tablesONLINE/CICS User's Guide

Release 6.0.3

Document Number: TBM005-R6.0.3v1.1

Publication History

Maintenance Release 6.0.1, - May 2004
Service Pack 2 R6.0.1v1.1 - November 2004
tableBASE Release 6.0.2v1.0 - May 2005
tableBASE Release 6.0.3 Preliminary - February 2009
tableBASE Release 6.0.3 Version 1.0 - April 2009

tableBASE Release 6.0.3 Version 1.1 - July 2010

Technical Support Hotline: (613) 523-5588

DataKinetics Ltd.
202 - 2460 Lancaster Road
Ottawa, ON
Canada K1B 4S5

Telephone: (613) 523-5500
1-800-267-0730 (toll free in the US and Canada)

Facsimile: (613) 523-5533

Email: tableBASE@dkl.com

http://www.dkl.com

Table of contents

Preface

This guide describes tablesONLINE/CICS, the online user interface to tableBASE available for a CICS environment.

Audience

This guide is intended for:

- analysts performing online table design and definition.
- programmers performing online table maintenance and testing, and, requiring interactive access to tableBASE calls.
- end users performing online table maintenance.

What You Should Know to Use this Guide

You should be familiar with an MVS or VSE environment, basic CICS sign-on procedures, and the content of the tableBASE Concepts and Facilities document.

There are terms used throughout the product documentation that are unique to tableBASE and tablesONLINE. These terms are identified and described below.

tableBASE and tablesONLINE term	**Description**
Data Table	Refers to the raw data. Each Data Table has a table definition (DT-BLOCK) that is used to generate the Index for the Data Table.

tableBASE and tablesONLINE term	**Description**
Index	An Index is defined for each Data Table. A Data Table Index is generated dynamically when a table is opened, based on the information in the table definition (DT-BLOCK).
Alternate Index	An Alternate Index is an Index that may be defined for a Data Table. The Alternate Index has an Alternate Index definition (ALT-DEFINITION) that defines the key, organization, and search order. Each Data Table may have as many Alternate Indexes as desired or none.

In addition, tablesONLINE has:

Term	**Description**
View	A View provides the field, edit, and display attributes for a Data Table with its Index.
Alternate Index View	An Alternate Index View is identical to a View, but applies to a Data Table with its Alternate Index.

Guide Conventions

This guide uses the following conventions:

Convention Example	**Description**
`Screen Text`	Text that appears on-screen is displayed in Courier font.
NEW, N	Commands and line commands are displayed in full uppercase characters.
Edit/Browse Table screen	Titles of screens are displayed in title case and in bold font.
View Name	Names of menu options and field names are displayed in title case.
<End>	Names of keys are enclosed in angled brackets.

What is Covered in this Guide

Chapter 1 provides an overview of some of the concepts and structures used by tablesONLINE.

Chapter 2 provides the basic information required to use tablesONLINE. It includes the sign on and sign off procedures and information about menu screen navigation.

Chapter 3 describes how to edit and browse Data Tables using tablesONLINE. This is prerequisite knowledge for creating tables and Views, which is covered in Chapter 4.

Chapter 4 describes the process used to create tables and Views using tablesONLINE. Both the basic table creation functions and the more advanced features such as Display Masks, data validation, use of alternate Views, and display order editing are discussed. Also covered in this chapter is the option for creating COBOL copybooks for tablesONLINE tables, printing of Views, restructuring of Data Tables, and the specification of many-to-many (M2M) relationships.

Chapter 5 describes the use of the tablesONLINE utility facilities: copy, delete, rename, passwords, and the User Profile.

Chapter 6 describes how to build table-driven applications by editing the Application Driving Tables delivered with tablesONLINE.

Chapter 7 provides a reference list of all messages that are returned by tablesONLINE.

For ease of explanation, this guide has been written with the assumption that the product has not been customized. Since it is possible that some of the facilities described in this guide have been customized by your tableBASE administrator, the screens, PF keys, transaction codes, etc. described in this guide may not exactly match those used in your environment.

What is New in Version 6

New features have been added tablesONLINE/CICS Version 6:

1. new Action Code added that tags changed rows with the User ID of the person making the change

2. enforce unique primary keys with an Alternate Index View

3. define the initial value, upper bound, and lower bound of a field

4. control simultaneous updates by multiple users to a single table

Note: TBOL/CICS requires that working storage be initialized to binary zeroes. Under LE, this is accomplished by ensuring that the run-time storage option

for the CICS region is STORAGE=(00,...). See the LE Customization manual for guidance in creating a CEEUOPT or CEEROPT module.

Naming Protocol

Version 6 features the new tableBASE naming protocol. All tableBASE executables begin with DK1 for easy identification, a prefix that has been reserved for exclusive use with IBM.

Aliases are retained so that no changes are required to your existing applications.

Release 6.0.3 modules should be used to entirely replace earlier Version 6 modules.

Additional tableBASE References

This guide is one of several that describe tableBASE; others include:

1. Concepts and Facilities
2. tableBASE Batch Utilities Guide
3. tableBASE Administration Guide
4. tableBASE Programming Guide
5. tableBASE Installation Guide
6. tablesONLINE/ISPF User's Guide
7. tableBASE Workshop Guide
8. Quick Reference Guide

1 Introduction

This chapter provides an overview of tablesONLINE/CICS. First, we will look at what tablesONLINE is, what it can be used for, and by whom. We then discuss the interaction between tablesONLINE and tableBASE. Finally, we examine the library structure used by tablesONLINE/CICS.

What is tablesONLINE/CICS

tablesONLINE/CICS is a flexible, interactive front end for tableBASE. It provides speed and convenience for end users wanting to create, update, manipulate, test, and process Data Tables. tablesONLINE handles data entry and table editing tasks and provides the access controls and data validation services essential to such tasks.

The product is highly configurable so that you can customize it to meet your needs and those of your users. In fact, because tablesONLINE is itself a table-driven system, you can readily tailor it to build a wide variety of applications. tablesONLINE can help you to control program complexity and reduce development and maintenance workloads.

Types of Users

tablesONLINE can be used to perform a wide variety of tasks by many different levels of users. The following is a list of the various types of jobs that people do and the tablesONLINE/CICS functions that they use to perform their jobs.

- **Database Administrators**
  Centralized control of table updating
  Interface with security systems
- **Analysts**
  Online table design and definition
- **Programmers**
  Online table maintenance and testing
  Interactive access to tableBASE calls
- **End Users**
  Online table maintenance without the need for IT department involvement

Uses of tablesONLINE

tablesONLINE is used interactively to manipulate tables for any tableBASE application and it can be used as a framework for building interactive applications. Most commonly, it is used as an interactive data entry and validation system for creating and maintaining Data Tables that may be used by tableBASE and other application programs.

Maintain Systems

When applications programs are table-driven, tablesONLINE becomes a system maintenance tool, as well as a data manipulation tool. Table-driven software can be reconfigured without recompiling, simply by editing the tables that control it. This significantly reduces maintenance programming for many applications.

Customize

Because tablesONLINE is itself table-driven, you can customize or reconfigure it for your own organization simply by editing the tables that control it. Developing application specific menus and/or data entry screens is one common example of such re-configuration. Another example is the ability to use tables to define permissions for the applications and data that each user has access to.

When using table-driven programs, systems can be set up so that users can reconfigure their own software rather than having to call on the IT department for assistance with minor changes. This arrangement improves service to users while reducing the IT department workload.

Depending on the application, it is possible for system managers, application developers and end users to reconfigure tablesONLINE. Some of the application specific modifications you may want to make are:

- adding help screens or tutorial material that is appropriate to local usage
- calling your own validation routines during data entry
- calling other routines to perform security checks, calculations, or other operations that are appropriate to your application
- setting up tables that control user access to the system

Build Applications

tablesONLINE includes both a menu front-end and a table editor. These components are used to build applications. The menu system manages the calling of various programs that make up tablesONLINE. It is designed as a tool able to call any program, to initiate any CICS transaction and to manage complex sequences of programs and/or transactions. The menu system can be made invisible when necessary. For example, the system may be set up to run an entire sequence of programs without displaying a menu for each step, so that the system, rather than the user, controls the sequence.

Extend Its Power

The table editor provides another type of facility, driven by changes in data rather than by choices of user actions. The user exit programs that the editor calls can be used to construct powerful data-driven systems.

tablesONLINE provides facilities for creating column definitions for each row in a Data Table. These definitions, called Views, are managed by tableBASE and used by tablesONLINE to control the display and editing of Data Tables. For both Views and Data Tables there are utility functions, such as copying and renaming, and functions to create, edit and display.

Validate Data Automatically

Whenever tables are edited using tablesONLINE, data validation is performed by extensive routines that are built into the product. These routines check for valid data based on the field type and information contained in other tables. The Views may also contain the names of additional user-written validation routines. If specified, these routines will be called whenever data in relevant fields are edited.

tablesONLINE Tables

One of the primary functions of tablesONLINE is the creation of tables. The completed tables may be accessed by application software or by tableBASE utilities. The following are some of the ways in which the Data Tables you create with tablesONLINE/CICS can be accessed:

- tableBASE batch utilities such as TBEXEC
- CICS applications that invoke tableBASE via the interface routines TBLBASE (or, for compatibility with releases before tableBASE Release 5.0, through TBCALLC and TBASEC).

tablesONLINE/CICS and tableBASE

tablesONLINE/CICS and tableBASE work together to provide the tools necessary to build applications and the tables with which these applications work. In some cases, these applications are constructed by the creation of customized tables, while in others they may require that application programs be written to make use of tableBASE and to interact with tables created in tablesONLINE.

Within a CICS environment, tableBASE and tablesONLINE provide the following ways to access tables created using tablesONLINE:

- **TBLBASE**
  tableBASE calls from within CICS programs
- **tablesONLINE**
  Interactive table manipulation
- **TBDRIVC**
  Interactive use of TBLBASE calls, mainly for testing. TBDRIVC can be called from tablesONLINE to create a complete, interactive table access facility.

Note: TBCALLC and TBASEC are supported for backward compatibility with all tableBASE releases previous to Release 5.1.

Detailed information on the use of TBLBASE and TBDRIVC are provided in the tableBASE Programming Guide.

tablesONLINE Libraries

Both the tables that are created using tablesONLINE and the tables that drive the application itself are stored in libraries. These libraries have been set up and are provided with the delivered defaults. You may want to store the tables that you create in these libraries or you may want to create your own library to house your organization's tables. The following is a list of the delivered libraries with a brief description of their intended purpose.

System Library

The tablesONLINE System Library (TBSYSLB) contains all of the system Views used in table definition, utility functions, etc. This library also contains tables used internally by tablesONLINE. The tables in this library have write passwords and are, in general, not intended to be modified by users.

Application Control Table Library

The Application Control Table Library contains the Application Control Table (ACT) used in system administration, security control, and user environment definition. This library is intended to hold any other security related tables that you may require for your own security procedures.

The tables found in the System Library and ACT Library contain the information that tablesONLINE requires to configure itself, and, to find the libraries containing the Data Tables on which it will operate.

Application Driving Tables Library

The Application Driving Tables Library contains tables for: menus, message text, PF key definitions, command aliases, help text, description text, and, a list of libraries. These are the tables that are required to create table-driven applications.

Chapter 6, *Building Applications*, describes how the Application Driving Tables are edited and used to create new applications.

Note: Special Views are provided in TBSYSLB for the tables in both the Application Control Table Library and the Application Driving Tables Library so that these tables can be edited with tablesONLINE.

Data Dictionary Library

The Data Dictionary Library is an optional library provided for the centralization of all Views and organization-standard data values that are edited by the table lookup at data entry (that is, the data values that are shared throughout an organization and that are used during data entry to restrict the input).

Data Table Libraries

These libraries are created by users to contain the Data Tables unique to an organization. Data Tables fall into three main classes:

- **Common corporate data**, used by several applications and many users but altered only by authorized users or specialized applications. Often, all such data is kept in one library and all applications look there. In some installations, there may be a hierarchy of corporate data, divisional data, departmental data, and so on.
- **Application-specific user data**, typically held in a separate library searched by only one stand-alone application or a group of related applications.
- **Application-driving data**, seen as data by the developer but not by the user. The user sees it as part of the system and typically does not manipulate it.

You can choose to place all application-driving data in one central library or place it in application specific libraries along with the user data. Both methods, and various compromises, are possible.

A typical application, then:

- learns its language from the System Library tables;
- finds its program in the Application Control Table Library and finds the Application Driving Tables in the Application Driving Tables Library; and
- operates on data from both corporate and private Data Table libraries.

