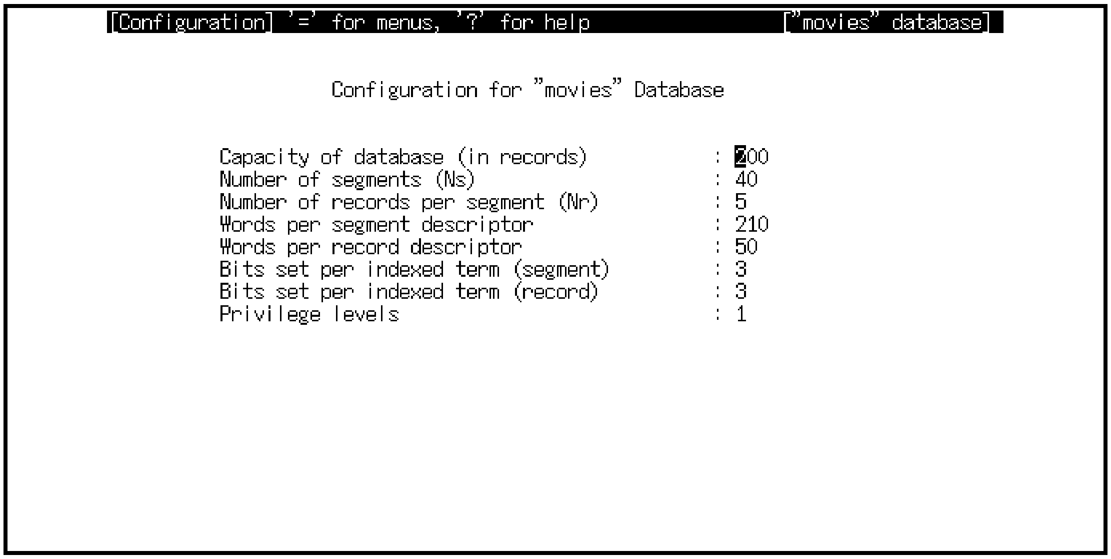
After the forms have been designed and the database configured, the database must be initialised . Initialisation simply creates the files needed by KE Texpress in order to handle the general usage of the database. The files are the Index files, Distribution file, Key files, Data files, Statistics file, and Look-up table files. Any lock files are also removed.

Note that initialising databases with existing data will delete the data. For very large databases, initialisation of the Index files can take a considerable amount of time, so theDBA should be patient.

Configure database ([Admin] Function⇒ Configure)

Once the database forms are designed, it is necessary to specify the configuration of the database. This defines the internal characteristics used by KE Texpress to implement efficiently the partial match retrieval method upon which the system is based.

Running this command displays the following parameters :



```
[Configuration] '=' for menus, '?' for help               ["movies" database]

                    Configuration for "movies" Database

          Capacity of database (in records)          : 200
          Number of segments (Ns)                    : 40
          Number of records per segment (Nr)         : 5
          Words per segment descriptor               : 210
          Words per record descriptor                : 50
          Bits set per indexed term (segment)        : 3
          Bits set per indexed term (record)         : 3
          Privilege levels                           : 1
```

Configuration parameters

The movement commands are used to step forward and backward respectively through the parameters. When paramaters are changed, the configuration needs to be saved. This in effect updates the configuration values for the database. A message will then display :

```
Re-configuring database . . .
```

Exiting returns the user to the Admin form. If exit is selected after some configuration parameters have been modified, the system will first prompt for confirmation:

```
Configuration modified. Really exit? [y/n]
```

If the configuration is changed after the database has been initialized, several courses of action are available, including automatically rebuilding KE Texpress's index or discarding all existing data. For more information, refer to Chapter 9 of this guide.

The configurationparamaters are discussed below:

Capacity of Database

This defines the maximum number of records the database can hold when the fast insertion mechanism has not been enabled. When the fast insertion mechanism is used, there is no absolute maximum capacity. However, KE Texpress is more efficient if this figure is not exceeded.

There are a few very important points which must be considered before a capacity is chosen. The speed of record retrieval is directly related to the number of records the database is designed to hold, not just the number of records actually inserted into the database. For example, a database configured to hold 100,000 records but with only 50,000 present will have similar retrieval response times to a database configured for 100,000 records and with 95,000 inserted. Therefore a database should only be configured at a value slightly above the maximum number of records expected over a reasonable period of time.

For example, consider a database with an initial size of 200,000 records. If it is expected that an additional 18,000 records (approximately) will be inserted each year then the best strategy is to configure the database initially for say 220,000 records and increase the capacity by some 20,000 each year.

In this manner the query time in general is as fast as possible relative to the actual number of records in the database. The other advantage is that disc space is not wasted allocating large KE Texpress index files, part of which are not currently being used. Increasing the capacity requires the database to be re-indexed using the automatic rebuild facilities. This is a straightforward task, although for large databases it can take a significant amount of time. (Refer to the KE Texpress Maintenance Guide for more details.)

When a value is entered for the database capacity, KE Texpress automatically calculates values for the next six configuration parameters displayed on the screen. These calculated values are near optimal for the given forms and the designated database capacity. For this reason it is important that the database forms are designed before configuration is attempted. If any data exists, then the configuration is also based on an estimate of the average number of indexed terms in each record.

It is possible to set values explicitly for the six configuration parameters below the capacity parameter. However, the DBA should be fully conversant with the method of partial match retrieval upon which the KE Texpress retrieval mechanism is based. Inappropriate settings of these parameters may well result in inadequate database performance. This practice is therefore not recommended. For a full discussion on the setting of these parameters, refer to Chapter 8 in this manual.

No. of Privilege Levels

This value indicates the number of privilege levels to be used by the database. The maximum number of privilege levels permissible is 9, ranging from 0 (highest privilege) to 8 (lowest privilege). Privilege levels are hierarchical in scope, that is, users at level 0 are privileged to access everything in the database. Users at level 1 can access everything from level 1 down to 8. Users at level 2 can access everything at level 2 down to 8, etc.

To update the number of privilege levels, a number in the range 1 to 9 should be entered. If it is desired for all users to be able to access all information then the number of privilege levels should be set to 1.

Re-Configure an Initialised Database

Altering any of the first seven configuration parameters on an initialized database makes the current index invalid. This necessitates an index restructure which is performed using the database rebuild facility. These operations are described in more detail in Chapter 9 of this guide.

Note that altering the number of privilege levels of a database does not require an index rebuild. If the number of levels is increased, then no change is required to the data or forms. If the number of levels is decreased, then records within the database and items within the form which are at or below the lowest level for the database are silently changed to the lowest level.

Initialise database ([Admin] Function⇒ Initialise)

Database initialisation is performed when the forms have been designed and the database configured. Initialisation simply creates the files needed by KE Texpress in order to handle the general usage of the database. The files are the Descriptor files, Distribution file, Key files, Data files, Statistics file, and Look-up table files. Any lock files are also removed. Note that initialising databases with entered data will lose the data. For very large databases initialisation of the Descriptor files can take a considerable amount of time, so the DBA should be patient.

There are various checks employed in the initialisation program to ensure correct usage. If the database is already initialised then a warning message will display:

```
"dbname" database has already been  initialized.
If you continue you will lose the data already entered.
Permanently remove data? [y/n].
```

Choosing no returns control to the Admin mode whereas choosing yes results in a final confirmation request :

```
Last chance before data is removed. Abort? [y/n]
```

Choosing yes returns the user to Admin mode, otherwise database initialisation proceeds and the data is lost.

If initialisation is attempted when the data is saved for automatic re-build, a message will display :

```
Data for "dbname" database is saved for an automatic rebuild.
If you continue you will lose the data already entered.
```

A similar verification procedure then follows.

Chapter 7

User Accounts

Overview

The user accounts program is used to maintain the information regarding the users permitted to access the database and the privileges of each of these users. The user accounts program allows for the insertion, modification and deletion of user accounts. It also contains a facility to display a summary of all users and their associated privileges.

Using this program, it is possible to create a controlled and secure database environment. In order to hide the existence of privileges from the user, KE Texpress does not display commands which require a privilege that has not been granted. Thus for each user, anything which appears on the screen can be used. By changing a user's privileges it is possible to alter the commands available to that user within KE Texpress.

Database Accounts

A user's account consists of his/her Unix login Id and database access privileges. Each KE Texpress database can have unlimited numbers of users and thus user accounts. A user account can be broken down into the following categories:

- User login Id
- Privilege levels
- Maximum retrieved record limit
- User privileges
- Administrator privileges
- Look-up table privileges

KE Texpress uses this information to control user access. The following sections discuss each of the above categories.

User Login Id

A user's Id is the Unix login Id used when commencing a Unix session. KE Texpress does not request a user to enter a login Id, but rather uses his/her Unix login Id. Thus a user does not require a separate database Id and password for accessing a database.

The user Id is the primary key for retrieving and inserting user accounts into the user registration file. Note that database accounts can be given to only valid Unix users, i.e. users with valid Unix accounts. Also, each user of the database can have only one entry in the user file.

Privilege Levels

Privilege levels are used to control data access at the record level and at the item level.

The privilege levels are implemented as a number between **0** (most privileged) and ***n - 1*** (least privileged) where ***n*** is the number of privilege levels specified in the database configuration (refer to the previous chapter on Configuration and Initialization). A user is assigned two distinct privilege levels:

- Display privilege level
- Insertion privilege level

Display privilege level

During the design phase of a database, each item, header and box of the Insertion and Query Forms is assigned a privilege level. When a user retrieves a record following a query, only those items, headers and boxes with a privilege level lower than or equivalent to the user's Display privilege level will display on the screen. All other items, headers and boxes will be invisible to the user.

Insertion privilege level

When a new record is inserted, it is assigned a privilege level equivalent to the user's Insertion privilege level (see the KE Texpress User Guide). When performing queries, only users with a privilege level greater than or equivalent to the record's privilege level are able to retrieve that record. In other words, for insufficiently privileged users, the record is invisible.

Only items, headers and boxes with a privilege level lower than or equivalent to the user's Insertion privilege level appear on the screen whenever a record is inserted or edited.

If the database is configured to contain only one privilege level, the Display and Insertion Privilege levels are automatically set to **0** and cannot be altered.

Maximum Retrieval Limit

To prevent a user from monopolizing the computing facilities by performing overly-general and resource-consuming queries, maximum retrieval limits can be used to limit the maximum number of matches for any query. If a user's query results in more matches than the maximum retrieval limit, the following message is displayed:

```
Too many records matching this query. Please be more specific
```

Note that the query terminates as soon as the maximum retrieval limit is reached and the viewer can either view the retrieved records (up to the limit) or perform another (more detailed) query.

The DBA can set a user's maximum retrieval limit to be any value between one and infinity. A value of **infinity** implies that there is no limit placed on the number of records that the user may retrieve. To set the retrieval limit to infinity, the DBA needs to enter**0** in this field.

Access Privileges

Access privileges define the database operations that a user is permitted to perform. If the privilege is granted, the user can use the facility. Otherwise the operation is denied to the user, who also will not be aware that the operation even exists.

The access privileges are described in detail below. Associated with each privilege is a letter which is used to identify the privilege in the summary operation described in the Editing User Accounts section in this chapter.

Record insertion (I)

Allows a user to insert data permanently into the database.

Record edit (E)

Allows a user to edit a record stored in the database.

Record deletion (D)

Allows a user to delete a record stored in the database. The operation of record deletion is permanent and cannot be undone.

Change record privilege level (P)

Allows a user to alter the privilege level of a record stored in the database. It is relevant only to databases with more than one privilege level and to users whose Insertion privilege level is not the lowest privilege level of the database.

Query (Y)

Allows a user to perform queries.

Destructive edit (U)

When a record is edited, the previous version can be saved and added to the record's history chain. If this privilege is granted, the user is given the option of saving or discarding the old version. This privilege also allows the user to delete old versions of the record.

Validation override (V)

Allows a user to override the validation for a record and therefore insert a record which is invalid.

Videodisc access (X)

Allows a user to access the videodisc whilst running KE Texpress. It is relevant only to databases with a videodisc library item and on machines which connect to a videodisc supported by KE Texpress.

Temporary record creation (T)

Allows a user to insert a record into the Temporary file. It is relevant only to databases with a Key item.

Temporary file summary (F)

Allows a user to obtain a summary of all of the records stored in the Temporary File. It is relevant only to databases with a Key item.

Temporary record retrieval (R)

Allows a user to retrieve any record from the Temporary file, regardless of who inserted the record. It is relevant only to databases with a Key item.

Report access (C)

Allows a user toutilize the reporting facilities provided by KE Texpress.

Report create / delete (K)

Enables a user to create, modify and delete Report Forms. Only Report Forms owned by the user can be edited or deleted.

Unix facilities (N)

Allows a user to use the Unix facilities provided within KE Texpress, such as listing the current directory, changing directories and executing Unix commands.

Set user options (S)

Enables a user to set options interactively using the Options menu. If this privilege is not granted then the user must use the environment to alter any KE Texpress options.

Administrator Privileges

Selected users may be permitted to perform some database maintenance operations. These operations are controlled by the Administrator privileges, which are similar in structure to the User access privileges previously described. The actual maintenance operations controlled by these privileges are described in more detail in the KE Texpress Maintenance Guide.

Users must be a member of the Unix group **Texpress**, in order to use the maintenance facilities.

The administrator privileges are as follows:

Administrator menu (M)

Allows a user to gain access to Admin mode. It provides **read only** access allowing a user to view the current design or settings but not to alter them. The following administrator privileges are dependent on the user being granted this privilege.

Open / Close database (O)

Allows a user to open and close the database. Closing a database may involve forcibly terminating other users' sessions.

Backup database (B)

Allows a user to perform database backups.

Look-up table edit (L)

Allows a user to edit directly the values stored in Look-up tables used in the database.

Audit control (A)

Allows a user to enable and disable auditing and to produce auditing reports.

Database administrator

This privilege is assigned to the DBA. It is granted when the database is created and cannot be granted to any other user. (The super user may change the DBA of a database by using the **texchdba** command. Refer to the KE Texpress Utilities Guide.)