2
Getting Started

This chapter provides the basics for using tablesONLINE. It begins with a broad overview of logging in to, and exiting from, tablesONLINE, followed by a discussion of common facilities, such as the help system, PF key definitions, security features, and customization of the user environment. The remainder of the chapter discusses the **Menu** screen.

Note: tablesONLINE allows each organization to reconfigure the PF keys used by tablesONLINE/CICS. All of the descriptions and references to PF keys in this guide refer to the delivered default PF key values. If your organization has already customized these keys, you may find that the PF keys do not perform the functions described in this guide. Please check with your tableBASE administrator.

Starting tablesONLINE

Access tablesONLINE by providing the appropriate CICS transaction code. TBOL is the default setting; however, you should consult your administrator to confirm the necessary code. Depending on your requirements, your system will be set up so that you will access a specific tablesONLINE menu on start-up. If you are an application developer, you will likely see a series of menus beginning with the Application Developer's menu (see Figure 2-1).

```
tablesONLINE 6.0.2 Administrator ---------- Application Developer's Menu ------
COMMAND ===>


To select, enter number/symbol on command line:

A    EDIT TABLE           - Add/Change/Delete Rows in a Table
B    BROWSE TABLE         - Display Contents of a Table
C    TBDRIVC              - Execute TBLBASE Commands
D    DEFINE TABLE         - Define Table, View and Data Descriptions
U    UTILITIES            - Copy/Rename/Delete a Table
1    EDIT MENU TABLE      - Add/Change/Delete Application Menu Items
2    EDIT PFKS TABLE      - Add/Change/Delete Application PF Keys
3    EDIT CMDS TABLE      - Add/Change/Delete Application Alias Commands
4    EDIT HELP TABLE      - Add/Change/Delete Application Help Items
5    EDIT MSGS TABLE      - Add/Change/Delete Application Messages
6    EDIT DESC TABLE      - Add/Change/Delete Application Screen Descriptions
7    EDIT LIBR TABLE      - Add/Change/Delete Application Library Names

Enter HELP at any stage for help within tablesONLINE.
Enter PF for program function key assignments.
Enter X to suspend tablesONLINE and return to CICS.
```

Figure 2-1: Application Developer's Menu

This initial menu is configurable within tablesONLINE by the system administrator. The menus can be modified so your initial menu may not appear exactly as shown. If you are an end user, the first menu that you see may be a tablesONLINE editing or browsing menu, or it may be a custom menu created by an application developer for the specific application you will be using.

Although menus are configurable, lines 1 and 2 of each menu are seldom changed. Line 1 contains the screen title. Line 2 is known as the command line; this is where you enter commands and other information to instruct tablesONLINE to perform a particular task.

Screen Types

There are four basic screen types used in tablesONLINE. You should be familiar with each of these types as they allow you to perform all of the operations required to define and edit tables. The screen types are **Menu**, **Identify Table/Row**, **Edit/Browse Table**, and **Edit/Browse Row** (see TableTitle 2-1).

Table 2-1: Screen Types

Screen	**Usage**
Menu	Selecting tablesONLINE actions
Identify Table/Row	Selecting a table
Edit/Browse Table	Selecting rows for browsing or editing
Edit/Browse Row	Actual data browsing or editing

These screens are fully described in this chapter and the next chapter.

Exiting tablesONLINE

When you have completed all of the operations you wish to perform in tablesONLINE and wish to exit the program, step back through the screens until you reach your start-up menu. This is done using the <PF3> key. Pressing <PF3> from your start-up menu exits the program and returns to CICS where you can proceed with other tasks, or, sign off the system using your standard CICS sign-off procedure.

Using Help

Help can be obtained by either typing HELP on the command line or pressing <PF1>.

tablesONLINE provides context-sensitive help when the cursor is positioned on a blank command line, general help and a list of possible commands is given. If the command line is blank and the cursor is positioned somewhere else on the screen, then the cursor position dictates the help that is offered. For a table editing screen, help is provided for the field that contains the cursor. For a **Menu** screen, help is provided for the menu option where the cursor is positioned.

All of the help messages in tablesONLINE are table-driven and may be modified by the user. For information on editing Help tables see “Help Tables” on page 139.

Levels of Help

There are four different levels of help.

The first level of help displays any multi-line error messages. A single message is shown on the main screen but if there is more than one line of messages they are saved and the main screen displays:

"TB-5003W Enter Help command for a display of 2 lines of messages."

When this message appears, press <PF1> to read the messages.

If there are no multi-line error messages, this help level is skipped and the second level is entered directly.

The second level of help is optional and is dependent on the attributes of the field for which help is being requested.

If the field has a list of all possible values associated with it, this level of help will display that list and a value may be selected.

If the field contains a library name, this level of help will display a list of the available library names that may be selected.

If the field contains a table name, this level of help will display a list of the available table names that may be selected. To move one of these values or library or table names into the field, use the <Tab> or <Newline> key to move to the desired entry, then insert the S (SELECT) line command to the left of the entry and press <Enter>, or, simply press <Enter> without the S.

The third level of help contains short explanations intended to assist a user when entering data on the screen or using a particular menu item.

The fourth level of help contains tutorial material intended for new or occasional users.

Typing HELP, or pressing <PF1> repeatedly, moves you through these levels in the appropriate order: level 1, level 2, level 3, and then level 4.

Using Help from the Command Line

You can specify what type of help is required by entering one of the options listed in Table 2-2 on the command line and pressing <PF1>.

Table 2-2: Help Options

Option	Displays
TABLE	Help on the current table
MENU	Help on the current menu
ABC	Help on rows whose keys start with “ABC'
’TABLE,ABC’	Help on the current table whose row keys start with ABC

If more than one parameter is required, the set of parameters must be enclosed in single or double quotes.

If you wish, you can also enter a complete command-line help command. For example, typing HELP, 'TABLE,ABC' in the command area and pressing the <Enter> key is equivalent to typing 'TABLE,ABC' and pressing <PF1>.

Help for Protected Fields

When performing certain functions in tablesONLINE, you will find that there are fields that are protected. These fields contain data that cannot be changed by the user. If you are in an **Identify Table** screen, for instance, and wish to move through the fields on the screen, you will find that the cursor skips over these fields, moving directly to the next field that can be edited.

Although a field may be skipped over, there may still be help available that describes the contents of the field. To view help for a protected field, use the up and down arrow keys to move the cursor to the protected field and press <PF1> for help.

Using PF Keys

Typing PF on the command line will display the PF keys that have been defined for the current screen and their respective functions. The default PF key assignments used in the command menus of tablesONLINE, as distributed defaults, are shown Figure 2-2.

```
tablesONLINE 6.0.2 Administrator ------ Menu ------------------- PFKS/COMMANDS
COMMAND ===>


Program Function Key Assignments:

PF 1 HELP                - If errors, display messages else Help/Tutorial
PF 2
PF 3 END                 - Return to Previous Screen
PF 4
PF 5
PF 6
PF 7 UP                  - n Menu Rows (Default = 12 or cursor position)
PF 8 DOWN                - n Menu Rows (Default = 12 or cursor position)
PF 9 WINDOW              - Next Window (- for previous, NEW to create one)
PF10
PF11
PF12 CANCEL              - Return to Previous Screen


Enter END to return to the previous screen.
Enter UP or DOWN to display the remaining function key assignments.
Enter X to suspend tablesONLINE and return to CICS.
```

Figure 2-2: PFKS/Commands for Menu Processing

PF key assignment within tablesONLINE is table-driven and may be different for each application or user. It is possible to assign any command that the application recognizes on the command line to a PF key.

Default PF Key Definitions

The default PF key definitions for tablesONLINE follow the TSO/ISPF standard, also widely used under VM. The six keys displayed on the screen above are the keys that are active at all times. Table 2-3 contains the delivered defaults for tablesONLINE PF keys.

Table 2-3: PF Key Definitions

PF Key	Key Name	Description
PF1	The HELP key	Displays various levels of help on successive presses - error messages, if applicable - field values list, library and table lists - help screens - tutorial information
PF2	The EXECUTE key	Executes the contents of the screen. - In Update mode, the row is updated - In New mode, a new row is created - In Utilities mode, the utility is invoked
PF3	The END key	Moves you back to the previous screen (One level up, if hierarchical menus are in use) Enters any changes made on the current screen
PF4	GET table row by COUNT	Default is next row if no count is given Count may be entered on the command line Unsigned numbers are absolute position in table Signed numbers are relative to current position Does not work with Hash tables
PF5	Find a named field	
PF6	GET table row by KEY (or reposition in table)	
PF7	Scroll UP	
PF8	Scroll DOWN	
PF9	The WINDOW key	Switches you between various tablesONLINE windows/sessions.
PF10	Scroll LEFT	
PF11	Scroll RIGHT	
PF12	The CANCEL key	Moves you back to the previous screen discarding any data entered on the current screen.
PF21	FREEZEKEYS	Locks left most columns in place so they do not scroll.

tablesONLINE supports up to 24 PF keys. In the distributed defaults, there are 24 PF key entries. Keys 1-12 and key 21 are as listed above with keys 13-24 being duplicates of the first 12. <PF13> is the same as <PF1>, <PF14> the same as <PF2> and so on. Because <PF21> is defined with a unique function, <PF9> has no duplicate.

Each tablesONLINE screen will have a different subset of PF keys active. Type PF in the command line of any screen to find out which keys are active. Pressing an inactive PF key will result in an error message.

Using Multiple Windows

The WINDOW command can create multiple windows and navigate from one window to another. This command is available from any screen in tablesONLINE. In the distributed defaults, <PF9>, the WINDOW key, is used for this purpose. Table 2-4 lists the parameters for this command

Table 2-4: Multiple Window Command Parameters

Command	Parameter	Short Version	Effects
WINDOW	NEW	NW or WN <Enter>	Create another window
WINDOW	-	W - <Enter>	Move to previous window
WINDOW	+ or blank	W<Enter>	Move to next window

To create a new window, insert NEW on the command line and press <PF9>.

To move from one window to the next, insert W in the command line and press <Enter> or simply press <PF9>. To move to the previous window, insert W- and press <Enter> or insert just a - and press <PF9>. When you reach the last window, another invocation causes you to move to the first window and vice versa. To terminate a window, enter END from the top-level menu.

Customizing the User Environment

tablesONLINE includes a User Profile table that maintains a user's environment. The initial set up of options in this table is performed by the system administrator. Following the set up, there are a number of options that can be set based on your preferences. These options include Confirm Save, Confirm Delete, and Suppress Information.

The Confirm Save option determines whether or not you are asked to confirm that you wish to save a table.

The Confirm Delete option determines whether or not you are asked to confirm that you wish to delete a table.

The Suppress Information option indicates whether or not information messages are displayed. Error and warning messages are always displayed regardless of the setting.

These options can be modified at any time by entering the command or short form listed in Table 2-5 on the command line.

Table 2-5: Suppress Information Command Options

Command	Short Form	Value	Effect if Value = Y
CONFIRMSAV	CS	Y or N	Confirm that you wish to save a table.
CONFIRMDEL	CD	Y or N	Confirm that you wish to delete a table.
SUPPRESSINFO	SI	Y or N	Indicate that you wish to suppress Information Messages.

Menu Screen

A menu screen provides a listing of selectable tablesONLINE actions. Figure 2-3 is a sample of the format that the menu screens will take.

```
tablesONLINE 6.0.2 Administrator ------ ------------ Administrator ------------
COMMAND ===>


To select, enter number/symbol on command line:

D    DEVELOP APPLICATION  - Create and Develop Table Applications
M    EDIT MRO TRAN IDS    - Add/Change Sets of tablesONLINE Transaction IDs
VTS  EDIT VTS ASSIGNMENT  - Add/Change/Delete DDname - VTS Assignments
1    EDIT APPL. CONTROL   - Add/Change/Delete Applications and/or Users
2    DELETE SESSIONS      - Delete Active tablesONLINE Sessions
3    COPY APPLICATION     - Copy an Application's Controlling Tables
4    EDIT USER PROFILES   - Add/Change/Delete Items on User Profile Table
5    EDIT HELP TABLES     - Edit tablesONLINE Help Tables
6    EDIT TUTORIAL TABLE  - Add/Change/Delete tablesONLINE Tutorial Table
7    EDIT X-AUTHORIZATION - Add/Change/Delete Cross Authorizations for Users
8    EDIT USER APPL TABLE - Edit the User/Application Relationship Table
9    EDIT CONSTANTS       - Add/Change Sets of tablesONLINE System Constants

Enter HELP at any stage for help within tablesONLINE.
Enter PF for program function key assignments.
Enter X to suspend tablesONLINE and return to CICS.
```

Figure 2-3: Administrator's Menu Screen

The following menus are delivered with tablesONLINE:

- Administrator's menu contains entries for all of the administrative functions that may be performed. This is the initial menu from which you will begin to tailor the system for your needs.
- Application Developer's menu contains entries for all of the functions required to create an application. This is accessed from the Administrator's menu.
- Utilities menu contains entries for all of the tablesONLINE utilities that may be performed. This is accessed from the Application Developer's menu.
- Define Table and View menu contains entries for creating and editing the format of tables. This is accessed from the Application Developer's menu.
- Help/Tutorial menu contains entries for editing all of the tablesONLINE Help tables. This is accessed from the Application Developer's menu.
- User Applications menu contains entries for all of the user applications that have been defined. This is accessed from the Application Developer's menu. The items in this menu are defined by your organization.

Menu Navigation

To move around a menu screen, use the <Tab>, <Shift-Tab> and <Newline> keys.

To select an item from a menu, enter its number or symbol on the command line and press <Enter> or move the cursor to the desired menu item and press <Enter>.

Some menus contain more options than can be displayed on the terminal screen at one time. In these cases, the menu screen will display a message indicating that there are additional entries above or below the displayed entries, or both. You can scroll up or down to see the additional entries using the <PF7> and <PF8> keys (representing up and down respectively). You can also type the commands UP and DOWN directly on the command line and press <Enter>.

Scrolling is sensitive to the cursor position. To scroll a specific number of lines, type the desired number of lines on the command line and press <PF7> to scroll up, or <PF8> to scroll down.

3
Editing and Browsing Tables

In the previous chapter we looked at navigating through the menu system. In this chapter we discuss how to navigate through the tablesONLINE Editor. Become familiar with the tablesONLINE Editor for editing and browsing, as it prepares you for the next chapter, Defining Tables, which uses the same navigation principles that are covered here.

In this chapter you will learn how to select a table to browse or edit and how to navigate through the table to find the row(s) you wish. Special attention will be paid to the line commands, which enable you to add, change, delete and move rows. This chapter concludes with a discussion of the options available when the editing of a table is complete.

Editing or browsing a table requires two components: a View and a Data Table. The View is a template that is placed on the Data Table and governs the column display and edits that can be performed on the data. Once you are familiar with using a View to browse, populate or edit a Data Table, Chapter 4 discusses how to create those Views and the Data Tables using the tablesONLINE editor. To perform editing or browsing you will use the first two options in the Application Developer's menu, A - EDIT TABLE, and B - BROWSE TABLE.

Selecting a Table to Browse or Edit

Whether you are editing or browsing a table, you select the table in the same way. Begin the selection process on the Application Developer's menu (Options A - EDIT TABLE and B - BROWSE TABLE). You will be presented with an **Identify Table/Row** screen (see Figure 3-1).

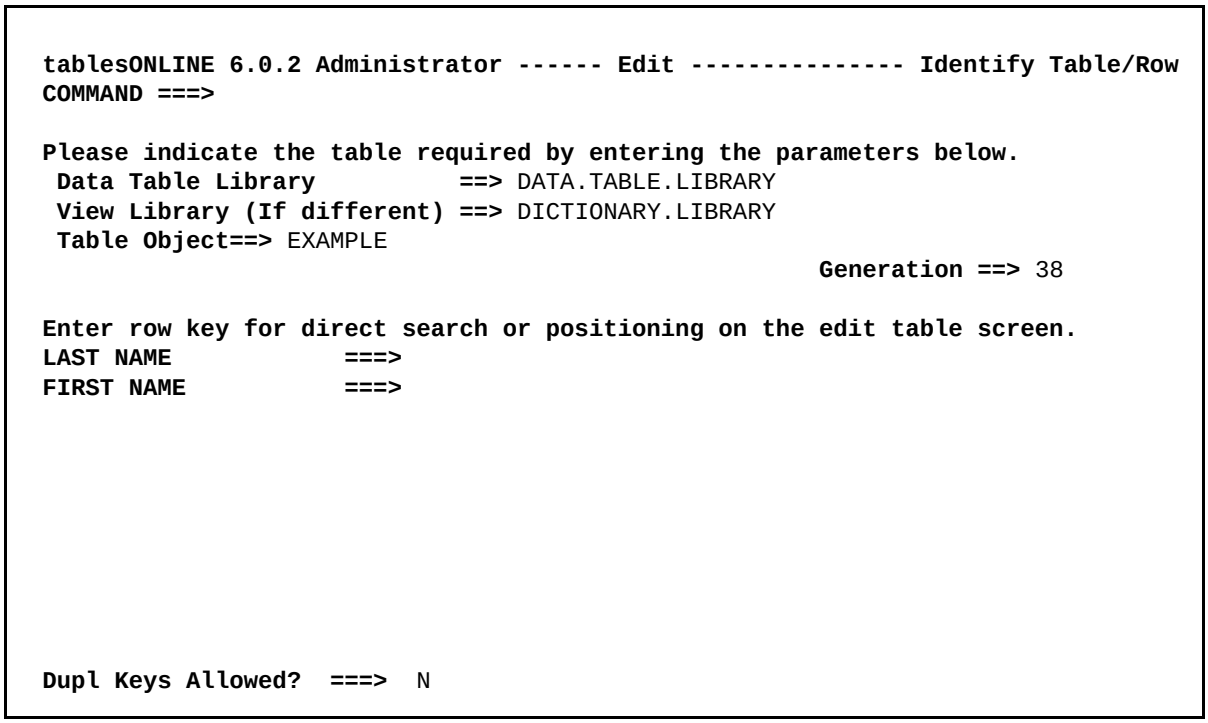

Figure 3-1: Identify Table/Row Screen

The display portion of the screen is divided into three sections. At the top of the screen, the table is identified using the Library Name, View library, Table Object Name and Generation. In the middle section, you would enter the key of the row you wish to see (assuming you know what the key is). At the bottom of the screen is a display indicating whether duplicate keys are allowed; this is an optional field whose presence is determined by the View.

You will notice that the fields in this screen already contain information about the last Table Object you worked on in tablesONLINE. If this is the Table Object that you wish to browse, press <Enter>. If you wish to browse or edit a different Table Object, you must modify the information contained in this screen as follows.

Specifying a Table Object

If you know and wish to specify the name of a Table Object in an **Identify** screen, you can simply type the name in the Table Object field.

It is not necessary to modify the Data Table Library field and the View Library field in order to specify a Table Object. If you modify the name in the Table Object field, the libraries containing the View and Data Table will automatically be inserted in the Data Table Library and the View Library fields when the Table Object is found.

If tablesONLINE cannot find a Table Object that you know exists, clear both the Table Library and View Library fields and press <ENTER>.

Table Lists

To display a list of all Table Objects in the specified library, clear the Table Object field and press <PF1>.

To see a list of the Table Objects in all of the authorized libraries, clear the Data Table Library field and the View Library field and press <PF1>, with the cursor in the Table Object field. This displays a list of all Table Objects.

To select a Table Object from the list, place the cursor beside the row and press <Enter> or use the S line command.

To limit the list of Table Objects, an asterisk can be used to replace the characters in the Table Object name. For example, T*B* will list all Table Objects whose name contains a T in the first position and a B in the third position.

Specifying a Data Table and a View

When you select a table to browse or edit, there must be both a Data Table and a View for that Data Table. tablesONLINE includes a facility known as Many-to-Many (M2M). This facility, described in the next chapter, allows for the definition of relationships between multiple Views and multiple Data Tables.

These relationship definitions are stored and referenced in a table. When a Data Table and View combination is defined in the M2M table, it is given a Table Object name. It is this name that is used in the **Identify Table/Row** screen (see Figure 3-1). This feature simplifies the referencing of tables by removing the need to know both the name of the Data Table and the appropriate View.

The delivered default is M2M. A Table Object name is requested in the **Identify Table/ Row** screen (see Figure 3-1) for both browsing and editing.

If M2M is turned off, the **Identify Table/Row** screen will not prompt you for a Table Object name. Instead it will ask for both the name of the Data Table that is to be browsed or edited and the name of the View that is to be used to access the Data Table. This is done only when a View/Data Table relationship has not been specified. It is the user's

responsibility to ensure that the View that is specified is compatible with the Data Table specified. For complete details of M2M processing, see "Define Many To Many (M2M)" on page 98.

Whether M2M can be turned on or off is governed by a setting in the Application Control Table that is set up by your tableBASE administrator. If an application has been set up so that M2M can be turned on and off, the following actions may be performed:

- to turn M2M off, type M2M OFF in the command line and press <Enter>.
- to turn M2M back on, type M2M in the command line and press <Enter>.

All other aspects of browse and edit remain the same whether M2M is on or off.

Identifying a Library

Throughout tablesONLINE you will use the **Identify Table/Row** screen (see Figure 3-1) to specify the name and location of a table to browse or edit. On these screens, in most cases, you will be asked to supply the name of the library that contains the table or View that you wish to use.

The most common way to specify the library name is with a List Facility. If you wish to see a list of either View or Data Table libraries, insert the cursor in the appropriate library field and press <PF1>. This will display a list of all libraries that the user is authorized to access.

To select a Library from the list, place the cursor beside the row and press <Enter> or use the S line command.

There are four other ways to specify a library name:

- insert a DDNAME
- insert a Dataset Name
- specify a library alias
- specify a number from 1-7

While a library can be identified by a DDNAME or Dataset Name, tablesONLINE provides two additional methods to specify a library. You can set up aliases for your tablesONLINE libraries that can then be used to reference that library. An alias is simply another name for a library that can be used in tablesONLINE more easily than the library Dataset Name. These aliases are stored in the table. For complete details for editing the Library Table see "Library Table" on page 163.

Or, you can identify a table in a Library Name field by inserting a number from 1-7 in the field. The number refers to the position of the library in the library list for the application as defined in the Application Control Table. When an application is initially set up by the tableBASE administrator, an entry is made in the Application Control Table. The Application Control Table includes seven fields that are used to identify the libraries that will be used by that application. The relative position of a library in that list can be used to identify a library in the Library Name field of the **Identify** screen.

Protected Library or Table Names

In some cases, when you use other menu selections to edit tables, the library and table name fields may be displayed for reference, but can be protected from change. This is controlled by the menu table row used to invoke the editor. This feature is discussed in detail in Chapter 6, Building Applications. If these fields can be altered, they will be displayed on the screen in red. If they cannot be altered, they will be displayed in white. In the event of a monochrome screen, when moving through fields using the <Tab> key, any fields that may not be altered will be skipped over.

Passwords

Both read and write passwords may be defined for tables in tablesONLINE. The password field will appear with a prompt on the **Identify Table/Row** screen for any table that has been assigned a password. If no password has been assigned to the table, the field and the prompt will not appear.

If a table has a read password, a password will be required to open the table for browsing or editing. If a table has a write password only, you will need the password for editing but not browsing.

Specifying Table Generations

Each of the **Identify Table/Row** screens will contain a reference to a table generation. This field is used to indicate the generation of the table that is to be browsed or edited.

tableBASE marks each generation of a table that is stored in a library with an absolute generation number from 1-255. The table definition specifies the number of generations to be kept, allowing a maximum of nine generations. Once the maximum number of generations has been reached, storing a new generation causes the oldest generation to be deleted from the library.

A generation can be specified in several ways:

- A positive integer refers to an absolute generation number.
- Generation zero refers to the current generation.
- A negative integer refers to a relative generation number.
- A blank refers to the current generation, which is the default.

For example, if generations 2, 4 and 5 are available, then generation 5 is the current generation.

You can specify either a relative (see Table 3-1) or an absolute (see Table 3-2) generation number, refer to:

Table 3-1: Relative Numbers

Using Relative Numbers	Indicates	Generation
0	5	current
-1	4	one before current
-2	2	two generations back

In the above example of three generations, any other negative integer results in a "Generation not found" message.

Table 3-2: Absolute Numbers

Using Absolute Numbers	Indicates
5	5
4	4
2	2

Any other positive integer results in a "Generation not found" message.

Note: Care must be taken when performing data recovery operations. If you have a corrupt generation, for example, generation 0, editing the previous generation (for example, -1) and saving it gives you a valid current generation. During the save, the previously current (and corrupt) generation (0) becomes the previous generation (-1). A safer course of action is to delete the corrupt generation with the tablesONLINE delete table utility.

Specifying a Row Key

If the information contained in the **Identify Table/Row** screen (see Figure 3-1) when you first enter is correct, the bottom of the screen will display field names that represent the key fields. If you have modified the information in this screen to select a different table to browse, you must press <Enter> to refresh the screen to display the key fields of the newly specified table.