Look-up Table Privileges

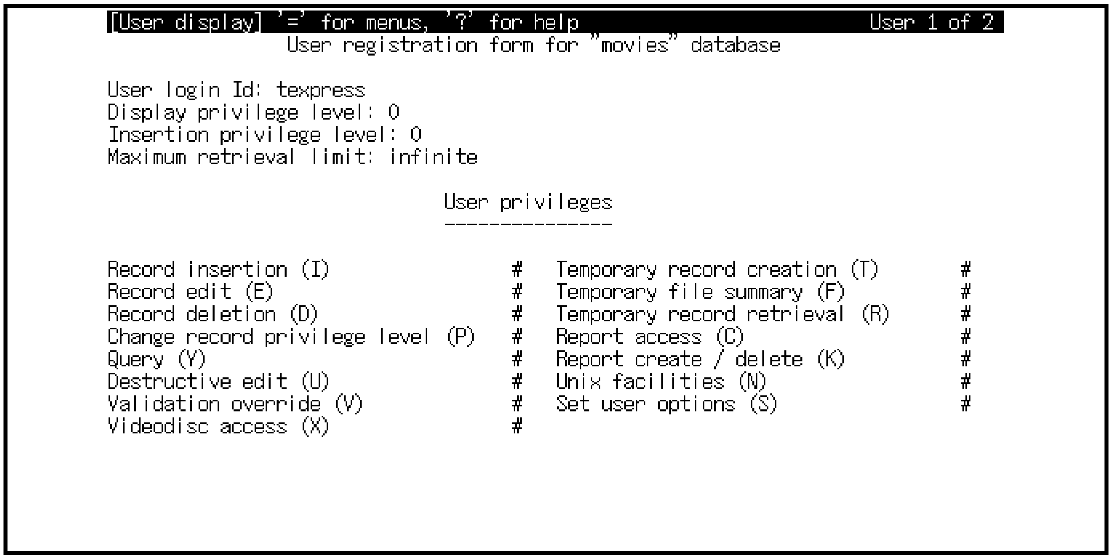
For each user, a Look-up table may be set as **read only** (i.e. insertions of new entries into the Look-up table are not allowed) or **read/write** (i.e. new values may be inserted into the Look-up table). If a Look-up table is part of a hierarchy, access rights affect the complete hierarchy. It is also possible to allow write access to individual Look-up tables for each user.

The only exception is for Look-up tables which have been designated **read-only** by the DBA using the Look-up table edit facility (refer to the KE Texpress Maintenance Guide). This read-only status applies to all users of the database and over-rides all user privileges..

User Registration Form

This form (consisting of at least 2 screens) is used throughout the program for displaying, registering, deleting and modifying user accounts. Examples of the screen images are shown in the following figures.

If the database contains Look-up tables, a third page will display and alter the Look-up table access privileges. If the list of Look-up tables cannot be displayed on a single screen, then additional pages are created to cater for the associated Look-up table privileges. An example of a Look-up page is also shown.



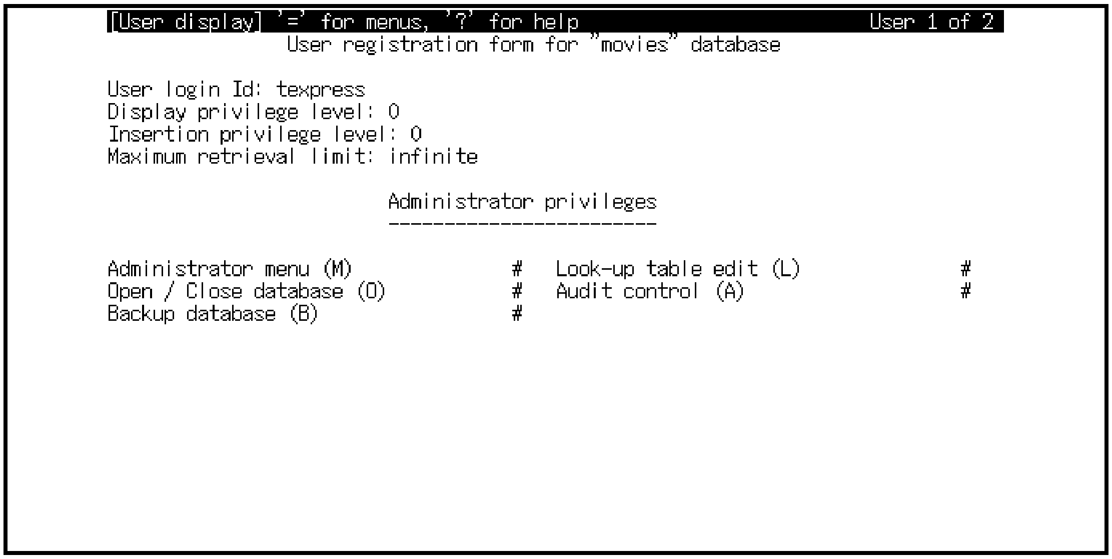
```
[User display] '=' for menus, '?' for help                    User 1 of 2
                User registration form for "movies" database

User login Id: texpress
Display privilege level: 0
Insertion privilege level: 0
Maximum retrieval limit: infinite

                              User privileges
                              ---------------

Record insertion (I)                #   Temporary record creation (T)      #
Record edit (E)                     #   Temporary file summary (F)         #
Record deletion (D)                 #   Temporary record retrieval (R)     #
Change record privilege level (P)   #   Report access (C)                  #
Query (Y)                           #   Report create / delete (K)         #
Destructive edit (U)                #   Unix facilities (N)                #
Validation override (V)             #   Set user options (S)               #
Videodisc access (X)                #
```

User privileges



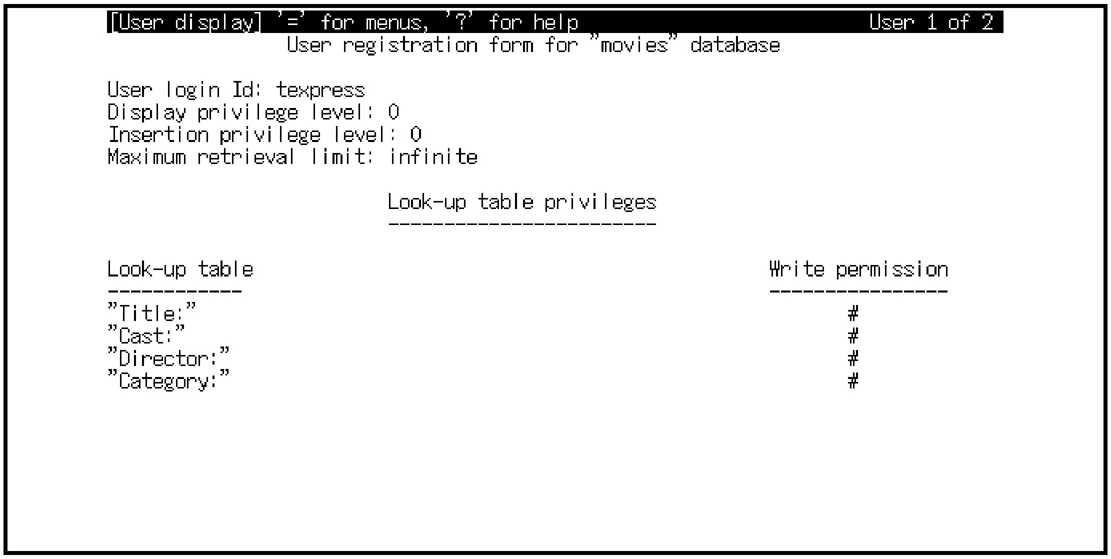
```
[User display] '=' for menus, '?' for help                    User 1 of 2
                User registration form for "movies" database

User login Id: texpress
Display privilege level: 0
Insertion privilege level: 0
Maximum retrieval limit: infinite

                         Administrator privileges
                         ------------------------

Administrator menu (M)              #   Look-up table edit (L)             #
Open / Close database (O)           #   Audit control (A)                  #
Backup database (B)                 #
```

Administrator privileges



```
[User display] '=' for menus, '?' for help                     User 1 of 2
                User registration form for "movies" database

User login Id: texpress
Display privilege level: 0
Insertion privilege level: 0
Maximum retrieval limit: infinite

                         Look-up table privileges
                         ------------------------

Look-up table                                          Write permission
------------                                           ----------------
"Title:"                                                      #
"Cast:"                                                       #
"Director:"                                                   #
"Category:"                                                   #
```

Look-up table privileges

Display Mode

When the user accounts program is invoked, the user enters a mode of operation similar to the Display mode in **texforms**. (See the KE Texpress User Guide). The access privileges of the first account are displayed. The mode is set to **User display** and in the right edge of the menu bar is a message in the format:

```
User x of y
```

where ***x*** represents the number of the current user's account (initially 1) and ***y*** is the total number of registered user accounts for the database.

The available commands are as follows:

Edit account ([User display] Function⇒ Edit)

This command allows the current user account to be edited. The form editing commands are described in the Editing User Accounts section in this chapter. At the termination of the editing, the user is returned to Display mode.

Insert account ([User display] Function⇒ Insert)

This command adds a new user account to the database. KE Texpress prompts with:

```
Enter user name:
```

and the new user's Unix Id can be entered. A new account with minimum privileges is created and the user commences editing this account as if an edit operation had commenced. Entering no name escapes the command. Only existing Unix login Id's can be entered. Also, the superuser of the system (typically **root**) cannot be given a database account as this user can bypass any locking schemes. The account insertion command is available only to the DBA.

Delete account ([User display] Function⇒ Delete)

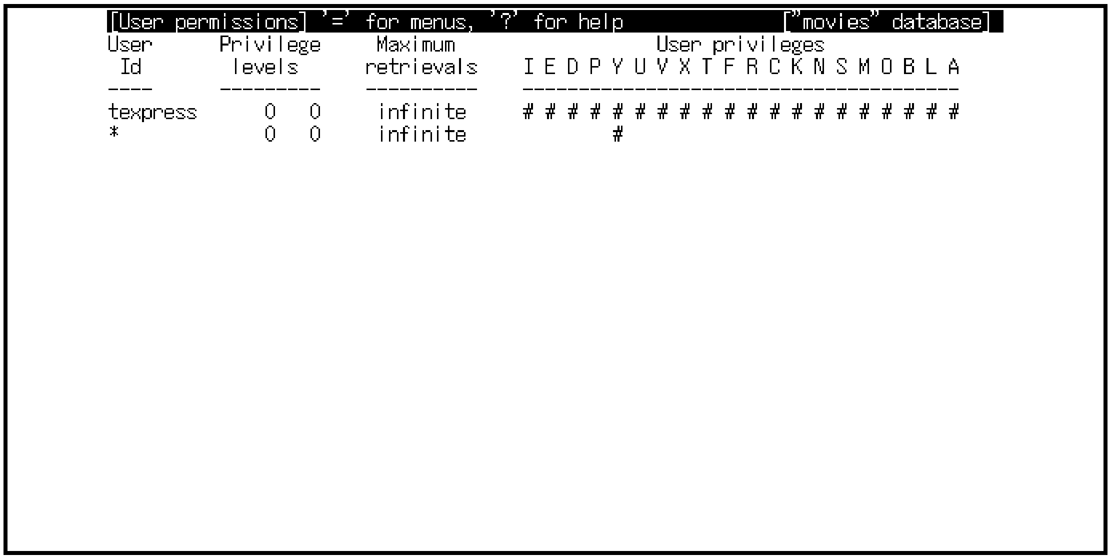
When the command is run, KE Texpress prompts for confirmation :

```
Delete "userid"? [y/n]
```

where ***userid*** is the Unix Id of the current user. Choosing yes results in the account being deleted and the user will no longer have access to the database. Choosing no aborts the operation. Note that he DBA's account cannot be deleted.

Summarise account ([User display] Function⇒ Summarise)

A summary of the user accounts for the database is displayed using in Perusal mode. This summary contains the user login Id, privilege levels, maximum retrieval limit, access and administrator privileges for all of the registered accounts. An example of the screen is given in the following figure.



```
[User permissions] '=' for menus, '?' for help              ["movies" database]
User     Privilege     Maximum                User privileges
 Id       levels     retrievals   I E D P Y U V X T F R C K N S M O B L A
----     ---------    ----------  ---------------------------------------
texpress     0   0     infinite   # # # # # # # # # # # # # # # # # # # #
*            0   0     infinite           #
```

User summary

Exit account ([User display] Function⇒ Exit)

This exit the user accounts program and the user is returned to Admin mode.

Next account ([User display] Move⇒ Forward)

The identical page of privileges for the next user account is displayed. The account after the last account is the first.

Previous account ([User display] Move⇒ Backward)

The identical page of privileges for the previous user account is displayed. The account before the first account is the last.

Next page ([User display] Move⇒ Next)

The next page of privileges for the current user account is displayed. The first page contains the access privileges. The second contains the administrator privileges and the third and subsequent pages contain the Look-up table access privileges.

Previous page ([User display] Move⇒ Previous)

The previous page of privileges for the current user account is displayed.

Editing User Accounts

When the DBA elects to insert a new account or edit an existing account, User edit mode is invoked using the user registration form. Text edit mode commands are available to change the values of the **privilege levels** and **Maximum retrieval limit**fields.

When entering a new privilege level, only a value between 0 and the lowest privilege level is accepted. All other values are rejected and an appropriate error message is displayed. Note that the field is only one character in length and accepts only a digit. It cannot be reached if the database is configured for only one privilege level (when all users must be at level 0).

The maximum retrieval limit accepts only integer values. A value of zero, when entered, is replaced by the text, **infinite** and is used to indicate that the retrieval limit for the user is infinite (limitless).

The remainder of the fields (the login Id field is not accessible) represent the privileges and can only be set or reset using the toggle command. When inserting a new account, all privileges are initially in the**un**set state (turned off).

Commands available in User edit mode are:

Toggle privilege ([User edit] Function⇒ Toggle)

Toggle the setting of the current privilege. If it is set, the character, **#**, is displayed next to the privilege.