In either of the cases described above, you can insert either the full key or a partial key for a row in the table you wish to edit. If you do not specify a key, the table will be opened for browse or edit with the first row in the table at the top of the screen.

Once you have completed all of the information in this screen, press <Enter> to display the **Edit/Browse Table** screen (see Figure 3-2).

If a row key is given and an exact match for it is found on the table, then the **Edit/ Browse Table** screen (see Figure 3-2) is skipped and the **Edit/Browse Row** screen (see Figure 3-3) is brought up immediately. Otherwise, the **Edit/Browse Table** screen with its multi-row display comes up so that you can select a row.

Duplicate Keys Control

The tableBASE administrator controls the permissions for entering rows with duplicate keys. This is done by allowing duplicates when defining the View for this table and the menu selection used to access this screen.

If both conditions prevail, the Duplicate Keys Allowed field will be present on the **Identify Table** screen (see Figure 3-1) and will contain the default setting.

If the default setting is Y, you may override it and prevent rows with duplicate keys to be added to the table by changing the setting to N. If the default setting is N, you may change it to Y.

Edit/Browse Table Screen

An **Edit/Browse Table** screen (see Figure 3-2) displays the rows of a table in a horizontal manner with field names displayed at the top of each column.

```
tablesONLINE 6.0.2 Administrator ------ Edit Table ---------------------------
COMMAND ===>

Table Name : EXAMPLE                                  Row Location : 1

             LAST            FIRST      DIVISION DEPARTMENT  SEX      DATE
             NAME             NAME                                     OF
_____ ____________________ _____________ ________ __________ _____ CONTRIBUTION
      ANCHRUTHER           DORA          ADMIN    SECT         F    19870626
      ASSIGNY              MICHEL        KAMIN    SECT         M    19870626
      AXOLOTLOVOVITCHSKI   STEFANIE      ADVTG    ISO9001      F    20080225
      BAKER                JOHN          ADVTG    NEWS         M    19870626
      BLOGGINS             JOHN          MIS      BANAL        M    19870626
      BROWN                FREDERICK     ADMIN    PERSNL       M    20000626
      BROWN                GEORGE        ACCTG    ISO9001      M    19870626
      BROWN                JOHN          OPNS     SCH          M    19870626
      BROWNBAG             FREDDA        ADMIN    PERSNL       F    19870626
      CALLAN               MICHAEL       MIS      PGMG         M    19870626
      DALE                 TERENCE       OPNS     OPR          M    19870626
      DUMAS                SAMANTHA      ACCTG    P/R          F    19870626
      FUTA                 MARY          OPNS     OPR          F    19870626
++++  GALSWORTHY           SARAH         ACCTG    A/R          F    19870626
```

Figure 3-2: Edit Table Screen/Browse Table Screen

If no key was entered, the screen appears with the first row of the table at the top, as shown in the example. If a partial key was entered, or the search for a full key failed, the same screen appears but the positioning in the table is different. In a sequential table, the row after which the searched-for key would reside is placed in the second position from the top of the screen.

In many cases, all the information in your table will not fit on the terminal screen. The rows may be longer than the screen width and there may be more rows than will fit vertically on the screen. In these situations you can use any of tablesONLINE flexible navigation facilities to locate the information you require.

Table Navigation

You may move through the rows in your table using either PF keys or commands.

PF keys allow you to scroll up, down, left, and right through the table.

Commands allow you to search through the table by specifying things such as the count, the full or partial field name, or the full or partial key of a row. It is also possible to freeze certain fields on the screen so that they do not scroll.

Note: The following descriptions assume that the PF keys, as delivered with tablesONLINE, have not been changed by you or your system administrator.

Scrolling

If the table you are editing is too large to fit vertically on the display screen of your terminal, you will see a series of plus signs (++++) at the left of your screen. These signs indicate that there is additional information that may be viewed. If the plus signs appear at the top of the list of rows, there is additional information above what is currently displayed. If the plus signs appear at the bottom of the list of rows, there is additional information below what is currently displayed.

To view these additional rows you must scroll through the table (see Table 3-3)

Table 3-3: Scroll Up/Down

PF Key	Function	Description
PF7	Scroll Up	Enter the number of rows you wish to scroll in the command line. If the command line is left blank, the display will scroll up one full screen.
PF8	Scroll Down	Enter the number of rows you wish to scroll in the command line. If the command line is left blank, the display will scroll down one full screen.

When you are in the **Edit/Browse Table** screen, not all the fields in a particular row(s) may fit horizontally on the screen. In this case, you may scroll left or right (see Table 3-4):

Table 3-4: Scroll Left/Right

PF Key	Function	Description
PF10	Scroll Left	Enter the number of columns that you wish to scroll in the command line. If the command line is left blank, the display will scroll to the left most field.
PF11	Scroll Right	Enter the number of columns that you wish to scroll in the command line. If the command line is left blank, the display will scroll one screen to the right.

Note that the number of columns that can be scrolled left and right when using these PF keys depends on the size and number of frozen fields in the table. If the frozen fields take up half of your screen, then the balance of the screen will be used to scroll the other fields in the table left and right.

To Limit Scrolling

In some cases, you may wish to scroll left and right through the data in your table while always having the first few columns of your table remain visible on the screen. To do this use <PF21> (see Table 3-5)

Table 3-5: Limit Scrolling

PF Key	Function	Description
PF21	FREEZEKEYS	Enter in the command line the number of columns to freeze, or, enter zero to unfreeze any frozen columns.

In the **Edit/Browse Table** screen (see Figure 3-2), the frozen columns will remain at the left of the screen. The balance of the columns will scroll left and right leaving the frozen columns displayed at all times.

In the **Edit/Browse Row** screen, the frozen fields will remain at the top of the screen. Here, the balance of the fields will scroll up and down leaving the frozen fields displayed at all times.

Table Search Commands

tablesONLINE offers you a variety of ways of moving to specific rows or columns in a table. You may use PF keys, commands, or a mixture of the two. Table 3-6 summarizes the possibilities.

Table 3-6: Table Search Commands

Function	PF Key	Entered in Command Line
GETCNT	4	Number of the row to be displayed.
FINDFLD	5	Full or partial name of column/field to display
GETKEY	6	Full or partial key of row to display
GETNEXT		GN
GETPREVIOUS		GP
GETFIRST		GF
GETLAST		GL

Depending on whether you are in the **Edit/Browse Table** screen (where multiple table rows are displayed) or in the **Edit/Browse Row** screen (where a single table row is displayed), the above commands will appear to have different effects.

Let's look at these options more closely.

GETCNT

The GETCNT table search command can be given as a line command, or by pressing the <PF4> key. The GETCNT command retrieves and displays the row that is identified by the count specified in the command line. If no count is specified, the next row in the table will be retrieved. An unsigned number represents the absolute table position of the row. A signed number represents the position relative to the current position in the table, negative to indicate up and positive to indicate down

In the **Edit/Browse Table** screen, this command will move the row to the **Edit/Browse Row** screen. In the **Edit/Browse Row** screen, this command will display the new row. For example, if a table has 100 rows and the last row fetched was number 60, then Table 3-7 shows some examples of the rows that will be returned based on the number inserted in the command line when <PF4> is pressed.

Table 3-7: Returned Rows

Initial Row	Argument	Resulting Row
60	None	61
60	10	10
60	+10	70
60	-10	50
60	70	70
60	+70	100
60	-70	1
60	150	100

A specified count that is out of bounds beyond the end of the table positions you at the last row in the table, while a specified count that is out of bounds before the beginning of the table positions you at the first row in the table.

FINDFLD

The FINDFLD table search command can be given as a line command, or by pressing the <P5> key. The FINDFLD command locates the first occurrence (relative to the current position in the table) of the particular field requested and displays it as the first field on the screen.

If you have frozen any fields, the found field will be positioned right after the last frozen field. In the **Edit/Browse Table** screen, FINDFLD will move the specified column into the first position to the right of any frozen columns. In the **Edit/Browse Row** screen, FINDFLD will move the specified field to the display line below any frozen fields.

Successive uses of FINDFLD will move you through any subsequent fields that match the criteria specified.

GETKEY

The GETKEY table search command can be given as a line command, or by pressing the <PF6> key. Using GETKEY in an **Edit/Browse Table** screen moves the row matching the specified key into the second position on the screen. In an **Edit/Browse Row** screen, using GETKEY displays the row with the matching key in the **Edit/Browse Row** screen.

When using GETKEY, you may enter several keys in the command line, separated by commas, with the whole list enclosed in quotes. tablesONLINE assembles these into a complete key as required by tableBASE. For example, if the key fields are a 20-character last name and a 20-character first name and "SMITH, JOHN" is given on the command line, then

```
'SMITH                JOHN        '
```

with the correct number of padding spaces, will be passed to tableBASE.

Partial keys may also be used with GETKEY as a way of positioning within a table rapidly. For example, "SMITH, JO" will likely locate John Smith. If, however, your company has a Jody Smith, this row will be located instead. This operation will probably position close enough to the desired entry that you can scroll through the table from this point to the required entry.

Note: Using an entry such as "S, JOHN" is unlikely to be effective. It will find the first entry in the table whose key begins with S.

tableBASE has adopted conventions as to where you will be positioned in the event that a key is not found. Your position in the table depends on the table organization. The rules listed in Table 3-8 apply in cases where a full key is specified but not found, and, when a partial key is specified.

Table 3-8: Table Rules

Table Organization	**Full Key Given But Not Found or Partial Key Given**
Sequential, or Descending	Position in the table where the row would be, if it existed on the table.
Random or User Ordered	End of the table.
Hash	The position where the hashing algorithm calculates the row would be, if it existed on the table.

Both the GETKEY and GETCNT commands, when used in the **Edit/Browse Row** screen, first call validation routines on the screen data for the current row. If data errors are detected, the current row will remain on the screen and an error message will be displayed. Any errors must be corrected before you can move to another row. If there are no errors in the data for the current row, the next row will be displayed.

Line Commands

Line commands are special commands designed to be used in the **Edit/Browse Table** screens. These commands allow you to perform operations on selected table rows. They allow you to specify not only which row you wish to edit, but also what edit operation you wish to perform. Line commands may be used by entering the letter code at the left edge of the screen next to the row you want to edit. Table 3-9 list the line command codes.

Table 3-9: Line Command Codes

Command	Description
S	Select row
U	Update row (edit only)
N	Create new Row using this one as a template (edit only)
I	Insert new row initialized with spaces/zeroes (edit only)
D	Delete row (edit only)
DD	Delete a series of rows (edit only)
Dn	Delete n rows (edit only); where n is the number of rows
M	Move a row
MM	Move a series of rows
Mn	Move n rows; where n is the number of rows
(blank)	Repeat previous line command (S, U or N) for row at the current cursor position

When you are creating a View, one of the options is whether you wish to edit key fields. You select this option using the Protect Keys Indicator. When this key is set to Y, the terminal user will not be allowed to edit key fields and the command will fail. Thus, if in the U, N, I or D commands you attempt to edit a key field and the indicator is on, the command will fail.

When you select a row with the S, U, N or I commands, the **Edit/Browse Row** screen appears (see Figure 3-3).

S Select Row

The Select Row (S) line command allows you to select a row for editing/browsing. When editing, using this command permits editing of all fields except key fields. This occurs regardless of the setting of the Protect Keys Indicator.

U Update Row

Like the Select Row command, the U line command allows you to select a row for editing. However, in this case, you may be permitted to edit key fields, depending on the setting of the Protect Keys Indicator.

If you skipped the **Edit/Browse Table** screen by entering a row key on the **Identify Table** screen, this has the same effect as using the U line command on the **Edit-Table** screen.

Once the U line command has been specified, it will remain in effect until you change it. This allows you to place the cursor anywhere on the row; simply pressing <Enter> will invoke the command. Similarly, if a row is highlighted, it will be selected using the U line Command if the cursor remains on the command line.

If U was used to select a row to edit, and you then decide you would like to create a new row using this one as a template, you can make the desired changes. Simply enter NEW on the command line and press <Enter>. If you then decide you wish to leave the original row unchanged, press <PF12> to cancel. If you wish to accept the changes made in the original row, press either the <PF2> key to EXECUTE, or the <PF3> key to EXIT.

N Create New Row

This command allows you to create a new row in the table using an existing row as a template.

The N line command reads in an existing row and allows it to be modified. The existing row is unchanged and a new row is created with the modified information. Once the N line command has been specified, it will remain in effect until you change it. This allows you to place the cursor anywhere on the row and press <Enter> to invoke the command. Similarly, if a row is highlighted, it will be selected using the N line Command if the cursor remains on the command line.