Copy user account ([User edit] Function⇒ Copy)

Copy the privileges assigned to another registered user account to the current user account. The privileges already granted to the current user are removed. The message:

```
Enter user Id:
```

is displayed. The Unix login Id of the user account to be copied can be entered. If the Id given is that of a valid database user, the privileges associated with that user are assigned to the current user. Entering no user Id escapes the command.

Save User Account ([User edit] Function⇒ Save)

This command saves this user account. When the command is completed, the user is returned to Display mode.

Exit User Account ([User edit] Function⇒ Exit)

Return to Display mode without saving the account. If the account has been modified, KE Texpress prompts with:

```
Modified. Really exit? [y/n]
```

Choosing **no** aborts the exit and leaves the user in Edit mode.

Next Privilege Field ([User edit] Move⇒ Forward)

Move the cursor forward to the next field on the current page of the user registration form. The field after the last on this page is the first field on this page.

Previous Privilege Field ([User edit] Move⇒ Backward)

Move the cursor backward to the previous field on the current page of the user registration form. The field before the first on this page is the last field on this page.

Next page ([User edit] Move⇒ Next page)

Display the next page of the user registration form and commence editing the privileges on that page. The page after the last is the first for this user.

Previous page ([User edit] Move⇒ Previous page)

Display the previous page of the user registration form and commence editing the privileges on that page. The page before the first is the last for this user.

Global Access Account

This special account allows access to all users not individually registered to use the database. This account Id is represented by the single character *****. The set-up procedure for this account is identical to that for individual user accounts as described in the previous section.

This is an optional account and if set up, all Unix users without a specific account for the database will be assigned its privileges.

Any administrator privileges granted to this account however will apply only to users who are in the**Texpress**Unix group.

Chapter 8

Configuration Parameters

Overview

This chapter contains information which is peripheral to the normal development of a database. It describes the retrieval mechanism used by KE Texpress and methods for the selection of optimal configuration parameters. It is not essential for a DBA to read this section of the manual. However, it is supplied both for completeness and to provide an opportunity to gain a better understanding of the workings of KE Texpress.

KE Texpress is capable of selecting near optimal values for the configuration of a database. However, to give the DBA more control over these parameters without requiring an extensive knowledge of the method of storage employed, KE Texpress provides a program which may be used to suggest possible database configurations. This program is called **texconf** and can be used to explore a variety of configurations while estimating the probable cost of each.

Although it is not necessary to have an extensive knowledge of the storage method, several concepts must be understood to make best use of this program. This chapter is divided into the following sections:

(1) A brief description of the method of data storage employed by KE Texpress.

(2) A description of the use of **texconf** and of the output it produces.

(3) Advice on attempting to select optimal configuration parameters for a particular application.

The KE Texpress Storage Method

KE Texpress uses a method of storage the technical name for which is a Two Level Superimposed Coding Scheme for Partial Match Retrieval. The two levels referred to are known as the segment level and the record level. A segment can be considered to be a bin into which a series of records can be placed. There are *Ns* such segments in a database each of which can hold *Nr* records. Thus from these two parameters, the capacity of the database can be calculated by the simple expression:

```
Capacity = Ns *  Nr
```

Note that *Ns* and *Nr* correspond to the parameters, "Number of segments", and "Number of records per segment", respectively, on the Configuration menu.

If the fast insertion mechanism is not enabled on a database, then as each record is inserted, it is automatically fully indexed, i.e. the record is totally inserted. The capacity defined above then represents the absolute maximum number of records which the database can hold.

If the fast insertion mechanism is enabled, each record is only partially inserted, i.e. information providing access to the data is stored, not in the index files, which are limited in size, but in the overflow file, which has no size limit. In this case, there is actually no limit to the number of records the database can hold. Storing many partially inserted records causes a degradation in the query response time. The capacity defined above then represents the optimal capacity of the database.

Each record has an associated record descriptor, which contains sufficient encoded information to provide KE Texpress with a summary of what is contained in the record. This descriptor is in fact a bit string, the length of which is specified in logical words (32 bits). The number of words required for each record descriptor is referred to as *Br* and corresponds to the parameter, "Words per record descriptor", on the Configuration menu. An entry at the record level also contains a pointer to the data for the record. Thus the actual size of a record level entry, in words, is:

```
Entry = Br + 1
```

Similarly, each segment has an associated segment descriptor, which contains sufficient encoded information to provide a summary of what is contained in all records stored in that segment. The number of words required for each segment descriptor is referred to as *Bs* and corresponds to the parameter, "Words per segment descriptor", on the Configuration menu. Typically, this parameter is much larger than*Br* as it must encode *Nr* times as much information.

From these parameters, an estimate of the space overhead for a database can be given by the following formulae:

Segment level

$$\mathtt{Ssize = Ns * Bs * BPW}$$

Record level

$$\mathtt{Rsize = Ns * (Nr * Entry * BPW)}$$

Total

$$\mathtt{Overhead = Ssize + Rsize}$$

where **BPW** represents the number of bytes per word, i.e. 4. Here, both Ssize and Rsize are calculated in bytes. The calculation of Rsize is simplified here. Due to the efficiency of reading complete disc pages rather than reading information straddled across page boundaries, each segment is rounded up so that its size is an integral number of disc pages. Thus Rsize is more aptly represented (using integer arithmetic) as:

$$\mathtt{Rsize = Ns} * \frac{\mathtt{Nr * Entry * BPW + BPP - 1}}{\mathtt{BPP}} * \mathtt{BPP}$$

where BPP represents the number of bytes per page. This figure is dependent on the natural page size supported by the machine but is generally 512 or 1024.

The space allocated to the record descriptors, which is used purely to round up the size of each segment, is wasted. The total amount of wasted space is in fact Ns times this rounding amount (this amount is wasted for every segment) and is referred to as *Waste*.

Notice that the calculation of the space overhead can be deceptive. KE Texpress does not store blank padded records. Only the characters entered as part of the record description are stored in the data file (plus, at most, one delimiter for each field and an overhead of between 2 and 10 characters per record). Many other systems reserve the maximum space allowed per record for every record in the database. Thus for a database with the occasional verbose record but with mainly sparsely filled records, the descriptor overhead can be more than absorbed by the data file compaction.

Finally, the formation of the descriptors involves using each indexed term of a record to contribute some encoded information to the descriptor or bit string. Initially, each bit string is set to zeroes. Each indexed term is converted, using a technique known as hashing, to a sequence of bit positions in the appropriate descriptor. These bit positions are then set to '1'. The number of bit positions set by this operation in the segment descriptor is called Ks and corresponds to the parameter, "Bits set per indexed term (segment)", of the Configuration menu. The number of bit positions set in the record descriptor is called *Kr* and corresponds to the parameter, "Bits set per indexed term (record)", of the Configuration menu.

If two terms are indexed into the one descriptor, (say, at the record level), at most

```
    2 * Kr
```

bits are set to '1'. However, hashing produces near-random numbers and so it is possible that each indexed term may result in the same bit being set and in fact perhaps only

```
    2 * Kr - 1
```

bits actually become set in the descriptor. Thus if the descriptor size (*Br*) is too small, or the number of bits set per indexed term (*Kr*) is too large, or the number of indexed terms per record (**Indexed_terms**) is too high, then more bits are found to "clash" in this way.

The result of "clashing" bits in a descriptor is that there arises the possibility of the descriptor of a query matching that of a record when, in actual fact, the information in the record does not match that of the query. Such a record is termed a false match.

To simplify this discussion consider the following example (which considers only the record level). Let the record descriptor be a 128 bit long string where the bits are numbered from 0 to 127. Assume that there exists a function, **hash**, which can convert an indexed term into a set of three numbers between 0 and 127 inclusively (i.e. *Kr* is 3). Assume also that for the indexed terms, **large**, **brick**, **house** and **igloo**:

```
    hash(large)    =      {5,     39,     102}
    hash(brick)    =      {27,    42,     48}
    hash(house)    =      {39,    82,     120}
    hash(igloo)    =      {5,     42,     120}
```

If a record containing the three indexed terms, **large**, **brick** and **house**, is inserted into the database, its record descriptor would have bits:

```
    5    27    39    42    48    82    102   120
```

set. Bit 39 is a clashing bit within this record.

If the indexed term,**house**, is specified in a query then all records with the bits:

```
    39    82    120
```

set in their descriptors would be found as possible matches. Thus the record described above would be found.

From this example, however, it can be seen that a query specifying the indexed term, **igloo**, would be matched by a record containing the indexed terms, **large**, **brick** and **house**. This is because each of the bits set by the indexed term, **igloo**, "clashes" with a bit from each of the other indexed terms. This same situation can arise at the segment level.

The probability of a false match occurring (at either the segment or record level) is thus dependent on the size of the descriptor, the number of bits set per indexed term and the number of indexed terms. As these are controllable parameters, however, it is possible first to set the false match probability and then to calculate appropriate values for the configuration parameters. Thus the following variables have been defined:

Sfalse The false match probability at the segment level is such that one false segment match is expected for every

```
Sfalse * Ns
```

matching segments. (A false match at the segment level implies that a segment containing no matching records was searched unnecessarily. Thus the query was less efficient.)

Rfalse The false match probability at the record level is such that one false record match is expected for every

```
Rfalse * Nr
```

matching records.

The final variable to be considered in this discussion is the length of the tag field (**Taglen**). At the record level, a section of the record descriptor is reserved for the encoding of the privilege level of the record and the range into which each ranged field value of the record fell. The size of this tag field is determined by allowing eight bits for the encoding of the privilege level and two bits for each range value of each ranged item in the Insertion form.

Given values of Ns,Nr, Sfalse andRfalse, it can be shown that the equations:

$$\mathtt{Ks} = \frac{\mathtt{ln(Sfalse * Ns)}}{\mathtt{ln(2)}}$$

$$\mathtt{Bsbits} = \frac{\mathtt{Nr * Indexed\_terms * Ks}}{\mathtt{ln(2)}}$$

$$\mathtt{Kr} = \frac{\mathtt{ln(Rfalse * Nr)}}{\mathtt{ln(2)}}$$

$$\mathtt{Brbits} = \frac{\mathtt{Indexed\_terms * Kr}}{\mathtt{ln(2)}} + \mathtt{Taglen}$$

can be used to select appropriate values for the remainder of the configuration parameters. Note that *Bsbits* corresponds to the number of bits required for the segment descriptor. It can be converted to *Bs* by dividing by the word size of 32 (rounding up if required). Similarly, *Brbits* corresponds to the number of bits required for the record descriptor. It can be converted to *Br* in the same way.

The variables for which values have been assumed here include *Ns*, *Nr*, *Indexed_terms* and *Taglen*. The values for these variables which result in optimal database performance are heavily dependent on the format of the database's Insertion form, the format of the data and the type of queries to be performed. KE Texpress can make educated estimates for these values based on the design of the database and any existing data and thus can achieve near-optimal values for the configuration parameters. Only a person with an excellent knowledge of the database usage plus a reasonable knowledge of the methods employed by KE Texpress is expected to be able to improve on these values. Care should be taken when attempting to alter the default values produced by KE Texpress, as these parameters are sensitive to small changes and badly chosen values can result in a grossly inefficient database.

Finally, the values, *Sfalse* and *Rfalse*, ifp2s10

set extremely large, result in an efficient database with very few false matches. However, such values require large descriptor sizes and thus result in much greater space overhead.

A Program to Suggest Configurations

The program to suggest configurations is called **texconf** and in the simplest case is invoked from the shell by the command line:

```
texconf numrecs
```

where ***numrecs*** is the minimum allowable capacity for the database. This results in output of the form:

```
Capacity of database (in records):        1000

Number of segments (Ns):                  40
Number of records per segment (Nr):       25
Words per segment descriptor:             101
Words per record descriptor:              9
Bits set per indexed term (segment):      3
Bits set per indexed term (record):       4
False match probability (segment):        0.006250     [1 / (4*Ns)]
False match probability (record):         0.001250     [1 / (32*Nr)]
Expected number of indexed terms:         10
Record descriptor tag length (bits):      108
Segment size:                             1024
Segment descriptor file size:             15.78K
Record descriptor file size:              40.00K
Percent of record descriptor file wasted: 2.34%
Total descriptor file overhead:           55.78K
```

The first seven parameters are as defined in the previous section and correspond to the parameters listed on the Configuration menu. Both false match probabilities are listed together with the expected number of indexed terms per record. This last figure is generally estimated as the average number per record plus four times the square root of this number. Assuming a Poisson distribution of terms per record for a large number of records, this ensures that more than 97% of records will contain no more than *Indexed_terms* terms.

Notice that the values for descriptor file overheads have a single character appended to them. This character can be one of:

b The size is given in bytes.

K The size is given in kilobytes (1024 bytes).

M The size is given in megabytes (1048576 bytes).

The **texconf** program, if given no other information, assumes default values for the unknown parameters. As can be seen from the output shown above, *Sfalse* assumes the value 4, *Rfalse* assumes the value 32, *Indexed_terms* assumes the value 10 and *Taglen* assumes the value 108. These default values can be altered, however, using additional command line options. The program can be invoked by a command line resembling:

```
texconf option option ... numrecs
```

where each option is one of the following:

-s*n* Set the value of *Sfalse* to ***n***. This is used to adjust the false match probability at the segment level.

-r*n* Set the value of *Rfalse* to ***n***. This is used to adjust the false match probability at the record level.

-i*n* Set the value of *Indexed_terms* to ***n***. This is used to alter the average number of indexed terms expected per record.

-t*n* Set the value of *Taglen* to ***n***. This is used to alter the length of the tag field in the record descriptor.

-d*dbname*

Use information from the database, ***dbname***, to determine the values of *Indexed_terms* and *Taglen*. Only the DBA of this database is permitted to use this option. The value of *Taglen* can be calculated exactly. However, the value of *Indexed_terms* can only be estimated. If data exists in the database already, then the average number of indexed terms can be extracted from the statistics. This figure is calculated using the design of the form as at the time of the insertion of the record and can be misleading if the design has been altered to index items which were not previously indexed.

If no data exists then the estimate is achieved by assuming one indexed term per string, integer or real field of each indexed item in the form. For indexed items with text fields, each 10 characters, or part thereof, of the first field are assumed to form one indexed term. For the second field, the quantity used is 15 characters and for the third, 20 characters, and so forth.

-n*n* This option is applicable only if the **-d** option has been used. It results in up to ***n*** records of the database being read to determine the expected number of indexed terms per record. If ***n*** is omitted, then all records are read.

The **texconf** program uses these values in the calculation of possible configurations. The output produced reflects the new values chosen. If conflicting arguments are given (such as for indexed terms, **-d** and **-a**) the last of these takes precedence.

The automatic configuration option on the Function Menu in Admin mode uses this same approach to select a configuration.

Select an Optimal Configuration

When configuring a database, optimization is somewhat arbitrary and effectively applies to only one particular set of queries. One of the most important design considerations of KE Texpress has been speed of information retrieval. Hence it probably makes sense to attempt to optimize with respect to the time taken to perform a query. The particular set of queries to be considered should be the most common or the "average" queries.

To optimize with respect to the time taken to perform a query, one should consider the slowest of all computer operations - disc accesses. If the number of disc accesses is minimized for the average query then in general the database should perform better. To achieve this, the expected number of disc accesses per query must be expressed in terms of one variable. Given that *Indexed_terms*, *Taglen*, *Sfalse* and *Rfalse* can be set for a particular application (depending on the Insertion form, the data and the storage space available), then the only variables are *Ns* and *Nr*. As *Ns* can also be expressed in terms of *Nr* by:

$$Ns = \frac{numrecs}{Nr}$$

then *Nr* is the variable with respect to which disc accesses should be minimized.

An estimate of the expected number of disc accesses per query can be given by:

$$Disc\ Accesses = Ks + f(Nr) * PPS$$

where *PPS* is the number of pages per segment. Here *Ks* is calculated as before and represents the cost, in disc accesses, of searching at the segment level. The cost of searching a single segment is *PPS* and is calculated as:

$$\frac{Nr * Entry * BPW}{BPP}$$

where this is rounded up if required. Both *Ks* and *PPS* can be expressed in terms of *Nr*.

The final unknown quantity in the expression is *f(Nr)*, the expected number of segment matches per query. Unfortunately, this quantity is dependent, not only on the choice of *Nr*, but also on the data in the database and the types of queries to be performed. For this reason, *f(Nr)* is an unknown function in the general case. Hence, KE Texpress can go no further towards providing optimal configurations.

However, for a particular application and over an extended period of time, a set of data points can be built up such that this function can be approximated. From the statistics available within KE Texpress, the average number of segment matches per query can be found. This value represents one data point and corresponds to the value of*Nr* in the current configuration.

To obtain other data points, the configuration should be altered (by selecting a new value for *Nr* and calculating appropriate values for the remaining parameters). The database should be used for a period of time and the average number of segment matches per query observed again. This process should continue until the function, *f(Nr)*, can be approximated, using a standard function approximation technique. This function can then be substituted back into the above equation for the number of disc accesses per query. Finally this equation should be minimized with respect to *Nr*. The value of *Nr* which achieves this can then be used to calculate the remaining configuration parameters. The resulting configuration is then optimal for the database and the query distribution as they exist at the time these calculations are made.

Index Analysis Programs

In some cases, the default configuration parameters calculated by KE Texpress are too conservative, relative to the actual data being stored in a particular database. Cases can occur where the configuration provides very good query response times, but allocates an unnecessarily large amount of disk space to the index files.

Alternatively, there may be cases where the index files are too small for the application, the result being an excessive number of false matches and reduced query performance.

Two programs, **texanalyse** and **texdensity** may be used to help determine a configuration closer to optimal than the default configuration for a particular application. It is expected that most administrators will never need to use these programs.

texanalyse

The program **texanalyse** may be used to analyse the data in a database. The output from **texanalyse** may be used to suggest options to be used with **texconf**, to determine an appropriate configuration for the database under analysis. The command is used as follows:

```
texanalyse  [-nnum]  [-s]  dbname
```

The command reads records from the database, ***dbname***, and produces a table which is a summary of the number of indexed terms in each record.

The options are as follows:

-n*num* Read at most num records. Otherwise, by default, all the records in the database are read

-s Print an analysis of the record size in bytes. This part of the output is another table, containing a summary of the record size.

Examination of the output from **texanalyse** yields the maximum number of indexed terms encountered in any record in the database. This number may be considerably lower than that predicted by **texconf** as the expected number of indexed terms, based on the design of the Insertion form. The default configuration may, in such a case, cater for index files much larger than necessary. The **texconf** program may then be used with the **-i** option to predict an appropriate configuration, using the maximum number of indexed terms encountered in the analysis.

texdensity

This program analyses the density of set bits in the descriptors of a database. The command is used as follows:

```
    texdensity  [-nnum] dbname
```

The program reads bit-slices from the segment file of the database, ***dbname***, and produces as output a table showing a summary of the number of bits set in each descriptor, together with the average and maximum bit densities of the record descriptors. The option is used as follows:

-n*num* Read only the first***num*** bit slices from the segment file

If the density of set bits is excessive, the probability of false matches may be unacceptably high, and reconfiguration to obtain less dense descriptors may be advisable.

Chapter 9

Database Modifications

Overview

KE Texpress allows the DBA to modify the design of a database at any time while maintaining consistency with any existing data. For minor modifications (e.g. the addition of new items or adjusting the form layout), all that is required is a change to the Insertion and/or Query Forms via **texdesign**. Some modifications however necessitate a restructure of the database. Such changes include increasing the capacity of the database, adding a Look-up table to an existing item or indexing a previously non-indexed item. Both **texdesign** and the configuration facility automatically detect when a database restructure (or reindex) is required and notify the DBA. If a reindex becomes necessary due to some design modification, the database cannot be used until the process is complete.

This chapter describes the facilities for the alteration of an initialized database and the automatic reindexing of the database. A facility which monitors the status of the reindex is also described. Due to the significant amount of time required to reindex a large database and due to the possibility of a system crash occuring during the reindex, a facility enabling the recovery of an incomplete reindex is also provided.

Design Modifications

Database design modifications can be made at any time using **texdesign**. This facility can be selected from the Function menu of Admin. If the database has been initialized, a message displays when texdesign is called up:

```
[Insertion form] [Init]. Restrict editing? [yes][no]
```

This indicates that the database has been *initialized*. Thus changes to the Insertion form may require a reindex of the database. As such a reindex will be time-consuming for very large databases, texdesign provides a **restricted** mode in which changes necessitating a rebuild are prevented. This mode can be selected by choosing yes to the above prompt.

Unrestricted mode may be selected by choosing no to the above question. In this mode, any design changes can be made. If necessary, KE Texpress will restructure the database so that existing data can be interpreted according to the modified design.

Caution must be used when deleting items, however, as KE Texpress ensures that such items can never be accessed unless a recent backup of the database is restored. (refer to the KE Texpress Maintenance Guide).

Texdesign does not make changes permanent until the form is saved. Is is possible to abort changes made since the last time the form was saved, by exiting the form without saving it. This results in a message of the form:

```
[Insertion form] [Init] [Modified]. Really exit?
[yes] [no]
```

indicating that the Insertion form has been modified since the last write to file. If the text, **[Auto]**, appears in the message then changes necessitating a reindex have previously been performed. Choosing yes results in the changes being discarded. Choosing no results in a resumption of form editing.

If changes necessitating a reindex have been made and an attempt is made to save the form, then several warnings are issued before the changes are made permanent. These warnings allow the user to abort the changes should the ensuing reindex not be desired. The warnings are as follows:

```
Major changes to form. Continue? [yes] [no]
```

indicating that either a reindex or database re-initialization will be required if the form is written. Choosing no here escapes the operation. Choosing yes results in the following message:

```
Automatic rebuild is required. Prepare for rebuild? [yes]
[no]
```

Choosing yes response results in the form being written. If changes to indexed items have been made, then the following message is also displayed:

```
Configuration must be performed before rebuilding
```

In this situation the database must be reconfigured before the rebuild is initiated.

Choosing no response to the above question indicates to KE Texpress that the database is to be re-initialized. Any existing records are discarded. However, before KE Texpress takes this step, several more warnings are issued. These are as follows:

```
Query and/or Report Forms will be removed. Continue? [yes]
[no]
```

Choosing no returns the user to forms editing. Choosing yes results in the following warning being displayed:

```
Permanently remove data? [yes] [no]
```

Choosing no returns the user to forms editing while choosing yes response results in the last warning being displayed:

```
Last chance before data is removed. Abort? [yes] [no]
```

This time a response of yes returns the user to forms editing. A response of no to removes the data so that re-initialization can be performed. Note that the last promptrequires a different response form the others. This is to prevent users from holding their finger on the **yes** key andaccidently removing all of their data.

If the database is to be re-initialized, the user must proceed to database configuration and then initialization before the database can be used. These functions are as described in Chapter 6 of this guide.

If the user commences texdesign in a database which has a rebuild pending, then a message will display:

```
Database requires an automatic rebuild
```

No further messages are issued when the form is written.

Configuration Modifications

Database reconfiguration may be necessitated by design changes (as described above) or by the need to increase the capacity of the database or the desire to select a better configuration. In this last case, as KE Texpress bases its configuration on the existing data, then if the characteristics of the data should change over time (for example, if the average number of indexed terms per record should increase), the configuration program should be run again, so KE Texpress can calculate a better configuration.

The reconfiguration process is described in Chapter 6. If the configuration is not required because of design changes, then when the new configuration is saved, a series of warnings similar to those issued for design changes is displayed before the action is performed. These warnings are described below.

Altering any of the first seven configuration parameters on an initialized database makes the current index invalid. If the *save* command is selected and it detects that this situation has occurred then the question:

```
Changes require index rebuild. Continue? [yes] [no]
```

is asked. Choosing no aborts the save and returns the DBA to the configuration screen, (exiting from which leaves all parameters unchanged). If yes is choosen then a prompt displays:

```
Prepare database for an automaticreindex? [yes] [no]
```

If yes is selected, then the database becomes unusable until its index is rebuilt. If no is selected, a prompt displays:

```
Permanently remove data? [yes] [no]
```

is displayed. Choosing no aborts the save and returns the DBA to the configuration. If yes ischoosen, a prompt displays:

```
Last chance before data is removed. Abort? [yes] [no]
```

If yes is choosen, the DBA returns to the configuration screen, whereas choosing no removes all existing data and the new configuration is saved.

Automatic Database Rebuild

If a database reindex is required (due to design or configuration changes), the database cannot be used until the reindex has been performed. The reindex program can be invoked by selecting the *Rebuild/load/progress* option from the Maintenance pull-down menu in Admin mode. Alternatively it can be directly invoked by typing **`Ctrl+A`**.

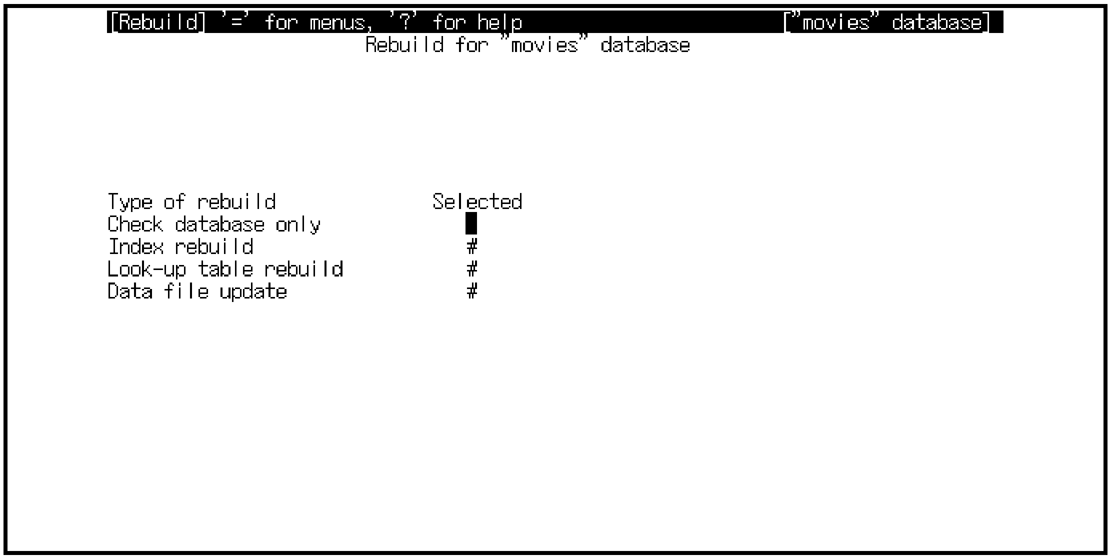
The reindex facility can perform other operations such as re-evaluating assignment expressions in all existing records. Thus it may be desirable to reindex a database even when not absolutely necessary. In this case, a message displays:

```
It is not necessary to perform a rebuild.
However, you may continue if you wish.
Do you wish to continue? [yes] [no]
```

Choosing no aborts the rebuild operation and returns the user control to the Admin screen. Choosing yes enables a database reindex to be performed, even though it has been indicated that it is not strictly necessary.

Database Rebuild Options

The menu for rebuilding the database is displayed in the following. This display may vary depending on the type of rebuild selected.



```
[Rebuild] '=' for menus, '?' for help                      ["movies" database]
                        Rebuild for "movies" database





Type of rebuild              Selected
Check database only             █
Index rebuild                   #
Look-up table rebuild           #
Data file update                #
```

Database rebuild options

KE Texpress automatically determines the type or types of database rebuild required and selects them. For example, if a Look-up table is added to an existing item, the Look-up table Rebuild option is set. So if a rebuild is required, all the necessary options will be automatically selected and there is no need to change the selection.

The available commands are as follows:

Next option ([Rebuild] Function⇒ Forward)

Move to the cursor forward to the next valid option. This may not necessarily be the next option on the screen as the cursor skips options which are automatically set by KE Texpress. This movement wraps around from the last option to the first.