The new row will be placed in the table according to key sequence. If the key is a duplicate of an existing key, the new row will be placed before the row containing the matching key. If you are editing a User Ordered or Random table or editing a View, the new row will be placed in the table or View one position below the currently selected row. In a user ordered table, if you wish to indicate a specific position for the new row you can insert NEW, followed by the desired row location, on the command line. For example, NEW 10 will create a new row and place it in the tenth position in the table.

If N was used to select a row to be used as a template for a new row, and you notice that there is an update required to the existing row once it is retrieved, you can make the desired changes, enter UPDATE on the command line, and press <Enter>. The row will be updated and you will remain on the same screen where you can continue with the creation of the new row.

Note: If the Protect Keys Indicator is set to Y in the View, this operation will fail since this indicator prevents the creation of new rows. If the Duplicate Keys Allowed indicator in the View is set to N, or if the Duplicate Keys Allowed Indicator on the Identify Table/Row screen has been set to N, then any edit operation that would create a row with a key matching an existing row will fail.

I Insert New Row

This command allows you to create a new row in the table. The row is created with all the fields initialized appropriately — to the initial value, spaces, zeroes or the current date, depending on the field type — ready for input of new data.

It is also possible to create a new row from each of the tablesONLINE **Edit Table** screens by entering INSERT on the command line. This will create an empty row that can then be completed.

The new row will be placed in the table according to key sequence. If the key is a duplicate of an existing key, the new row will be placed before the row containing the matching key. If you are editing a User Ordered or Random table or editing a View, the new row will be placed in the table, or View, one position below the currently selected row. In a user ordered table, if you wish to indicate a specific position for the new row you can insert NEW, followed by the desired row location, on the command line (for example, NEW 10 will create a new row and place it in the tenth position in the table.)

Note: If the Protect Keys Indicator is set to Y in the View, this operation will fail since this indicator prevents the creation of new rows. If the Duplicate Keys Allowed Indicator in the View is set to N, or if the Duplicate Keys Allowed Indicator on the Identify Table/Row screen has been set to N, then any edit operation that would create a row with a key matching an existing row will fail.

Delete Rows

These line commands will allow you to delete a row(s) in the table. This operation will fail if the Protect Keys Indicator is set to Y in the View, since this indicator does not allow the deletion of rows from a table.

The three variations of the Delete Row Line command are listed in Table 3-10.

Table 3-10: Delete Row Line Commands

Command	Description
D	Delete a single row.
DD	Delete a series of rows enclosed by two DD markers. Insert DD to the left of the first row to be deleted and insert DD again to the left of the last row to be deleted.
Dn	Delete n rows beginning with the marked row; where n is the number of rows from 1-999.

Note: Confirmation for deletion may be required depending on the settings in the user's profile (see Edit Profile on page 114).

Move Rows

For any tables that have been set up with Random and User Ordered organizations, it is also possible to move rows within the table.

Table 3-11 lists the Move Row Line commands.

Table 3-11: Move Row Line Commands

Command	Description
M	Move this row.
A	Move to the position after this row.
B	Move to the position before this row.
MM	Move a series of rows enclosed by two markers; i.e., place MM to the left of the first row to be moved and place MM again to the left of the last row to be moved.
Mn	Move n rows beginning with the marked row; where n is the number of rows from 1-999.

The MOVE command requires that the row(s) being moved be marked with an M (or MM or Mn) and that the destination be specified with either an A or a B. The line(s) that is being moved will be placed after a row labeled A or before a row labeled B. If no such label is given or if more than one is entered, an error message is given.

Once these commands have been entered, press <Enter> to complete the operation.

Repeat Previous Line Command

If you have executed the S, U or N line command and wish to repeat the same command for another row in your table, move the cursor to the desired row and press <Enter>. It is not necessary to repeat the actual line command.

Canceling Line Commands

To cancel any of the line commands, either use the <Delete> key to remove the line command, type over it with spaces or use <PF12> to clear all line commands.

If there are no line commands to cancel and <PF12> is pressed, you will return to the **Identify Table** screen.

Edit/Browse Row Screen

An **Edit/Browse Row** screen is used to edit the data in a row. This screen is displayed once a row has been selected with the line commands S, U**,** N or I on the **Edit/Browse Table** screen.

The **Edit/Browse Row** screen displays one individual row in a vertical manner. The field names now appear to the left of the row data. Once you have selected a row for editing and it has been displayed in the **Edit/Browse Row** screen, you can move one by one through the rows in the table using the <PF4> (GETCNT) key without specifying a count. More on navigation between rows can be found later in this chapter (see "Navigating to Other Rows" on page 37).

In the example (see Figure 3-3), all of the table fields can be displayed on a single screen. For a table containing more fields, the first eighteen are displayed with a row of plus signs (++++) next to the last display line, indicating that there are additional fields below.

To view these additional fields, use the <PF7> scrolling key to scroll up the page or the <PF8> key to scroll down the page.

```
tablesONLINE 6.0.2 Administrator ------ Edit Row -----------------------------
COMMAND ===>

Table Name : EXAMPLE        Update Mode               Row Location : 4


     LAST NAME            : BAKER
     FIRST NAME           : JOHN
     DIVISION             : NEWS
     DEPARTMENT           : ADVTG
     SEX                  : M
     DATE OF CONTRIBUTION : 19870626
     CHARITABLE DONATION  : 500.00
```

Figure 3-3: Edit/Browse Row Screen

To move to a specific field, enter the field name on the command line (this may be a full or partial field name) and press <PF5> (to find the named field). Successive presses of <PF5> move you sequentially through the fields whose names begin with the entry that was placed on the command line.

Modifying Data in a Field

To modify the data in any field, you can either type over the existing data or clear the data from the field and enter new data into the empty field. If the Protect Keys Indicator in the View definition is set to Y, then key fields will be protected and cannot be updated.

Data Validation

tablesONLINE has built-in validation for each of its field types, including date fields. Data validation is applied to all fields displayed. In addition to validation based on field type, tablesONLINE offers other forms of data validation.

Edit Pattern Validation

Any alphanumeric field (Display Format X) may require that data conform to a specific pattern. For example, the edit pattern may specify that the first character must be numeric, the second character must be alphabetic and the third character must be the letter Q. If the entered data does not conform to the required pattern, a message will be displayed indicating the correct pattern.

Display Mask

A Display Mask may also be specified for a field. If this is the case, data can be entered in one of two ways:

- The field is cleared of any existing data and only data (i.e., no mask characters) is entered. The data, if valid, is displayed again with the Display Mask.
- Data is entered with the Display Mask in place. In this case, care must be taken not to modify the Display Mask. If it is inadvertently modified, it must be corrected or stripped out; failing this, an error will occur.

For more information about Display Masks, see "Display Mask" on page 57.

Initial Value

If the field has an initial value set, then it will appear as the default value when a new row is created. Unless the field is protected, typing over or clearing the entry can override the Initial Value.

Edit Highest Value
Edit Lowest Value

If the Edit Highest Value and Edit Lowest Value fields have been set, the data entered will be validated against an upper and lower bound. If a data outside the bounds is entered, the user receives an error message indicating the limit of the exceeded bound.

Searching Other Tables

Secondary tables may be searched to ensure that the input data exists, does not exist, or is within a range of values. A message is displayed if the search fails, allowing the data to be re-entered. For this type of validation you can use the <PF1> help key to display a list of values that may be selected.

Field Exit Programs

tablesONLINE can also call field exit programs for additional application-specific validation. These exit programs are invoked whenever the fields for which they have been defined are edited.

Exit programs can also be defined and called at points above the field level such as when rows are input and output, for cross-field validation within a row, and at table open, close, and store time.

For detailed information on the creation and use of user exit programs in tablesONLINE/CICS, see the tablesONLINE/CICS Exit Programming Chapter of the tableBASE Programming Guide.

Invalid Data Handling

If invalid data is entered in a field during editing, you will not be able to leave the current edit screen. The incorrect field remains on the screen so that it can be corrected. If there is invalid data in the table when tablesONLINE loads it (this can occur only if the table was not created by tablesONLINE or if field definitions were changed between storing data with tablesONLINE and reading it), then the field containing the invalid data is displayed with a row of asterisks in the data area.

In either of these cases, an error code appears to the left of the field name label and a message will appear at the top of the screen. The error code has two parts: a letter to indicate the type of error, and a number used to match errors on this screen with corresponding messages on the **Help** screen. The possible error message codes are:

- E - Error
- W - Warning
- I - Information
- A - Abend

tablesONLINE requires that you correct any errors in edited fields before any additional input will be accepted. The UP, DOWN and FINDFLD commands will not work until all of the fields currently displayed contain valid data. Press <Enter> to check the validity of current screen data without attempting to scroll.

Navigating to Other Rows

The following subparagraphs identify some different methods to move directly to rows in a table, all available from the **Edit/Browse Row** screen.

Leaving a Row

There are several ways to leave the **Edit/Browse Row** screen. They are listed in Table 3-12.

Table 3-12: Leaving Edit/Browse Row Screen Commands

Key	Command	Effect
PF3	END	Update/create row and return to the previous screen
PF4	GETCNT	Update/create row and get new row by count Unsigned number - absolute position Signed number - relative to current row No count (default) - next row Not available for Hash tables

Table 3-12: Leaving Edit/Browse Row Screen Commands (Continued)

Key	Command	Effect
PF6	GETKEY	Update/create row and get new row by key Default if no key - row with lowest key Default if no match - next greatest row
PF12	CANCEL	Quit without updating/creating new row CANCEL does not check the validity of edited data, it discards any entered data and returns to the previous screen

You may also enter any of the following in the command line: GF (GETFIRST), GL (GETLAST), GN (GETCNT) or GP (GETCNT -1).

If any of the commands that update the row are called while there is invalid data in an edited field, the row is left unchanged and the row remains on the screen.

If one of the GET commands is called with valid data in all edited fields but with a count or key that cannot be found, then the row is updated, an error message is issued and the row remains on the screen.

Moving to Another Row

Typically, you do not return to the **Edit/Browse Table** screen after every row browse or edit, but instead move through a series of rows either sequentially using <PF4> (GETCNT) or by key using <PF6> (GETKEY). This will have a different effect depending on which line command was used to select a row.

If N or I was used, a valid new row must be created before you can go on to the next row. If U was chosen and the key is modified, you will still go to the next row (even though the row you have edited may have moved to a different location). For Hash tables, "next row" is meaningless, so the <PF4> key will have no effect and the same row will continue to be displayed.

Execute Command

Entering EXECUTE on the command line or pressing <PF2> can be beneficial when creating a series of new rows, when the N line command was used to select a new row. It will save a few keystrokes by not having to re-select the row on the **Edit/Browse Table** screen or entering the NEW command line for each new row.

Delete Command

Entering DELETE on the command line and pressing one of the keys listed in Column 2 in Table 3-13 will also exit a row. Each of these keys has a slightly different effect on exit.

Table 3-13: Delete Commands

Type in command line	Key	Command	Effect
DELETE	<PF2>	EXECUTE	Delete row, and if a Hash table or at the end of the table, return to previous screen. Otherwise, present next row.
DELETE	<PF3>	END	Delete row and return to previous screen.
DELETE	<PF12>	CANCEL	Quit without deleting and return to previous screen.
DELETE	<ENTER>	DELETE	Delete row, if a Hash table or at the end of the table, return to previous screen. Otherwise, present next row.

Leaving the Editor

When you have finished editing individual rows, press <PF3> or type END on the command line to leave the row editor. Depending on how you entered the **Edit/Browse Row** screen, you will return to either the **Edit/Browse Table** screen or the **Identify** screen (see Figure 3-4). Pressing <PF3> or typing END repeatedly returns you to the tablesONLINE menu where you selected the Editor.


```
tablesONLINE 6.0.2 Administrator ------ Identify Screen -------------- MESSAGES
COMMAND ===>

Messages generated processing the table:

TB-5052W Changes have been made to table 'EXAMPLE '.

         To confirm that you indeed wish to save the changes and create a new
         generation:
                             Select 'ENTER' or 'END'

         To cancel the changes that have been made:

                             Select 'CANCEL'.

         To go back to editing without saving or cancelling

                             Enter 'RESHOW'.
```

Figure 3-4: Confirm Save Table Screen

Depending on the setting of the CONFIRMSAV indicator in your User Profile, the **Identify** screen (see Figure 3-4) may be displayed so that you can confirm that you wish to save the changes you have made to the table. If the indicator is set to N, this screen is not displayed and a new table generation is saved when you exit the Editor with END or <PF3>.

4
Defining Tables

Table-defining information is maintained in two forms, physical information and row format information. Physical information, such as table organization, search method, and row size, is required by tableBASE for its data management functions. tablesONLINE operations require that the system has row format information about the format of the rows of the tables on which it operates, in effect a data dictionary. This chapter illustrates how to create and maintain both forms of table-defining information using tablesONLINE.

Note: Chapter 3 contains information on how to use the tablesONLINE editor and is a prerequisite for this chapter.

Overview

Both tableBASE and tablesONLINE need to know the basic physical attributes, such as table organization, search method, and row size, of a table. In addition, tablesONLINE requires more detailed information in order to allow you to browse and edit table rows online.

This detailed information, which is stored in a View, includes information about:

- where each field of data is positioned within a row
- whether to interpret data as alphanumeric characters, a binary integer, or more complex data types, such as a date
- where to position data on the screen
- the name of each field

tablesONLINE sees tables in pairs. Each table consists of a Data Table and a View containing the data definitions for the table. One Data Table may have several Views, each of which defines a different layout of the underlying data.

Views are used to support data validation during data entry and, later, browsing of table data. High volume processing, whether transaction or batch oriented, typically ignores any Views that may exist and uses only the Data Tables. It is possible, however, to code

your own applications to take advantage of the inherent capabilities of Views. For example, if range checking or other validation of fields within a row is required at certain points in the application, the View can provide all the field details needed to do this. The table-defining options of tablesONLINE allow the user to:

- define tables - by creating tableBASE table definitions
- define rows - by creating tablesONLINE Views

Naming Tables

While tableBASE will accept almost any eight bytes as a table name this is not the case for tablesONLINE. Table 4-1 lists the case naming conventions that must be followed if the table is to be used in tablesONLINE.

Table 4-1: Case Naming Convention

Byte	Acceptable Values
1	Uppercase alphabetic, numeric
2-8	Uppercase alphabetic, numeric or blank

Tables with Multiple Row Layouts

It is possible to create a table using tablesONLINE that contains more than one type of row. When several row types exist in a single table, a row identifier field within each row is used to identify a different View for each row type. This allows rows with different layouts to be properly displayed.

In order to make use of this feature, you must include a field in your Data Table when it is created. This field, known as a Dynamic View Suffix, will contain the indicator for each row, specifying the View to be used to display that row.

For additional information on this feature, see the description of "Key Field Indicator" on page 50.

Table-Defining Information

One of the principal functions of tablesONLINE is the creation of tables that can subsequently be used by tableBASE. The Application Developer's menu contains an entry, D - DEFINE TABLE pointing to the Define Table and View menu (see Figure 4-1) that contains all of the options necessary to create and edit tablesONLINE tables.

In the Define Table and View menu, there are four options used to create a table:

1. Define View - describes the row layout of a Data Table
2. Define View Supplementary - describes table access controls
3. Define Data Table - describes the physical table attributes
4. Edit Display Order - allows you to change the order in which table fields are displayed to the user

As mentioned earlier, a tablesONLINE table is actually made up of two separate tables. The first is the Data Table and the second is the View of that Data Table. The Data Table is defined by assigning the physical table attributes. If a table is to be used exclusively by tableBASE, only these physical table attributes need be defined. However, to browse or edit the table using tablesONLINE, you must also provide the row layout information. This information is stored in the table View. The View indicates to tablesONLINE how the information in the Data Table is to be displayed for editing and browsing purposes.

The table access controls and field display order provide additional control over the table. This information is not mandatory for either tableBASE or tablesONLINE, but can be used to refine the display of data for users.

Define View

This option allows you to define the fields within a row of a Data Table. The following items illustrate the type of information that can be specified:

- Field size and format for display (it is also possible to have fields that do not display).
- Field size and format for storage on table. Fields that do not exist on the Data Table. may be displayed, like comment fields.
- Field-level editing and validation information.
- Physical field order, which is identical to definition order.

This information is specified using the Define View option in the Define Table and View menu. The information provided is stored in a View. The View contains one row for each column (field) defined in the table.

Define View Supplementary

This option allows you to define the edit and display controls for the table. The following items illustrate the type of information that can be specified:

- row-level validation or security information
- table-level validation or security information
- Help table to provide assistance to users editing or browsing fields (table-specific help is searched before application help)
- the number of fields that should be frozen and remain visible when the table columns are scrolled left or right (may be changed by the user)

This information is defined using the Define View Supplementary option in the Define Table and View menu.

Define Data Table

This option allows you to define the physical characteristics of the table. The following items illustrate the type of information that can be specified:

- organization
- search method
- row size
- key location and size
- number of generations to be kept

This information is defined using the Define Data Table option in the Define Table and View menu. The information is stored with a Data Table in the tableBASE library.

Field Display Order

Use this option to define the order in which non-key fields in a table are displayed for a user when editing and browsing table rows. Normally, the fields in a table will be stored in the order in which they were created. The Field Display Order option allows you to modify that order to display the fields in the manner most appropriate for a given user. Multiple Views may be created for one table, each one with a different display order based on the user's requirements. Fields that are added to the table at a later date, using the restructuring facility described later in this chapter, may be displayed in any position you wish. The field display order does not affect storage order.

Table Defining Options

This section describes the options that you will require to create tables within tablesONLINE. These options are all contained in the Define Table and View menu (see Figure 4-1).

```
tablesONLINE 6.0.2 Administrator ------ ----- Define Table and View -----------
COMMAND ===>


To select, enter number/symbol on command line:

D    DEFINE ALL           - Define All Table Elements (View and Data)
P    PRINT VIEW           - Submit a Batch Job to Print a View
1    DEFINE VIEW          - Define the Fields in a Table's View
2    DEFINE VIEW SUPPLMT  - Define Supplementary Information for View
3    DEFINE DATA TABLE    - Define Data Table Definition (DT Block)
4    EDIT DISPLAY ORDER   - Edit the Order of Fields in View for Display
5    BROWSE VIEW          - Browse Field Descriptions in View
6    CREATE ALTERNATE     - Create/Edit Alternate Index for Data Table
7    RESTRUCTURE TABLE    - Restructure Data Table (After Updating View)
8    GENERATE COPYBOOK    - Submit a Batch Job to Create a Cobol Copybook
8C   GENERATE C STRUCT    - Submit a Batch Job to Create a C Struct from View
9    DEFINE M2M           - Assign Object Names to View and Data Combinations

Enter HELP at any stage for help within tablesONLINE.
Enter PF for program function key assignments.
Enter X to suspend tablesONLINE and return to CICS.
```

Figure 4-1: Define Table and View Menu Screen

Options 1, 2, 3 and 4 of this menu allow you to specify the types of information described in the previous section. While these options may be selected in any order when editing your table definition, there is a preferred order when creating your table definition as described in the following paragraph.

Option D - DEFINE ALL, has been provided to simplify this process. It includes options 1 - DEFINE VIEW, 2 - DEFINE VIEW SUPPLMT and 3 - DEFINE DATA TABLE. When you select Option D, you will be asked to define the fields that will make up your table rows. When you create a table using this option, some of the information you provide will be used by tablesONLINE in later options. Information such as row size will be calculated from the fields you define in the first stage. These calculations will be done for you if you select Option D or if you select Option 1 as your first step. Selecting Options 2 or 3 individually, without first completing Option 1, requires you to do these calculations manually.

Once your table has been created, you may wish to modify the order in which the fields are displayed. Option 4 - EDIT DISPLAY ORDER allows you to edit the display order of table fields. This does not affect the order in which table fields are stored, it simply modifies the order in which they are displayed.

Option 5 - BROWSE VIEW allows you to browse the contents of the View. This provides read-only access to the information without the ability to modify it.

In some cases, you may wish to access a table using different fields as key fields in the table. This requires the definition of an Alternate Index for the table. Option 6 - CREATE ALTERNATE allows you to create an Alternate Index for this purpose.

When making changes to a View using Option 1 from the Define Table menu, tablesONLINE will check to see if you are removing, resizing, adding, or repositioning fields in the View. It will also check to see if there is a Data Table associated with the View being modified. If a Data Table does exist, Option 7 - RESTRUCTURE TABLE allows you to restructure the Data Table to accommodate the changes made to the View.

Option 8 - GENERATE COPYBOOK allows you to generate a COBOL copybook for a table View. This copybook can then be used in an application program.

Option 8C - GENERATE C STRUCT allows you to generate a C structure from a table View. This structure can then be used in a C or C++ application.

Option 9 - DEFINE M2M will allow you to define Data Table/View relationships that are identified by a unique name.

Option P - PRINT VIEW to submit a batch job which will print a formatted report of the View specified.

Each of these options is described in detail in the following sections.

Define View

The first step necessary to create a tablesONLINE table is to create a View. This may be done in one of two ways: define a new View using the Define View option, or, copy an existing View.

Create a New View

When you choose to define a new View, the first screen displayed when you select the Define View option on the Main menu allows you to specify the name of the new View and the library where you want that View to be stored (see Figure 4-2).

To create a new View, insert the name you wish to assign to the View in the View Name field and the name of the library where you wish that View to be stored in the View Library field and press <Enter>. Both the View name and the library name will initially be set to default values based on the last table and library you used in tablesONLINE. If

the information in these fields is correct, leave it as it is; if not, make the necessary changes to the fields.

If you blank out the Library Name and press <Enter>, the default library for Views will be inserted in the View Library field. If you move the cursor to the View Library field and press <PF1> for help, a list of available View Libraries will be displayed. From this list, you can select a library with the S (SELECT) line command to move that library name into the View Library field. The generation field will automatically be set to zero (0) for a new View.

```
tablesONLINE 6.0.2 Administrator ------ Edit View ---------- Table's Row Layout
COMMAND ===>

Please indicate the view required by entering the parameters below.
 View Library (TBDICLB) =======> DATA.TABLE.LIBRARY

 View Name   ==>
                                                    Generation ==> 0
```

Figure 4-2: Identify a View Screen

tablesONLINE checks to ensure that a View with this name does not already exist in the library and asks you to confirm that you wish to create a new View. Do so by inserting NEW in the command line and pressing <Enter>.

tablesONLINE moves you to the **Edit-Row** screen (see Figure 4-3) where the fields that will make up the table can be defined. The next section, “Define Fields in a View” on page 48, describes how to define table fields.

Copy an Existing View

Instead of creating a new View, you may wish to copy and modify an existing one (see “Copy View” on page 109).

When creating a new View for an existing table you must ensure that the data currently in the table fits the new definition.

Once an existing View has been copied, you may wish to modify some of the defining information. This is done using the same options you would use to create a new View. The table defining Options, 1 - DEFINE VIEW, 2 - DEFINE VIEW SUPPLMT and 4 - EDIT DISPLAY ORDER, can all be used to modify an existing View as well as to define a new View.

Edit a View

If you wish to edit an existing View, select Option 1 - DEFINE VIEW from the Define Table and View menu (see Figure 4-1) and press <Enter>. The system displays the Edit View screen (see Figure 4-2). Here you can specify the View that you wish to edit. Insert the View Library and the View name and press <Enter> to display the fields that are currently defined for the table.

To modify an existing field, use the S (Select) or blank line command and press <Enter>. This displays the row in the **Edit-Row** screen where you may proceed to edit the information for that field.

To add a new field to the table, use either the N (NEW) or I (INSERT) line command and press <Enter>. Using N (NEW) creates a new field using the selected row as a template. The new field will be placed in the table in the position following the row that was used as a template. Using I (INSERT) will create a new field with all of the defining attributes initialized.

Defining fields in the View and the values that may be used in those fields are described in the following sections.

Define Fields in a View

Some of the fields in this screen (see Figure 4-3) should contain information, while others may be left blank. The fields that should contain information are the Key Field Indicator, Display Length and Display Format. If these fields are left blank, default values (N for KEY INDICATOR, 1 for DISPLAY LENGTH. X for DISPLAY FORMAT) will be inserted in these fields.


```
tablesONLINE 6.0.2 Administrator ------ Edit Field Definitions for a Row ------
COMMAND ===>

View Name : EXAMPLE          Update mode         Field Position : 7

     FIELD NAME           : DATE OF CONTRIBUTION
     KEY FLD IND          : N

     DISPL LENGTH (DATA)  : 8
     DISPL FORMAT         : A
     DISPL ATTR           :
     DISPL FEAT           :

     DISPL MASK LENGTH      10 .|....1....|....2....|....3....|....4....|....5
     DISPL MASK           : !!!! !! !!

     FIELD LENGTH         : 8
     FIELD FORMAT         : A
     INITIAL VALUE        :

     EDIT PATTERN LENGTH  : 0  .|....1....|....2....|....3....|....4....|....5
     EDIT PATTERN         :
     EDIT HIGHEST VALUE   :
     EDIT LOWEST VALUE    :

     ACTION               :
     SOURCE VIEW /TABLE   :
     SOURCE FIELD NAME    :
     TRIGGER FIELD NAME   :

     FIELD EXIT NAME      :
     FIELD EXIT VERSION   :    MANDATORY if the program is not current release
     FIELD EXIT INPUT IND :
     FIELD EXIT OUTPT IND :

     FIELD NAME EXTATT    : 00000000
     DATA DISPL EXTATT    : 00000000
```

Figure 4-3: Define/Edit a Field Definition Screen

The following section describes each of the field defining items in detail. Fields must be defined one at a time. Once you have completed all of the information for a particular field, press <PF2> to save that field. Repeat this process until you have defined all fields for your table and then exit the **Edit-Row** screen by pressing <End>.