Previous option ([Rebuild] Function⇒ Backward)

Move to the previous valid option. This movement wraps around from the first option to the last.

Set/Reset ([Rebuild] Function⇒ Toggle)

Toggle the current option between its set and reset states. All options which can be reached can be toggled. The cursor automatically moves to the next valid option.

List all available sub-options ([Rebuild] Function⇒ Expand)

Expands the options available for certain rebuild types. Both the Look-up table rebuild and the Data file update types can be selected by this command. This expansion of these options is discussed below.

Perform database rebuild ([Rebuild] Function⇒ Perform)

When all of the required options are selected, this command is used to signify that data rebuild may commence. Since the required options are automatically set by KE Texpress, this command is normally entered as the first and only command.

Exit Rebuild ([Rebuild] Function⇒ Exit)

Exit and return to Admin without performing the rebuild (i.e. a database rebuild may still be required before the database can be used).

Check Database Only

This option is used to check all records to ensure that they are valid. All assignment expressions are calculated, linked database entries are re-extracted and automatic items updated. A copy of each invalid record is written to an error file which is displayed upon completion of the rebuild.

This option allows full checking of all records when validation or assignment expressions are modified on the Insertion form. It also allows the DBA to find all invalid records in the database. Checking the data does not result in the modification of data already stored in the database. If this option is selected, no other option can be selected.

When the perform command is selected, a message display:

```
Commencing rebuild . . .
```

The progress can be monitored via the Rebuild Monitor facility. (See the Rebuild Progress Report section in this chapter).

Index File Rebuild

The index files contain an encoded description of the indexed and ranged data of each record in the database. They are used by KE Texpress to provide very fast retrieval of information from the database. Certain changes to the database structure, such as an alteration to the configuration or to the indexing characteristics of items in the Insertion form, can result in the index becoming out of date with the data and the design. In such circumstances, the index must be rebuilt. KE Texpress automatically determines when this is required.

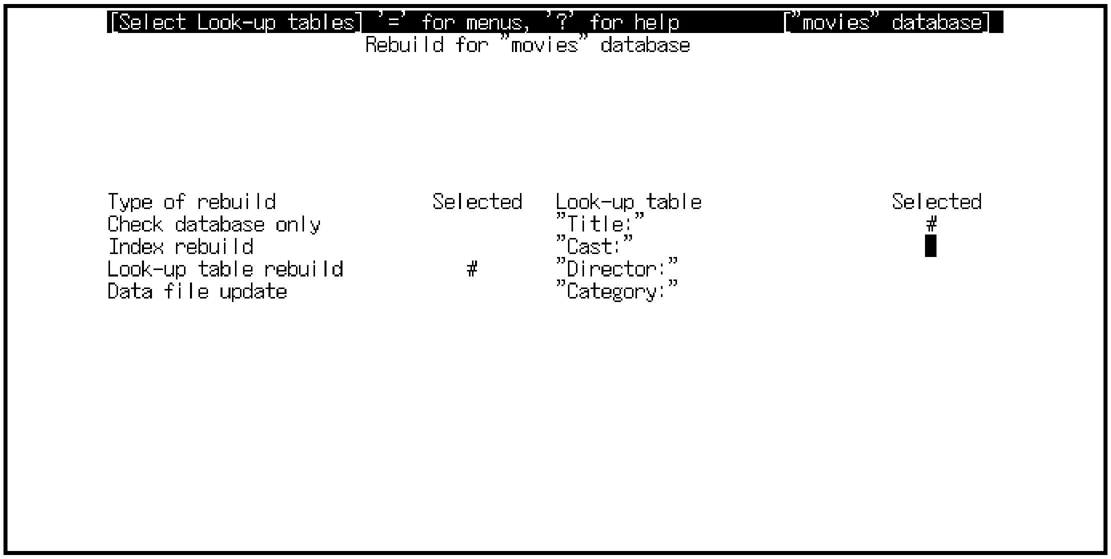
When index files are rebuilt, the database statistics are reset. This is because the statistical information is relevant only to the current configuration. All information regarding the query distribution is also reset.

Look-up Table Rebuild

A Look-up table rebuild must be performed when a Look-up table is added to an existing item, i.e. its Look-up table attribute is set or the characteristics of a hierarchy are altered. The Look-up table rebuild generates the Look-up table or hierarchy from the data already stored in the database. If a user specified Look-up table is being rebuilt, the order of the Look-up table values will be modified. The new order is based on the order in which records are scanned and new values detected by the rebuild.

It is possible for Look-up tables to become out of date with respect to the data. If a record is edited and an item with a Look-up table is modified, upon saving the record, no check is performed to determine if the old Look-up table entry has become redundant. Therefore after extensive record editing, Look-up tables may contain entries which are no longer required. A Look-up table Rebuild can be performed to discard these redundant entries.

A facility is provided to rebuild individual Look-up tables need to be rebuilt. To invoke the selection phase, the Expand command **F8**, **Ctrl+C** or **([Rebuild] Function ⇒ Expand)** is run when the cursor is located at the Look-up table rebuild option. A list of all the Look-up tables in the database displays, as shown in the following figure:



```
[Select Look-up tables] '=' for menus, '?' for help          ["movies" database]
                         Rebuild for "movies" database




Type of rebuild            Selected   Look-up table            Selected
Check database only                   "Title:"                    #
Index rebuild                         "Cast:"                     █
Look-up table rebuild         #       "Director:"
Data file update                      "Category:"
```

Look-up table rebuild selection

For the purposes of rebuilding, a hierarchy is considered as a single Look-up table and is represented by the prompt of the first item in the hierarchy. The suffix, **(Hier)**, is added to the prompt. All read only Look-up tables have **read only** placed in the **Selected** column and cannot be rebuilt. The cursor is placed on the select field of the first Look-up table which can be rebuilt.

If all Look-up tables are read only, the following message is displayed in the message window:

```
There are no Look-up tables to rebuild
```

Only those Look-up tables needing a rebuild are toggled (turned on). If the look-up table Rebuild option is toggled on manually, all look-up tables become set for rebuilding.

The commands available are:

Next look-up table ([Select look-up tables] Function⇒ Forward)

Move forward to the next Look-up table which may be selected. Read only fields are skipped. Moving off the end of the summary causes the next page of Look-up tables to be displayed. If the cursor is positioned on the last table the terminal bell sounds.

Previous look-up table ([Select look-up tables] Function⇒Backward)

Move backwards to the previous Look-up table which may be selected. Read only fields are skipped. Moving off the start of the summary causes the previous page of Look-up tables to be displayed. If the cursor is positioned on the first table the terminal bell sounds.

Next page ([Select look-up tables] Function⇒ Next page)

Move to the next page of Look-up tables. This command is available only if more than one page of tables exists. If the cursor is positioned on the last page the terminal bell sound.

Previous page ([Select look-up tables] Function⇒ Previous page)

Move to the previous page of Look-up tables. This command is available only if more than one page of tables exists. If the cursor is positioned on the first page the terminal bell sounds.

Toggle ([Select look-up tables] Function⇒ Toggle)

This command is used to toggle the rebuilding of individual tables. A # in the select field indicates that the table is to be rebuilt. The cursor automatically moves to the next Look-up table.

Exit ([Select look-up tables] Function⇒ Exit)

End Look-up table selection and return to the database rebuild type screen. The list of Look-up tables is removed from the screen.

Data File Update

Insertion form changes such as adding or changing assignment or validation expressions may cause existing records to be out of date with to the current database design. This may also occur if data in a linked database (with write links) is modified. If these changes are to be reflected in existing records, a data file update must be performed.

A data file update is the only type of rebuild which modifies the data file. For this reason, the database must be backed up before the update is attempted. To enforce this, the Data File Update option may be selected only after the database has been backed up. If the database is not backed up, a message displays:

```
The database must be backed up before updating
```

A data file update is actually a re-insertion of all record data. The DBA may select which sections of the data are to be updated. The available updates are as follow:

- Linked updates.
- Update automatic items.
- Perform validation.
- Calculate assignment expressions.

The **Linked database updates** has two variations. The first is a complete linked update which updates all linked fields. If there is no entry in the linked database for the linked key then any data in the linked fields is wiped out. The second type is a selective linked update which updates the linked fields only if the linked key exists.

Update automatic items simply updates the values in automatic fields. For example, if an automatic date item is on the form the date is updated to the current date.

If the validation on a field is modified it is possible that some of the records in the database may become invalid. The **Perform validation** update checks each record for validity. Any record which fails validation is inserted into the temporary file if the database has a Key item, or in the error file otherwise.

When an assignment expression is modified, values calculated via this expression are not updated in existing records. The **Calculate assignment expressions** update calculates all expressions and enters into each record the new values.

A facility is provided to allow for the individual selection of the type of update required. To select this facility, the Expand command **`Ctrl+C`** is selected when the cursor is situated at the Data file update option.The available commands are as follows:

Next option ([Select update] Function⇒ Forward)

Move to the next valid selection. If a full linked update is set, a partial linked update cannot be set until the former is reset and vice-versa. If the cursor is positioned on the last entry the terminal bell sounds.

Previous option ([Select update] Function⇒ Backward)

Move to the previous valid selection. If the cursor is positioned on the first entry the terminal bell sounds.

Set/Reset ([Select update] Function⇒ Toggle)

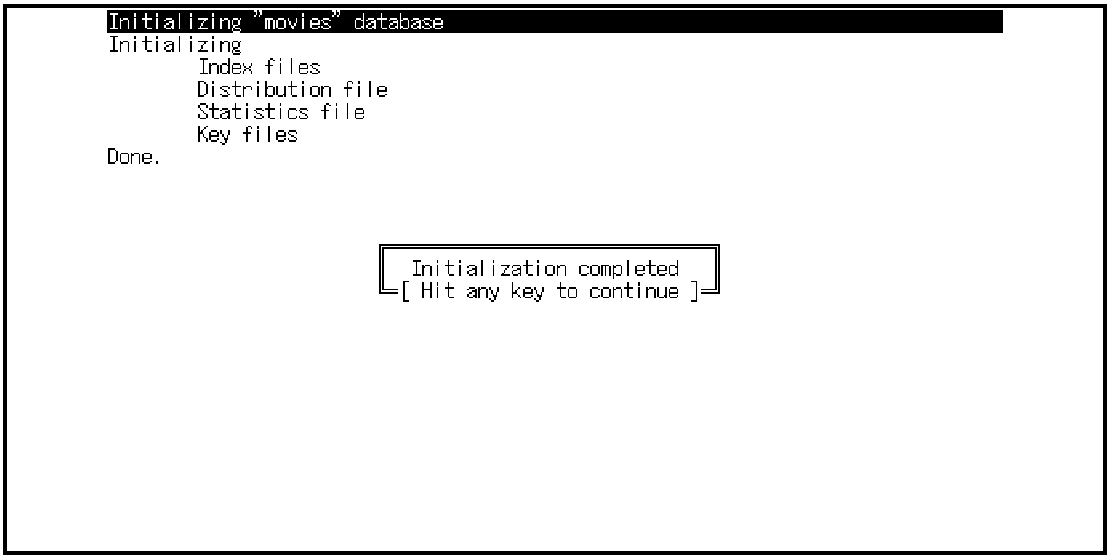
Toggle the current selection. This command is used to select the required update. If the update is selected, a # appears to the right of the update type. The cursor automatically moves to the next valid type.

Exit update selection ([Select update] Function⇒ Exit)

End update selection and return to the rebuild type selection screen. The Data file update menu is removed from the screen.

Initialization and Rebuild

After the database rebuild is performed **`([Rebuild] Function`** ⇒ **`Perform)`**, the initialization phase begins. The following figure shows an example of theinitialization sequence.



```
Initializing ”movies” database
Initializing
        Index files
        Distribution file
        Statistics file
        Key files
Done.

                          Initialization completed
                        [ Hit any key to continue ]
```

Database re-initialisation example

When the sequence has completed, a message displays:

```
Initialization completed
```

followed by:

```
Commencing rebuild . . .
```

Monitoring of the rebuild automatically commences (refer to the next section, Rebuild Progress Report).

Rebuild Progress Report

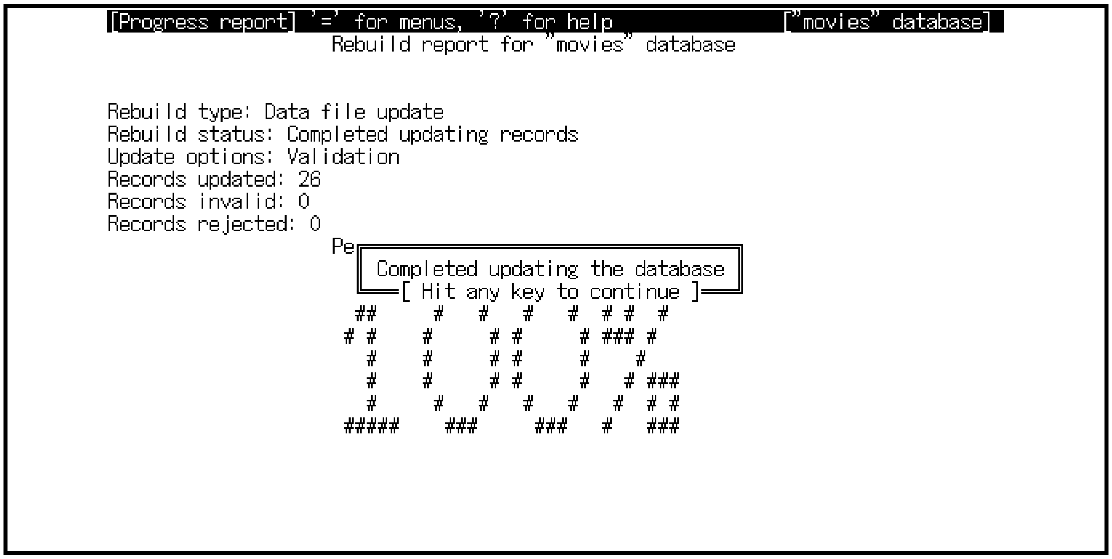
As a rebuild can take a considerable amount of time for large databases, it is performed in background. A facility exists to monitor the progress of the rebuild. This facility is invoked automatically when the rebuild commences but can be terminated at any time so that the terminal can be used for other operations while the rebuild continues. Monitoring can be resumed at any time by selecting the Rebuild/load/progress option from the Maintenance menu of Admin.