Field Name

This field contains the name of the table column being defined. It may be left blank, which means no name will appear as a heading for that column in the table. The field name may be up to 20 characters in length.

Key Field Indicator

The Key Field Indicator field designates whether a field is a key field and/or used as the Dynamic View Suffix. In addition, the Key Field Indicator is used in enforcing the Data Table key when using an Alternate Index (see "Enforcing Unique Data Table Keys with an Alternate Index" on page 91).

There are eight values for the Key Field Indicator (see Table 4-2). The last four are used exclusively for a View of an Alternate Index.

Table 4-2: Key Field Indicator Values

Value	Description
N or (blank)	Indicates that the field is not a key.
Y	Indicates that the field is a key.
S	Indicates that the field is to be used as the Dynamic View Suffix.
B	Indicates that the field is both a key, and is to be used as the Dynamic View Suffix.
P	Indicates that the field is a Data Table key of a View for an Alternate Index.
O	Indicates that the field is a Data Table key and a key of the Alternate Index of a View for an Alternate Index.
Q	Indicates that the field is both a Data Table key, and is to be used as the Dynamic View Suffix of a View for an Alternate Index.
A	Indicates that the field is a Data Table key, a key of the Alternate Index, and, is to be used as the Dynamic View Suffix of a View for an Alternate Index.

The default for the Key Field Indicator is blank (indicating that the field is not a key).

Note: It is possible to define up to 50 key fields in a table. However, all key fields must be contiguous and may not exceed 256 bytes combined. The set of contiguous fields that make up the key can be located anywhere in the row and do not necessarily have to begin with the first byte.

Dynamic View Suffix

In certain cases, your Data Table may be made up of a number of different types of rows. In order to indicate that one row has a different layout than another row, an identifier field must exist. This row identifier is referred to as the Dynamic View Suffix.

The identifier, which may be up to eight bytes in length, forms a suffix that will be used by tablesONLINE to overlay the rightmost portion of the View name. When the table is subsequently processed, tableBASE will examine the identifier for each row to determine which View should be used for that row. Since the identifier may be eight bytes long, it may completely override the View name.

For example, an identifier may be two bytes in length and contain the values 00 - 11, indicating 12 possible row layouts. The row identifier dynamically becomes the suffix of the View. Assume the View name is TB068A00. Prior to processing a row with a suffix of 03, tablesONLINE overlays the last two bytes of the View name and searches for a View with the name TB068A03. In order to process all 12 different row layouts, 12 Views must exist. These Views would be named TB068A00 through TB068A11. Although the rows may be mixed row types, all rows within one table must have the same row length.

Only one field may be designated as a Dynamic View Suffix within a View. As a result only one of the values S, B, Q and A may be used within a View.

Display Format
Field Format

These two formats are interrelated and determine how the data is to be displayed when editing or browsing (Display Format) and how the data is to be stored (Field Format).

The following lists the valid values and valid combinations of Display Format and Field Format. The alphanumeric, numeric, and date Display Formats can be combined with any of the corresponding alphanumeric, numeric, and date Field Formats (as indicated in Table 4-3)

Table 4-3: Display/Field Format Combinations

Display Format	Displayed As	Field Format	Stored As
X	Alphanumeric	X	Entered
		U	Uppercase
Y	Hexadecimal	X	Binary
N	Numeric	N	Zoned decimal
0	Leading zeroes	P	Packed decimal
1	1 decimal place	F	Binary fullword
2	2 decimal places	H	Binary halfword
:	:		
9	9 decimal places		
A	yymmdd/yyyymmdd	A	Dates are
B	mmddyy/mmddyyyy	B	edited and
C	ddmmyy/ddmmyyyy	C	converted to
D	yyddd/yyyyddd	D	Field Format.
E	ddmmmyy/ddmmmyyyy	E	

If Field Format is omitted, Table 4-4 indicates the default value to which it will be set.

Table 4-4: Display/Field Format Default Combinations

Display Format Entered	Default Field Format	Default Rule
X	U	Translate lowercase alphabetics to uppercase.
Y	X	Displayed/entered hexadecimal digits into binary.
N,0,1,2,...9	N	Store numbers as zoned decimal.
A	A	Store dates in format used on input; if LENGTHs are not entered, they will default to the longer lengths (for example, year represented as yyyy).
B	B	
C	C	
D	D	
E	E	

If neither Display Format nor Field Format is entered, they will default to X and U, respectively.

Display Length
Field Length

The Display Length determines the length of the field on the screen when editing the Data Table. The Field Length determines the actual number of bytes that this field will occupy when stored on disk.

These two options are interrelated and dependent on the Display Format and Field Format. The relationship between the Display Length and Field Length depends on what the associated values for the Display Format and Field Format. The following table shows the how these two options are related, as well as the minimum and maximum lengths allowed.

Table 4-5: Display/Field Length Combinations

Display Format	Display Length	Field Format	Field Length
X	1 - 50	X	Same as display
		U	Same as display
Y	2 - 50 (even lengths)	X	1 - 25
N	1 - 17 (including sign)	N	display digits
0	1 - 16 (no sign)	P	(display digits+1)/2
1	2 - 18 (including sign and decimal point)	F	4 (display length limited to 9 digits)
:		H	2 (display length limited to 5 digits)
9	10 - 18 (including sign and decimal point)		
A	6/8 yymmdd/yyyymmdd	A	6/8
B	6/8 mmddyy/mmddyyyy	B	6/8
C	6/8 ddmmyy/ddmmyyyy	C	6/8
D	5/7 yyddd/yyyyddd	D	5/7
E	7/9 ddmmmyy/ ddmmmyyyy	E	7/9

If neither Display Length nor Field Length is defined, the values of each will default to 1.

Because the Display Format is set within the View, it is possible to create different layouts of your Data Table for different users by creating multiple Views, each containing a different Display Format. It is possible, for example, to show a date to one group of users in format B, while the same date is shown to another group of users in format C.

Display Attributes

This field controls how the data from the field is to be displayed and whether user input will be accepted into the field when editing the table. The possible values are listed in Table 4-6:

Table 4-6: Display Attribute Values

Value	Display Field Data	Allow Edits	Usage
(blank)	Yes	Yes	Default, normal fields.
N	No	No	Suppress the display of the field and headings.
C	No	No	Field name displayed as a comment or blank line. Data is neither stored nor displayed.
F	No	No	Filler field. This field will be initialized according to type.
D	No	Yes	Dark field for password entry, suppressed on **Edit/Browse Table** screen.
Q	Yes	Yes	Dark and Mandatory field for password entry, suppressed on **Edit/Browse Table** screen. Data must be entered if the row is newly created.
S	No	No	Suppress on **Edit/Browse Table** screen.
P	Yes	No	Protected field. Displayed on both **Edit/ Browse Table** screen and **Edit/ Browse Row** screen.
p	Yes	No	Protected field. Suppressed on the **Edit/ Browse Table** screen and displayed on the **Edit/Browse Row** screen.
M	Yes	Yes	Mandatory field. Data must be entered if the row is newly created. Displayed on both **Edit/Browse Table** screen and **Edit/Browse Row** screen.
m	Yes	Yes	Mandatory field. Must be entered if the row is newly created. Suppressed on the **Edit/Browse Table** screen and displayed on the **Edit/Browse Row** screen.

Table 4-6: Display Attribute Values (Continued)

Value	Display Field Data	Allow Edits	Usage
V	Yes	Yes	Verification field. Verifies numeric data is non-zero, character data is non-blank and date fields are non-blank. Displayed on both **Edit/Browse Table** screen and **Edit/Browse Row** screen.
v	Yes	Yes	Verification field. Verifies numeric data is non-zero, character data is non-blank and date fields are non-blank. Suppressed on the **Edit/Browse Table** screen and displayed on the **Edit/ Browse Row** screen.

Suppressed Fields

Fields that are created with the display attribute S, p, m, or v are not displayed on the **Edit/Browse Table** screen used to select rows for editing but do appear on the **Edit/ Browse Row** screen during actual editing. This feature allows application developers to simplify the display for users.

Comment Fields

The C attribute designates a field that uses no storage in the Data Table but appears on the screen during table editing. This option may be used to insert a blank line to separate groups of fields. This can be done by specifying an attribute of C and a blank field name.

Display Features

The Display Features option allows you to specify whether or not the field name is displayed, how the display is punctuated, and where data displayed as the field value comes from. The default display includes the following components:

- the field name
- a colon separator
- field data in display format

During editing or browsing of the table it may be desirable to include text in the display that is not stored on the table. This text is referred to as field-related constants or descriptive text. Valid values for display features are listed in Table 4-7.

Table 4-7: Display Features Values

Value	Display Field Name	Display Colon	Data From	Usage
(blank)	Yes	Yes	Table entry	Default, normal display.
:	No	Yes	Table entry	Display a large field on multiple lines without redisplaying field name.
D	Yes	No	Description Table	Display ruler, comments, table instructions or other invariant data.
Y	Yes	Yes	Description Table	Display ruler, comments, table instructions or other invariant data.
N	No	No	Description Table	Display ruler, comments, table instructions or other invariant data.

Description Text and Field-Related Constants

Description fields allow you to insert features such as a ruler line that marks character positions, or, on-screen editing instructions using Field-Related Constants.

The description text is stored and retrieved at run time by a table lookup on a description table. The display attribute should be set to C. If the display attribute is not a C, the actual field data will also appear in the field with the description data. Using this feature, allows you to display data entry instructions from the description table next to a data field. To prevent the actual data from overlaying the description text you must provide enough spaces at the beginning of the description table entry to ensure that the actual data overlays only blanks.

See“Description Table Lookup Order” on page 147 for more information.

Display Mask

A Display Mask is a series of characters that determines how data is to be displayed. The Display Mask can be made up of any character, however, certain characters have special meaning. These special characters are referred to as delimiters and are different for fields that are defined as containing numeric or character data. The tables below illustrate the valid characters for each type of data. Display masks can be used with any display format. The number of characters represented in the Display Mask must be equal to the field length. A Display Mask of up to 50 bytes can be defined.

Numeric Data

Numeric data is filled from right to left and the decimal point is placed in the exact location specified in the Display Mask. The delimiters are replaced by numbers or leading blanks. Any characters introduced by the mask, such as separating commas, are suppressed if they are preceded by leading blanks in the data. The only exception to this occurs when the edit character is a decimal point and the absolute value to be displayed is less than 1. For example, `.0123`.

Table 4-8 identifies and describes the Special Delimiters for Numeric Fields.

Table 4-8: Special Delimiters for Numeric Fields

Delimiters	Meaning
.	Decimal Point. Inserts a decimal point at the location in the mask. For formats 1-9 the decimal must appear in the correct location in the mask. For N and 0 it can be placed anywhere in the number since it does not actually represent a decimal point in these cases. Formats are described in the section "Display Format Field Format" on page 52.
+ or **-**	A + or - can occur either in the leading or trailing mask areas. A leading plus sign causes both negative and positive data values to be identified by either a plus sign or minus sign as appropriate. A trailing minus sign causes only negative data values to be identified with a minus sign. A leading minus sign causes only negative data values to be identified with a minus sign. A trailing plus sign indicates that both negative and positive numbers are displayed with a trailing sign.
DB or **CR**	This symbol in a trailing position indicates that a negative number is displayed with a trailing blank and either DB (for debit) or CR (for credit).
( )	Parentheses placed around the number causes negative data values in the field to be enclosed in parentheses.
!	Exclamation mark. Identifies an open position in the mask where tablesONLINE places field data. As data is entered into the field, each character replaces one of the exclamation marks until they have all been replaced.
Other	Any other character is inserted exactly as shown in the mask into the display of the data value.