This progress report facility must be used at or after the completion of the rebuild so that the error report can be displayed. Note: The database cannot be used until this report has been viewed.

The rebuild monitoring facility provides information in the following areas:

- Rebuild type.
- Rebuild status.
- Number of records re-entered into the database.
- Number of errors (records rejected).

The type displays the rebuild being performed (for example, an Index file rebuild). The status indicates whether the rebuild is proceeding or if some sort of problem has been encountered. The remaining areas give an indication of progress of the rebuild. A sample report is shown in the following figure.



```
[Progress report] '=' for menus, '?' for help               ["movies" database]
                    Rebuild report for "movies" database


Rebuild type: Data file update
Rebuild status: Completed updating records
Update options: Validation
Records updated: 26
Records invalid: 0
Records rejected: 0
                    Pe╔═════════════════════════════════╗
                      ║ Completed updating the database ║
                      ╚═══[ Hit any key to continue ]═══╝
                      ##     #   #   #   #  # #  #
                     # #    #     # #     # ### #
                       #    #     # #     #    #
                       #    #     # #     #   # ###
                       #     #   #   #   #   #  # #
                     #####    ###     ###   #   ###
```

Rebuild progress report

The available commands are as follow:

View errors (Function⇒ View errors)

View the current error file.

Update frequency (Function⇒ Update time)

Change the frequency at which the report screen is updated. The information displayed in the report is provided by the automatic data rebuild program. By default, this information is provided by the rebuild program only after each lot of 100 records is processed. (This value can be altered by use of the loadfreq database option. (See also the Database Options - Rebuild, load and Batch Insertion Options section in Chapter 7 of the USer Guide). The screen is updated from this information after every *n* seconds. This command can be used to change the value of *n*. When selected, the message:

```
Enter new update delay (in seconds):   n
```

is displayed. The value, ***n***, is the current setting.

Exit monitor (Function⇒ Exit)

Exits from the monitoring procedure and returns to Admin. If the rebuild (or load) had not finished it will still continue in background.

The Rebuild Progress Report lists the following items:

Rebuild Type

This displays the type of rebuild requested by the DBA. The possible types are:

- Checking records
- Indexes
- Look-up tables
- Data file update

or a combination of these.

Rebuild Status

The status line indicates the rebuild function currently being performed. It displays one of following states:

- Checking records in data file
- Updating records in data file
- Re-entering records in data file
- Updating records in temporary file
- Re-entering records in temporary file
- Balancing Look-up tables
- Unbitslicing Index file
- Rebitslicing Index file
- Cleaning up
- Completed checking records
- Completed updating records
- Completed re-entering records

Insertion and Error counts

The number of records inserted and the number of erroneous records found is displayed. By default, these figures are updated every 50 records. To indicate the current progress of the rebuild, a percentage count is also displayed.

Rebuild Completion

When the rebuild is complete, the status line displays **Rebuild completed**, the figure **100%** is displayed for the percentage count of the database rebuilt and a message displays:

```
Completed rebuilding the database
```

If any errors occurred, the error file is then displayed using in Perusing mode. This error summary may be saved in a file if required. If the error file is not copied, a message displays:

```
Do you wish to save/print the report file? [yes] [no]
```

Choosing no discards the error file whereas a yes reply invokes the standard print or save procedure. Control then returns to the Admin screen and the database is ready for use.

Error reporting

If a system error is encountered during a rebuild, the status line displays:

```
System error
```

An error can be either recoverable or irrecoverable. Irrecoverable errors imply that the rebuild/load cannot be continued. In this situation that database must be restored from backup and the rebuild/load recommenced. Recoverable errors allow the rebuild/load to continue after the error has been fixed.

Recovery after System Crash

For large databases, a data rebuild may take a considerable amount of time, during which it is possible that the computer may crash for some unrelated reason. The rebuild facility has the capacity to recover from a crash and resume the rebuild. This section discusses the use of the recovery option.

After the machine has been rebooted the Rebuild/load/progress option should be selected. The status line indicates:

```
System crash
```

If the type of rebuild being performed is not a data file update, recovery is possible and a message displays:

```
Do you wish to recover? [yes] [no]
```

If yes is choosen, the recovery is commenced and the status of the recovery is displayed using the progress report monitoring facility.

Re-commencing a Rebuild

For some errors it is not possible to continue with the rebuild. These errors are generally database errors and are not caused by a system crash. Note however, that a system crash during a data file update is also considered to be an irrecoverable error.

If an irrecoverable error occurs, or a negative reply is given to the recovery prompt, the automatic data rebuild selection screen is displayed. The type of rebuild being performed before the crash is automatically selected and cannot be turned off. The normal rebuild procedure is followed.

Data File Update Recovery

If an error occurs, or the system crashes while a data file update is being performed, it is not possible to recover. The update must be restarted. Control is returned to the Admin screen and the database must be restored from backup (refer to the KE Texpress Maintenance Guide for information on restoring from backups). Once the database has been restored, the data file update may be re-commenced.

Appendix A
Texdesign Quick Reference Guides

This Appendix lists the keyboard accelerators that are available in **texdesign** during the various modes.

Field Editing Quick Reference

`Ctrl+H`	Move left one character position.
`Ctrl+L`	Move right one character position.
`Ctrl+I`	Insert characters before the current character.
`Ctrl+A`	Append characters after the current character.
`Ctrl+D`	Delete the character under the cursor.
`Ctrl+O`	Overstrike the character under the cursor with the next (single) character typed.
`Ctrl+C`	Copy data into the Copy buffer.
`Ctrl+E`	Enter data from the Copy buffer into the field.
`Ctrl+U`	Undo the previous change.
↵, ESC	Terminate Text edit mode.

Global Edit Mode Quick Reference

`Ctrl+G`	Copy screen image to a printer or Unix file.
`Ctrl+R`	Redraw the screen.
`Ctrl+Z`	Job control - stop process or invoke a shell.
`?`	Display helpful information.
`Ctrl+S`	Flow control - stop screen output.
`Ctrl+Q`	Flow control - restart screen output.

Insert Mode Quick Reference

Function Pull-down Menu

Ctrl+C, **F8**	Select the object (item, header or box).
Ctrl+TC	Select (mark) a Block.
ESC	De-select the current selection.
Ctrl+W, **F4**	Save the form.
Ctrl+X, **F1**	Exit without saving the changes.

Move Pull-down Menu

Ctrl+H, ←	Move 1 character position to the left.
Ctrl+L, →	Move 1 character position to the right.
Ctrl+J, ↓	Move 1 character position down.
Ctrl+K, ↑	Move 1 character position up.
H	Move left a tab stop.
L	Move right a tab stop.
J	Move down a tab stop.
K	Move up a tab stop.
Ctrl+F, **F5**	Move to the beginning of the form.
Ctrl+B, **F6**	Move to the end of the form.
Ctrl+TF	Find a prompt by its leading letters.

Edit Pull-down Menu

`Ctrl+I`	Insert new item.
`Ctrl+E`	Insert new header
`Ctrl+TE`	Insert new box.
`Ctrl+P`	Position (Move) selected objects.
`Ctrl+Y`	Copy object.
`Ctrl+TY`	Change object.
`Ctrl+TA`	Append fields to item
`Ctrl+D`	Delete the selected object(s).
`Ctrl+U`	Undo the previous command.

Define Pull-down Menu

`Ctrl+A`	Edit attributes of the current object.
`Ctrl+TN`	Edit the Form Expression.
`Ctrl+TH`	Create a hierarchy for the current item.
`Ctrl+TD`	Delete the hierarchy of the current item.
`Ctrl+TB`	Change the Form Bounds.
`Ctrl+O`	Change the Form Order.
`Ctrl+TO`	View the Form Order.

Info Pull-down Menu

`Ctrl+V`	Display the status of the form.
`Ctrl+TV`	Validate the form contents.
`Ctrl+TN`	Display the formcoordinates.

`Ctrl+TP`	View at a different privilege level.
`Ctrl+TW`	Copy/Print the image of the entire form.

Forms Pull-down Menu

`Ctrl+TK`	Extract a KE Texpress Key Form.
`Ctrl+TL`	Extract a KE Texpress Library Form.

Misc Pull-down Menu

`Ctrl+TI`	Switch to Query Form.

Attributes Mode Quick Reference

The following lists the keyboard accelerators of commands available when editing attributes in **texdesign.**

Function Pull-down Menu

`Ctrl+T`	Toggle (set/reset) the attribute.
`Ctrl+TE`	Edit the Field Expression.
`Ctrl+Y`	Copy an expression from a file.
`Ctrl+K`	Copy all attributes of a nominated field.
`Ctrl+X`, `F1`	Exit from editing attributes.

Move Pull-down Menu

`Ctrl+F`, `F5`	Move forward to the next editable attribute.
`Ctrl+B`, `F6`	Move backward to the previous editable attribute.
`Ctrl+N`	Move to the first editable attribute on the next page.
`Ctrl+P`	Move to the first editable attribute on the previous page.
`Ctrl+V`	Move to the attributes of the next object on the form.
`Ctrl+W`	Move to the attributes of the previous object on the form.

Edit Pull-down Menu

`Ctrl+U`	Undo the previous command.

Appendix B

Key Types

This section describes the Key types which are distributed with KE Texpress. Each Key type contains a given number of fields of the appropriate data type. In general, these attributes may not be altered. However, the field lengths, validation and prompt may be changed if necessary. A Key type may be designed by a DBA, after which it can be used by any DBA in the design of any database.

For an automatic Key value assignment to occur using the multi-field Key types, it is necessary to use the same prefix as that of the previous Key value entered by the user during the current session. Hence, at the commencement of a session, it is generally necessary to enter a prefix before any automatic Key value assignment is possible. The exception to this rule is when an assignment expression or a default value is used to automatically enter a value for the prefix.

annn

The annn Key type has the following format:

```
Key _ ______
```

It consists of two fields. Field 1 accepts string data. Its default validation is:

```
val = filled(#);
```

Field 2 accepts an integer. Its default validation is:

```
val = filled(#) && (# > 0);
```

Automatic Key values can be generated for this Key type.

integer

The integer Key type has the following format:

```
Key ______
```

It consists of a single field accepting an integer. Its default validation is:

```
val = filled(#) && (# > 0);
```

Automatic Key values can be generated for this Key type.

nnn.nnn

The nnn.nnn Key type has the following format:

```
Key _____ _____
```

It consists of two fields. Both fields accept integer values and have default validation of:

```
val = filled(#);
```

Automatic Key values can be generated for this Key type.

nnna

The nnna Key type has the following format:

```
Key ______ _
```

It consists of two fields. Field 1 accepts an integer. Its default validation is:

```
val = filled(#) && (# > 0);
```

Field 2 accepts string data. Its default validation is:

```
val = filled(#);
```

Automatic Key values can be generated for this Key type.

string

The string Key type has the following format:

```
Key __________
```

It consists of a single field accepting a string. Its default validation is:

```
val = filled(#);
```

yynnn

The yynnn Key type has the following format:

```
Key __ ______
```

It consists of two fields. Field 1 accepts integer data. Its default validation is:

```
val = filled(#);
```

Field 2 accepts an integer. Its default validation is:

```
val = filled(#) && (# > 0);
```

Automatic Key values can be generated for this Key type. If no value is supplied for the first field, then the last two digits of the current year are assigned.

Appendix C

KE Texpress Library Types

This section describes the currently available KE Texpress Libraries and their formats. The fields of these items have been defined to accept a restricted set of values during data entry.

ddmmyyyy

This has the following format:

```
Date __/__/____
```

Field 1 accepts an integer value in the range, 1 to the number of days in the month corresponding to field 2. Field 2 accepts a integer value in the range, 1 to 12. Field 3 accepts a 4 digit integer value which is non-negative. This library may be designated automatic and may also be ranged.

For range queries involving lower bounds, if either field 1 or field 2 is empty, it assumes the value, 1. If field 3 is empty, it assumes the value, 0.

For range queries involving upper bounds, if field 3 is empty, it assumes the value, 0. If field 2 is empty, it assumes the value, 12. If field 1 is empty, it assumes the value of the number of days in the corresponding month.

ddmmmyyyy

This has the following format:

```
Date __  ___  ____
```

Field 1 accepts an integer value in the range, 1 to the number of days in the month corresponding to field 2. Field 2 accepts a 3 character value which is one of:

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Field 3 accepts a 4 digit integer value which is non-negative. This library may be designated automatic, and may also be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 1. If field 2 is empty, it assumes the value, Jan. If field 3 is empty, it assumes the value, 0.

For range queries involving upper bounds, if field 3 is empty, it assumes the value, 0. If field 2 is empty, it assumes the value, Dec. If field 1 is empty, it assumes the value of the number of days in the corresponding month.

ddmmyy

This is identical to the ddmmyyyy library except that only two digits may be entered for the year value. It has the format:

Date __/__/__

This library may be designated automatic and may also be ranged.

ddmmmyy

This is identical to the ddmmmyyyy library except that only two digits may be entered for the year value. It has the format:

Date __ ___ __

This library may be designated automatic and may also be ranged.

ddmonyyyy

This library has the following format:

Date __ _________ ____

Field 1 accepts an integer value in the range, 1 to the number of days in the month corresponding to field 2. Field 2 accepts a character value which is one of:

January February March April May June July August September
October November December

Field 3 accepts a 4 digit integer value which is non-negative. This library may be designated automatic, and may also be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 1. If field 2 is empty, it assumes the value, January. If field 3 is empty, it assumes the value, 0.

For range queries involving upper bounds, if field 3 is empty, it assumes the value, 0. If field 2 is empty, it assumes the value, December. If field 1 is empty, it assumes the value of the number of days in the corresponding month.

mmddyyyy

This has the month field before the day field but otherwise is identical to the ddmmyyyy KE Texpress library. It has the format:

Date __/__/____

This library may be designated automatic and may also be ranged.

mmmddyyyy

This is similar to the mmddyyyy library except that the month field accepts a three character abbreviation of the month name (as described above). It has the format:

Date ___ __ ____

This library may be designated automatic and may also be ranged.

mmyy

This has the following format:

Date __/__

Field 1 accepts a integer value in the range, 1 to 12. Field 2 accepts a 2 digit integer value which is non-negative. This library may be designated automatic and may also be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 1. If field 2 is empty, it assumes the value, 0.

For range queries involving upper bounds, if field 2 is empty, it assumes the value, 0. If field 1 is empty, it assumes the value, 12.

monddyyyy

This is similar to the ddmonyyyy library (as described above) except that the first field represents the month and the second field represents the day of the month. It has the format:

Date _________ __ ____

This library may be designated automatic and may also be ranged.

yyyymmdd

This is similar to the ddmmyyyy KE Texpress library except that the order of the fields is reversed. It has the following format:

Date ____/__/__

This library may be designated automatic and may also be ranged.

yyyymmmdd

This is similar to the ddmmmyyyy KE Texpress library except that the order of the fields is reversed. It has the following format:

Date ____ ___ __

This library may be designated automatic and may also be ranged.

yyyybc

This has the format:

Date ____ __

Field 1 accepts a 4 digit integer value which is non-negative. Field 2 accepts a 2 character string which can take one of the values:

BC   AD

This library may be ranged. For range queries, if field 1 is empty, it assumes the value, 0. If field 2 is empty, it assumes the value, **AD**.

hhmm12

This has the format:

Time __:__ __

Field 1 accepts an integer value in the range 1 to 12. Field 2 accepts an integer value in the range 0 to 59. Field 3 accepts one of the strings:

AM   PM

This library may be designated automatic and may also be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 12. If field 2 is empty, it assumes the value, 0. If field 3 is empty, it assumes the value, AM.

For range queries involving upper bounds, if field 1 is empty, it assumes the value, 0. If field 2 is empty, it assumes the value, 59. If field 3 is empty, it assumes the value, PM.

hhmm24

This has the format:

Time __:__

Field 1 accepts an integer value in the range 0 to 23. Field 2 accepts an integer value in the range 0 to 59. This library may be designated automatic and may also be ranged.

For range queries involving lower bounds, if either field is empty, it assumes the value, 0.

For range queries involving upper bounds, if field 1 is empty, it assumes the value, 0. If field 2 is empty, it assumes the value, 59.

latdm

This has the format:

Latitude __ Deg _____ Min _

Field 1 accepts an integer value in the range 0 to 90. Field 2 accepts a real value in the range 0 to 59.99. Field 3 accepts a 1 character direction, one of '**N**', '**n**', '**S**' or '**s**'. This library may be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 0. If field 3 is empty, it assumes the value, **N**. If field 2 is empty, it assumes the value, 59.99, for queries where field 3 is **S** or **s**. It assumes the value, 0, otherwise.

For range queries involving upper bounds, if field 1 is empty, it assumes the value, 0. If field 3 is empty, it assumes the value, **N**. If field 2 is empty, it assumes the value, 59.99, for queries where field 3 is **N** or **n**. It assumes the value, 0, otherwise. Note also that in any comparisons, south is considered less than north.

latdms

This has the format:

Latitude __Deg __ Min __ Sec _

Field 1 accepts an integer value in the range 0 to 90. Field 2 accepts an integer value in the range 0 to 59. Field 3 accepts an integer value in the range 0 to 59. Field 4 accepts a 1 character direction, one of '**N**', '**n**', '**S**' or '**s**'. This library may be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 0. If field 4 is empty, it assumes the value, **N**. If either field 2 or field 3 is empty, it assumes the value, 59, for queries where field 4 is **S** or **s**. It assumes the value, 0, otherwise.

For range queries involving upper bounds, if field 1 is empty, it assumes the value, 0. If field 4 is empty, it assumes the value, **N**. If either field 2 or field 3 is empty, it assumes the value, 59, for queries where field 4 is **N** or **n**. It assumes the value, 0, otherwise. Note also that in any comparisons, south is considered less than north.

longdm

This has the format:

Longitude ___Deg _____ Min _

Field 1 accepts an integer value in the range 0 to 180. Field 2 accepts an real value in the range 0 to 59.99. Field 3 accepts a 1 character direction, one of '**E**', '**e**', '**W**' or '**w**'. This library may be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 0. If field 3 is empty, it assumes the value, **E**. If field 2 is empty, it assumes the value, 59.99, for queries where field 3 is **W** or **w**. It assumes the value, 0, otherwise.

For range queries involving upper bounds, if field 1 is empty, it assumes the value, 0. If field 3 is empty, it assumes the value, **E**. If field 2 is empty, it assumes the value, 59.99, for queries where field 3 is **E** or **e**. It assumes the value, 0, otherwise. Note also that in any comparisons, west is considered less than east.

longdms

This has the format:

Longitude ___Deg __ Min __ Sec _

Field 1 accepts an integer value in the range 0 to 180. Field 2 accepts an integer value in the range 0 to 59. Field 3 accepts an integer value in the range 0 to 59. Field 4 accepts a 1 character direction, one of '**E**', '**e**', '**W**' or '**w**'. This library may be ranged.

For range queries involving lower bounds, if field 1 is empty, it assumes the value, 0. If field 4 is empty, it assumes the value, **E**. If either field 2 or field 3 is empty, it assumes the value, 59, for queries where field 4 is **W** or **w**. It assumes the value, 0, otherwise.

For range queries involving upper bounds, if field 1 is empty, it assumes the value, 0. If field 4 is empty, it assumes the value, **E**. If either field 2 or field 3 is empty, it assumes the value, 59, for queries where field 4 is **E** or **e**. It assumes the value, 0, otherwise. Note also that in any comparisons, west is considered less than east.

Userid

This consists of an item with one field which accepts text input of up to eight characters. It has the format:

User Id ________

This library may be designated automatic.

video

This consists of an item with one field which accepts text input initially up to 25 characters. It has the format:

Video frames _________________________

This library is used to enter information such as frame numbers which can be used to drive avideodisc.

Appendix D

Noise Word List

Noise words are words deemed unlikely to distinguish a record. They are not indexed and so are ignored by KE Texpress if they are specified as query terms. They can be to added to or deleted, to meet specific needs of users by editing the noise word file. Noise words are stored in the file:

```
~texpress/ lang/english/noise/text
```

The default set of noise words used by KE Texpress is listed below:

```
a              about          all            also
although       an             and            another
any            are            as             at
be             because        but            by
for            here           if             in
into           is             it             its
of             or             so             than
that           the            then           there
this           though         to             too
very           was            were
```

Appendix E

Range Examples

In order for KE Texpress to perform efficient range queries on numeric data appropriate range intervals must be specified in the form design. If inappropriate range intervals are specified, queries can still be performed in the same manner, however response time will be slower.

The range intervals should reflect the anticipated distribution of records over all possible field values. The aim is to specify range intervals such that the number of records which falls between the interval pair is approximately constant. Range values of negative infinity and positive infinity can be assumed as the bounds of the first and last range values respectively.

Consider some examples of appropriate and inappropriate range intervals :

Location

Consider a database which contains latitude and longitude data defining the collection point of a certain record. If records are evenly distributed across Australia with only a small percentage collected elsewhere then the latitude and longitude range intervals should be specified simply for the Australian continent.

For latitude we might have:

Range? #

12 Deg 0 Min S **15 Deg 0 Min S** **18 Deg 0 Min S**
21 Deg 0 Min S **24 Deg 0 Min S** **27 Deg 0 Min S**
30 Deg 0 Min S **33 Deg 0 Min S** **36 Deg 0 Min S**
39 Deg 0 Min S **42 Deg 0 Min S**

and for longitude:

Range? #

115 Deg 0 Min E **118 Deg 0 Min E** **121 Deg 0 Min E**
124 Deg 0 Min E **127 Deg 0 Min E** **130 Deg 0 Min E**
133 Deg 0 Min E **136 Deg 0 Min E** **139 Deg 0 Min E**
142 Deg 0 Min E **145 Deg 0 Min E** **148 Deg 0 Min E**
151 Deg 0 Min E

Capture depth

Consider an item designed to store the capture depth of marine specimens. If the expected distribution of specimens is approximately 60% evenly between 0 and 10 metres; 25% between 10 and 25 metres and 15% between than 25 and 100 metres. An appropriate range interval specification is:

```
Range? #

2       4       6
8       10      15
20      25      50
75
```

whereas an inefficient range specification is:

```
Range? #

20      40      60
80      100
```

Appendix F

Forms Expression Examples

Assignment Expressions

Field scrolling

Consider a five field item (whose Id is **data**) which is to contain the five most recent values of a particular database attribute. When a new value is to be added we wish to automatically scroll the previous values down one field and discard the last value. The scrolling is to be triggered by entering **y** into the single field item called **scroll**. The following assignment expression on the scroll field will perform the required function:

```
if (filled(#))
{
        data_5 = data_4;
        data_4 = data_3;
        data_3 = data_2;
        data_2 = data_1;
        data_1 = "";
        # = "";
}
```

This expression achieves the desired result but if another field is added to the **data** item the assignment expression will need to be edited to cater for the new field. A more efficient way of writing the expression would to use a **for** loop and the KE Texpress functions **nfields(*id*)** which returns the number of fields in the item *id*.

```
if (filled(#))
{
        for (i = nfields(data);  i > 1; i--)
                data[i] = data[i - 1];
        data[i] = # = "";
}
```

This expression uses a temporary variable *i* and references the fields of the data item using the array subscript notation.

Degrees fahrenheit to celsius

Consider two items, each consisting of a single integer field. The item Id's are **fahren** and **celsius** respectively. Assuming all temperature data is in degrees fahrenheit, the corresponding celsius value can be automatically assigned. The following assignment expression for the 'celsius' field achieves the desired result.

```
if (filled( fahren))
        # = (fahren - 32) * 5/9;
else
        # = "";
```

Calculating a profit margin

Consider three items **Wholesale Price**, **Retail Price** and **Profit Margin** each consisting of a single field of real type. The item identifiers are **whole**, **retail** and **profit** respectively.

An assignment expression for the 'profit' item (assuming a percentage calculation) could be as follows:

```
if (filled(whole) && whole > 0.0 && filled(retail))
        # = ((retail - whole) / whole) * 100;
else
        # = "";
```

Validation Expressions

Value within range

A validation expression to ensure that if a value is entered for the current field then the value is within the range 0 to 100 inclusive.

```
if (filled(#) && (# < 0 || # > 100))
        val = 0;
```

String conformity

A validation expression to ensure that the entered string has the correct prefix and is at least 6 characters in length.

```
if (length(#) < 6 || prefix(#, 3) != "ABC"))
        val = 0;
```

Sum of other fields

A validation expression to ensure that the entered value is less greater than the sum of the fields in another item (Id **data**). Note the use of a temporary variable called **sum**.

```
if (filled(#))
{
        sum = 0;
        for (i = 1; i <= nfields(data); i++)
                sum = sum + data[ i];
        if (# <= sum)
                val = 0;
}
```

Branch Expressions

Yes/No branch

A branch expression to move the cursor to the appropriate field dependent on an affirmative or non-affirmative response. Any other response will result in the cursor moving to the next field in the form order. It is assumed that items with Id's **yes** and **no** exist in the form.

```
if (# == "y" || # == "Y")
        next = addr(yes);
else if (# == "n" || # == "N")
        next = addr(no);
```

Multiple value branch

Move the cursor to the appropriate *colour* field. It is assumed that items with Id's **red**, **blue**, **yellow** and **unknown** exist in the form.

```
if (# == "red")
        next = addr(red);
else if (# == "blue")
        next = addr(blue);
else if (# == "yellow")
        next = addr(yellow);
else
        next = addr(unknown);
```

Calculated branch

Move the cursor to the appropriate field of the **data** item or if an erroneous value is entered move to the item with Id **illegal**.

```
if (filled(#))
{
        if (# >=1 && # <=  nfields(data))
                next = addr(data[#]);
        else
                next = addr(illegal);
}
```

Appendix G

Forms Expressions Grammar

Following is the syntax of the grammar used for assignment, validation and branch expressions. The operator precedence table lists operators in order of decreasing precedence and the order of evaluation (associativity) of operators of equivalent precedence is also listed. Comments may be included in expressions and should be enclosed by/* and */.

Operator	Associativity
( ) [ ]	left to right
! - ++ --	right to left
* / %	left to right
+ -	left to right
< <= > >=	left to right
== !=	left to right
&&	left to right
||	left to right
= += -= *= /= %= .=	right to left

In the expression grammar text enclosed in `/* */` are comments provided for assistance. Symbols or words in bold type are typed directly as specified. Any symbol on the left hand side of a **:** or **|** may be replaced by any of the sequences on the right hand side of the **:** or **|** for that symbol. A semi-colon **;** on a line by itself terminates the selections for a particular symbol.

```
expression      :       stmtlist
;

stmtlist        :       /* null */
                |       stmtlist stmt
;

stmt    :       ;
        |       commaexpr ;
        |       if cond stmt
        |       if cond stmt else stmt
        |       while cond stmt
        |       for ( forexpr ; forexpr ; forexpr ) stmt
        |       { stmtlist }
        |       break ;
```

```
         |       continue ;
         |       return ;
;
cond     :       ( commaexpr )

forexpr  :       /* null */
         |       commaexpr
;

commaexpr        :       expr
         |       commaexpr ,  expr
;

expr     :     constant
         |     var
         |     assign
         |     function-name ( arglist )
         |     ( expr )
         |     expr + expr    /* add */
         |     expr - expr    /* subtract */
         |     expr * expr    /* multiply */
         |     expr / expr    /* divide */
         |     expr % expr    /* modulo */
         |     - expr         /* negate */
         |     expr . expr    /* string concatenate */
         |     expr == expr   /* equals */
         |     expr != expr   /* not equals */
         |     expr > expr    /* greater than * /
         |     expr >= expr   /* greater than or equal */
         |     expr < expr    /* less than */
         |     expr <= expr   /* less than or equal */
         |     expr || expr   /* logical or */
         |     expr && expr   /* logical and */
         |     ! expr         /* logical not */
;

constant :     PI
         |     E              /* base of natural logs */
         |     GAMMA          /* Euler constant */
         |     DEG            /* degrees per radian */
         |     PHI            /* golden ratio */
;

var      :     variable
         |     item-Id
         |     field-Id
         |     item-Id  [  expr ]
;

assign   :       var = expr   /* assignment */
         |     var += expr    /* add and assign */
         |     var -= expr    /* subtract  and assign */
         |     var *= expr    /* multiply and assign */
         |     var /= expr    /* divide and assign */
         |     var %= expr    /* modulo and assign */
         |     var .= expr    /* concatenate and  assign*/
         |     ++ var         /* pre-increment */
         |     -- var         /* pre-decrement */
         |     var ++         /* post-increment */
```

```
          |      var --         /* post-decrement */
;

arglist  :        /* null */
          |       expr
          |       arglist ,  expr
;
```

Appendix H

Forms Expressions Constants and Functions

Constants

The forms expression grammar supports the following constants :

PI	3.14159265358979323846
E	2.71828182845904523536
GAMMA	0.57721566490153286060
DEG	57.29577951308232087680
PHI	1.61803398874989484820

Functions

Following is a list of functions supported by the forms expression grammar. Each function is listed with its preferred argument types and the type of the value the functions returns. If the supplied arguments are not of the correct type they are automatically converted.

In the following list, where an integer or string value is requested a field Id may be substituted. Functions which return a boolean value in effect return the integer value 1 (for true) or 0 (for false).

Conventions used for argument types are:

f	**field Id**
s, t	**string**
i, j	**integer**
r	**real**

Function name	Returns	Description
tointeger(*s***)**	integer	truncate to integer
trunc(*s***)**	integer	truncate to integer
round(*s***)**	integer	round to integer
toreal(*s***)**	real	convert to real
tostr(*i***)**	string	convert to string

Function name	Returns	Description
to string (_I_**)**	string	convert to string
tolower(_s_**)**	string	convert to lower case
toupper(_s_**)**	string	convert to upper case
isalpha(_s_**)**	boolean	true if all characters alphabetic
isdigit(_s_**)**	boolean	true if all characters digits
ispunct(_s_**)**	boolean	true if all characters punctuation
isalnum(_s_**)**	boolean	true if all characters alpha-numeric
islower(_s_**)**	boolean	true if all characters lower case
isupper(_s_**)**	boolean	true if all characters upper case
strlen(_s_**)**	integer	number of characters in string
firststr(_s, t_**)**	integer	first index of string _t_ in _s_
laststr(_s, t_**)**	integer	last index of string _t_ in _s_
substr(_s, i, j_**)**	string	_j_ characters of _s_ starting at _i_
prefix(_s, i_**)**	string	first _i_ characters of _s_
suffix(_s, i_**)**	string	last _i_ characters of _s_
textlen(_s_**)**	string	length of text
textcmp(_s, t_**)**	integer	text comparison, returns < 0 if s < t, 0 if s == t, or > 0 if s > t
textncmp(_s, t, i_**)**	integer	compare only _i_ characters
textccmp(_s, t, j_**)**	integer	if _j_ is 1 case is significant
textnccmp(_s, t, i, j_**)**	integer	compare only _i_ characters and if _j_ is 1, then case is significant
stgcmp(_s, t_**)**	integer	string comparison, returns < 0 if s < t, 0 if s == t, or > 0 is s > t
stgncmp(_s, t, i_**)**	integer	compare only _i_ characters
stgccmp(_s, t, j_**)**	integer	if _j_ is 1 case is significant
stgnccmp(_s, t, i, j_**)**	integer	compare only _i_ characters and if _j_ is 1, then case is significant
empty(_f_**)**	boolean	true if field empty
filled(_f_**)**	boolean	true if field filled
nfields(_f_**)**	integer	number of fields in item f is in
nfilled(_f_**)**	integer	number of filled fields in item _f_ is in

Function name	Returns	Description
fldno(*f***)**	integer	field number of *f*
fldlen(*f***)**	integer	field length of *f*
prevvalue(*f***)**	string	contents of field *f* before last edit
ditto(*f***)**	string	contents of field *f* in previous record
getditto(*f1, f2, ...***)**	integer	get new ditto record, *f1*, f2, ... are the key field values, returns < 0 on failure
addr(*f***)**	field address	
cont(*field address***)**	string	
nextitem(*f***)**	field address	returns address of first field of next item in form order following item with field *f*
backward()	boolean	true if branch expression was invoked by a backward moving command
insert(*s, i , f***)**	boolean	in the item containing field *f*, insert *s* in field number *i* and scroll trailing fields down. Returns true if successful, or false otherwise
append(*s, f***)**	integer	in the item containing field *f*, put *s* in the first empty field. Returns field number if successful, or 0 otherwise
delete(*i , f***)**	integer	in the item containing field *f*, delete the value in the field *i* and scroll trailing fields up. Returns field number if successful, or 0 otherwise
find(*s, f***)**	integer	in the item containing field *f*, returns the number of the field containing *s*, or if *s* is not found, returns 0.
printf(*s, args , ..***)**	string	Similar to C language printf function
message(*act, s, args,..***)**	integer	display message *s*. *act* is either "wait" or "nowait"
display(*i, j, s, args,..***)**	integer	display message *s* on the screen at position *i, j*. "nowait" is assumed

Function name	Returns	Description
ask(*s*, *args* , ..**)**	integer	ask question*s*. return "y" or "n"
redraw()	integer	redraw the screen
clearmess()	integer	clear last message from screen
bell(*i***)**	integer	ring terminal bell *i* times
sleep(*i***)**	integer	sleep *i* seconds
newfile()	string	create a new (temporary) file and return its name
system(*s*, *args* , ..**)**	string	Similar to Unix system function
run(*s*, *args* , ..**)**	string	interactive system() function
uid()	string	current Unix user Id
gid()	string	current Unix group Id
privlevel()	integer	current user privilege level
mode()	string	current mode (Insert, Edit, etc.)
matchcnt()	integer	no. of records matching query
dispform()	string	return name of current display form
copyform(*fm*, *s***)**	integer	copy current record via the Report form *fm* to file *s*
loaditem(*f*, *s***)**	integer	load lines from file *s* into consecutive fields of the item containing field *f*, and return the number of fields loaded
addstr(*i***)**	integer	add *s* to input queue, and return number of characters added
addmacro(*i***)**	integer	add *s* to macro queue, and return number of characters added
getchar()	string	return next character typed
getenv(*s***)**	string	return the setting of the environment variable *s*
year()	integer	current year
month()	integer	current month
months()	string	short current month
monthl()	string	long (full) current month
day()	integer	current day
hour()	integer	current hour

Function name	Returns	Description
minute()	integer	current minute
second()	integer	current second
newday(*d, m, y, a***)**	integer	returns the day of the month after adding *a* days to the *d, m, y* date
newmonth(*d, m, y, a***)**	integer	returns the month after adding *a* days to the *d, m, y* date
newmonths(*d, m, y, a***)**	string	returns the short month name after adding *a* days to the *d, m, y* date
newmonthl(*d, m, y, a***)**	string	returns the long month name after adding *a* days to the *d, m, y* date
newyear(*d, m, y, a***)**	integer	returns the year after adding *a* days to the *d, m, y* date
valtime(*h, m***)**	boolean	true if a valid (24 hour clock) time value : integer hour (*h*), integer minute (*m*)
valdate1(*y, m, d***)**	boolean	true if a valid date value, integer year (*y*), integer month (*m*), integer day (*d*)
valdate2(*y, m, d***)**	boolean	true if a valid date value, integer year (*y*), 3 character month (*m*), integer day (*d*)
valdate3(*y, m, d***)**	boolean	true if a valid date value, integer year (*y*), full string month (*m*), integer day (*d*)
vallat1(*d, m, s, D***)**	boolean	true if a valid latitude, integer degree (*d*), integer minute (*m*), integer second (*s*), direction (*D*) of N or S
vallat2(*d, m, D***)**	boolean	true if a valid latitude, integer degree (*d*), real minute (*m*), direction (*D*) of N or S
vallong1(*d, m, s, D***)**	boolean	true if a valid longitude, integer degree (*d*), integer minute (*m*), integer second (*s*), direction (*D*) of E or W
vallong2(*d, m, D***)**	boolean	true if a valid latitude, integer degree (*d*), real minute (*m*), direction (*D*) of E or W

Function name	Returns	Description
fabs(*r*)	real	similar to Unix math library
sqrt(*r*)	real	"
exp(*r*)	real	"
pow(*r*)	real	"
log(*r*)	real	"
log10(*r*)	real	"
floor(*r*)	real	"
sin(*r*)	real	"
cos(*r*)	real	"
tan(*r*)	real	"
asin(*r*)	real	"
acos(*r*)	real	"
atan(*r*)	real	"
atan2(*r*)	real	"
sinh(*r*)	real	"
cosh(*r*)	real	"
tanh(*r*)	real	"

Appendix I

Texforms Validation Error Messages

These messages may be displayed in the validation report produced by **texforms**, indicating form inconsistencies. Messages are divided into two categories, errors and warnings.

Errors indicate that **texforms** has detected an inconsistency in the form which must be corrected before the database can be used.

Warnings indicate a non-critical inconsistency, however they should still be investigated.

Query Form Errors

Key item has a lower range without an associated upper range

If a Key item is to have ranges then it must have associated with it both a lower range item and an upper range item.

Key item has an upper range without an associated lower range

If a Key item is to have ranges then it must have associated with it both a lower range item and an upper range item.

Insertion Form Errors

No fillable items

The form does not contain any items which can be user keyed.

Item Attribute Errors

Item Id is compulsory

Each item must have a unique identifier.

Item Id must commence with a letter

The item Id must commence with a letter a - z or A - Z.

Non unique item Id "*Id*"

The item identifier *id* is already associated with an item.

No ranges specified

The range attribute has been set but no range values have been entered.

No database name specified

An item with the linked key item attribute set must have an value specified for the linked database name.

Current database can not be referenced

The linked database name can not be the name of the current database.

You are not permitted to use the "*dbname*" database

You do not have the privilege to access the *dbname* database.

"*dbname*" database non existent

The linked database name *dbname* refers to a non-existent database.

"*dbname*" Insertion form not yet designed

The Insertion form of the linked database *dbname* has not yet been designed.

No Key item in "*dbname*" database

The linked database *dbname* does not have a Key item, hence the linked database facility may not be used.

Incompatible field *n* types (*type1* : *type2*)

The type (*type1*) of field *n* of the local Key item differs from the type (*type2*) of field *n* of the linked Key item.

Incompatible field *n* lengths (*len1* : *len2*)

The length (*len1*) of field *n* of the local Key item differs from the length (*len2*) of field *n* of the linked Key item.

Incompatible number of fields

The local Key item and the linked Key item have an incompatible number of fields.

Item Attribute Warnings

"*dbname*" database not initialized

The linked linked database attributes refer to a database that has not yet been initialized.

Database Key item never referenced

Indicates that no item in the form references the linked Key item, hence the item is redundant.

Field Attribute Errors

No local item Id specified

The local item Id used to access the linked database has not been specified.

No local item with Id "*Id*"

The local item Id specified does not exist in the Insertion form.

Local item "*id*" not a linked Key item

The local item specified does not have the linked key item attribute set.

Bad local item Id fieldspecifier "*id*"

It is not necessary to give a fieldspecifier for a local item Id.

No linked field Id specified

The linked field Id of a field in the linked database has not been specified.

No linked item with Id "*Id*"

The linked field Id specified in the linked database field attribute does not exist in the linked database.

Bad linked item Id fieldspecifier "*id*"

The field specifier for the linked field Id exceeds the number of fields for that item in the linked database.

Linked field "*id*" type (*type*) incompatible with current field

The field type (*type*) of the linked field with Id *id* is not the same as the current field type.

Linked field "*id*" length (*length*) incompatible with current field

The field length (*length*) of the linked field with Id *id* is not the same as the current field length.

reference to undefined variable "*var*"

The variable*var* has been reference without previously being assigned a value.

illegal assignment to field "*fid*"

Fields may only be assigned values in assignment expressions.

"*func*" too few arguments (), *j* expected

The inbuilt function *func* requires *i* arguments but has been called with only *j* arguments.

"*func*" too many arguments (), *j* expected

The inbuilt function *func* requires *i* arguments but has been called with *j* arguments.

semi-colon missing

A semi-colon was expected at the end of the statement.

illegal indirect expression

An expression involving the field subscripting facility [ ] has been formed incorrectly.

illegal break statement

A break statement has been encountered outside of a while or for loop.

illegal continue statement

A continue statement has been encountered outside of a while or for loop.

bad field reference "*id*"

The field with Id*id* does not exist.

syntax error

The expression does not conform to the grammar.

missing quote

A string value has not been terminated with an end quote.

Field Attribute Warnings

identifier truncated

The item or field Id has too many characters. The forms editor has truncated it to the maximum permitted length.

id field specifier truncated

The field specifier of the field Id has too many characters. It has been truncated to the maximum permitted length.

string truncated

A string value has too many characters. It has been truncated to the maximum permitted length.

integer truncated

An integer value has too many digits. It has been truncated to the maximum permitted length.

real truncated

A real value has too many digits. It has been truncated to the maximum permitted length.

Appendix J

Videodisc Players Supported

KE Texpress currently supports the following videodisc player models:

Pioneer	**L4100**	Teletext supported
Pioneer	**TQ3032**	Teletext supported
Philips	**VP835**	Teletext supported
Sony	**LDP1500**	Teletext not supported

Index