If a numeric field (Display Format N) contains `3456789`, Table 4-9 shows what would be displayed with several different Display Masks.

Table 4-9: Example Display Masks for Numeric Data

Display Mask	Produces
`!,!!!,!!!.!!+`	`34,567.89+`
`Model: -!!!!!-!!.!!`	`Model:  345-67.89`
`Dfl -!.!!!.!!!,!!`	`Dfl  34.567,89`
Display Mask For a negative value (`!,!!!,!!!.!!`)	`(  34,567.89)`

Character Data

Character data is filled from left to right. The delimiters are replaced by alphanumeric characters or trailing blanks (see Table 4-10).

Table 4-10: Special Delimiters for Character Fields

Delimiters	Meaning
!	Exclamation mark. Identifies an open position in the mask where tablesONLINE places field data. As data is entered into the field, each character replaces one of the exclamation marks until they have all been replaced.
Other	Any other character is inserted, exactly as shown in the mask, into the display of the data value.

If a 10 character alphanumeric field contains ABCD1234, Table 4-11 shows what would be displayed with several different Display Masks.

Table 4-11: Example Display Masks for Character Data

Display Mask	Produces
`!!!,!!!.!!!!`	`ABC,D12.34`
`MODEL: !!!!-!!.!!!!`	`MODEL: ABCD-12.34`
`Type !!!! Rev: !!/!!/!!`	`Type ABCD Rev: 12/34/`
`!!!! FLAGS !!/!!/!! OFF`	`ABCD FLAGS 12/34/  OFF`

When entering data in a field that uses a Display Mask, you have the option of clearing the field using the <EOF> erase key and entering the data characters only, or, you can enter the data directly in the spaces indicated by the mask.

Note: The <PF3> key can be used to erase text. To erase text, place your cursor immediately before the text you want to erase. Press <PF3> and all text after the cursor is erased.

Initial Value

The Initial Value field allows you to set a default value for the field. When a new row is created using the INSERT command or the I line command in the **Edit-Table** screen (see Figure 4-2), the Initial Value will appear in the field. If a Display Mask is specified then the initial value will appear as specified by the Display Mask. For example, an Initial Value of 10 with a Display Mask of `$!!!.!!-` will appear as an initial value of `$ 10.00.`

The initial value must conform to the Edit Pattern, Display Format, Edit Highest Value and Edit Lowest Value, if these values are defined.

While creating a new row on the **Edit-Table** screen (see Figure 4-2), the Initial Value can be overridden by typing over or clearing the entry unless the field is protected. To protect the field, enter P in the Display Attribute field.

Edit Pattern

The Edit Pattern field is used to specify an edit pattern for data entry. This edit pattern is used to validate the data. Pattern validation can only be used for fields with a display format of X. The pattern length must match the field display length. In order for input data to be accepted, it must match the validation symbols specified for the field. Table 4-12 is a list of the acceptable edit pattern validation symbols.

Table 4-12: Edit Pattern Validation Symbols

Validation Symbols	Characters Represented
Z	Alphabetic (A - Z or a-z)
A	Alphabetic (A - Z or a-z) or blank
9	Numeric (0 - 9)
I	Numeric (0 - 9) or blank
Y	Alphabetic (A - Z or a-z) or numeric (0 - 9)
X	Alphabetic (A - Z or a-z) or numeric (0 - 9) or blank
B	Blank
C	Any character (no validation)

Table 4-12: Edit Pattern Validation Symbols (Continued)

Validation Symbols	Characters Represented
Literals	A set of characters surrounded by exclamation marks (!)
!	This is a delimiter. Characters between the delimiters must be present in the field.

Note: A minus sign(-) before any validation symbol indicates anything but the specified character or literal string.

The following examples of edit patterns illustrate the use of this facility.

`ZZ!-!9999` - this pattern verifies that the first two characters are alphabetic, the third character is a hyphen and the last four characters are numeric.

`!AB!-I99-!ZZ!` - this pattern verifies that the first two characters are AB, the third character must not be a numeric or blank, the fourth and fifth characters are numeric and the last two characters are anything other than `ZZ`.

Edit Highest Value
Edit Lowest Value

The Edit Highest Value and Edit Lowest Value fields allow you to constrain the value of the field within an upper and lower bound. The values entered must conform to the Edit Pattern and Display Format, if these values are defined.

The value set in the Edit Highest Value field must be greater than or equal to the value set in the Edit Lowest Value field. For example, if the lowest value is -100 then the highest value could be -90, -10, or 100 but cannot be -220.

The Edit Highest Value and Edit Lowest Value fields must be used together, that is, both set or both left blank.

While creating a new row on the **Edit-Row** screen, the user will receive an error message if a value outside the bounds of those set in the Edit Highest Value and Edit Lowest Value fields is entered.

Action Codes

This is a multi-purpose field. It may be used to specify:

- existence, exclusion or range validation
- importation of data from external tables
- automatic date updating
- insertion of user ID of the last person who modified a row

It is used in conjunction with the three fields that follow: the Source Table, Source Field, and Trigger Field Names. The valid action codes are listed in Table 4-13, together with the requirements for the other three fields:

Table 4-13: Action Codes

Action Code	Action Performed	Source Table	Source Field	Trigger Field
Y	Existence check - accept only if entered value matches comparison value. When editing the field, a list of all available rows on the Source Table can be produced by pressing <PF1> for help.	Yes*	Optional	No
N	Exclusion check - accept only if entered value does not match comparison value.	Yes*	Optional	No
R	Range check - accept only if entered value falls within the range of comparison values	Yes*	Optional	No
E	Effective value retrieval - same as a range check, however, lower limit of range is taken from previous row in table.	Yes*	Optional	No
I	Data importation - import data from another source triggered by successful Y, R, E.	Yes*	Yes	Yes
D	Duplicate value - duplicate the data into this field whenever the value changes in the source field.	No	Yes	No
C	Creation date - put current date into this field only when the data row is created. This code may be used only once in a table. This MUST be a date field.	No	No	No

Table 4-13: Action Codes (Continued)

Action Code	Action Performed	Source Table	Source Field	Trigger Field
U	Update date - put current date into this field whenever any field in the data row is changed. This code may be used only once in a table. This MUST be a date field.	No	No	No
B	Insert UserID - put the UserID of the user currently editing the table into this field if the row is modified (last UserID). Use this code only once in a table. Procedure: 1) Add an eight byte character field to the View. 2) Set the action code of the field to B. 3) Save the View. 4) Restructure the table. Whenever a row in the table is updated the UserID of the person updating the table will be placed in the field for that row.	No	No	No
***** For the action codes where you are required to insert a Source View/Table, you can display a list of the available Views and Tables by placing your cursor in the appropriate field and pressing <PF1> for help. To select the desired View/Table move the cursor to the row in the list and press <Enter> or use the S line command.				

Table 4-14 lists the action codes that when assigned to a table or library field will display the described list when HELP is pressed for that field.

Table 4-14: Action Codes for Listing Tables or Libraries

Action Code	Action Performed	Source Table	Source Field	Trigger Field
L	List libraries - produces a list of authorized libraries from the TBOLLIBR table. Both the DDNAME and the Dataset names are listed. If the field is defined to be 44 bytes in length, the Dataset Name will be returned. If the field is defined to be 8 bytes in length, the DDNAME will be returned.	n/a	n/a	n/a
V	List Views - produces a list of all authorized Views.	n/a	n/a	n/a
T	List Tables - produces a list of all authorized tables or alternate indexes.	n/a	n/a	n/a

Existence Check (Y)

Verify that the value entered in this field exists as a key on the table specified as the Source Table. If the key value is present on the Source Table, the field data will be accepted. When this action is used, it is possible to display a list of the rows from the Source Table by pressing <PF1> for help.

Exclusion Check (N)

Verify that the value entered in this field does not exist on the table specified as the Source Table. If the value is not present on the Source Table, the field data will be accepted.

Range Check (R)

Verify that the value entered in this field lies within one of the ranges on the Source Table. The ranges consist of pairs of lower and upper limits. The value entered is accepted if it is greater than, or equal to, the lower limit and less than, or equal to, the upper limit. Range checking with numeric fields may be performed only for positive ranges.

The comparison ranges are specified in two consecutive fields of the Source Table (see Table 4-15). The first of the pair, the lower limit, is the key, and the second, displayed to the right, is the upper limit.

Table 4-15: Example Range Check Table

Lower Limit (Key)	Upper Limit
5000	6000
3000	4000
1000	2000

In addition, the following rules must be observed for the Source Table. The lower limit, the upper limit and the field being validated must have the same length and format, and the Source Table (or Alternate Index) must have a descending organization.

You may perform a simple range check that consists of only one range. In this case, data must fall within that range in order to be accepted. For this type of range check, the Source Table will contain only one entry.

You may also specify single ended boundaries for the range. A single-ended boundary may specify that a value must be greater than, or equal to, a lower limit with no upper limit or that a value must be less than, or equal to, an upper limit with no lower limit. In order to set a single-ended boundary use the following:

Lower Limit - with no upper limit

The value entered must be greater than or equal to XXX. Specify XXX as the lower limit (key) in the Source Table and specify zero as the upper limit.

Upper Limit - with no lower limit

The value entered must be less than or equal to XXX. Specify XXX as both the lower and upper limit in the Source Table.

Note: For simple range checks between and upper and lower value or for checks between positive and negative numbers or upper and lower case ranges see “Edit Highest Value Edit Lowest Value” on page 61.

Effective Value Retrieval (E)

Effective value retrieval is a type of range checking. It presumes that the upper limit of one range is the lower limit of the next range. There is, therefore, no need for the upper limit field. Range selection with numeric fields can only be performed for positive ranges.

The primary purpose for this type of table is the Data Importation action described below.

The value entered determines which range is selected. A range is selected if it is greater than, or equal to, the lower limit (the key on the Source Table) and less than the next higher key, the upper limit.

In addition, the following rules must be observed for the Source Table. The lower limit (the key on the Source Table) and the field being validated must have the same length and format, and the Source Table that lists the acceptable ranges (or Alternate Index) must have a descending organization.

Data Importation (I)

Data Importation specifies a field into which data is imported when a Y, R or E action is successful in a previous field. The value to be imported is identified as follows:

- Both a Source Table name and an import Source Field name must be specified.
- A trigger field name must also be specified. The Trigger Field is a field in the View with an action of Y, R or E. The length and format of the Trigger Field must match the key of the Source Table.

When a value entered in a Trigger Field is accepted, a row is retrieved from the Source Table using the entered value as the key. The imported data value is taken from the source field of the retrieved row in the Source Table.

Warning: An import field can have multiple triggers. It is possible to have multiple fields defined with the same name and defined as potential trigger fields. If an import field is then defined with this name as its trigger field, then the import will be triggered by a change in any of the fields with the same name.

Duplicate Value (D)

When this action code is specified for a field, the value in the field is copied from a source field in the same table row. The value in the field is automatically duplicated whenever the source field value is changed. The length and format of the source field must match the field being specified as a duplicate. In order to perform this function the Source Table Name column must be blank.

Warning: A duplicate field may have multiple sources. It is possible to have multiple fields defined with the same name. If the source field name of a duplicate field exists multiple times, the duplicate field will be updated whenever one of these fields is modified.

This action code can be used in situations where it is necessary for users to view a table using two different keys. Suppose that the table contains the following fields,

State

City

Street Name

THEN:

- User1 wishes to view the table with the key “City/Street Name”
- User2 wishes to view the table with the key “Street Name/City”

Using the Data Duplication action code you can define a duplicate of the City field that follows the Street Name field in the table. The result is the following fields in your table: State, City, Street Name, City.

When the Views are created for each user, one of the City fields is suppressed in each case and the key is defined using the remaining City and Population fields. This results in:

- Key 1 - City/Street Name
- Key 2 - Street Name/City

Both users are actually viewing the same data through their own custom View, but only one copy of the data must be maintained since the duplicate is updated automatically when the source field is modified. Remember, keys need to be contiguous data fields.

Creation Date (C)

This causes a date field to receive the operating system's current date when a row is first created. A date field with this action is protected from update by the user. The Creation Date code may only be used once in a table.

Update Date (U)

This causes a date field to receive the operating system's current date when the row is updated. A date field with this action is updated automatically by the program and is protected from update by the user. The Update Date code may only be used once in a table.

Source View/Table

The name of the View/Table entered here will be used for action codes Y, N, R, E and I. This field must be left blank when the action code is D, C or U.

The edit action codes Y, N, R and I are usually performed with reference to a View. If a corresponding View is not available for the source View/table, the source field name is left blank. This indicates that tablesONLINE accesses the Data Table or Alternate Index directly. In this case, the key of the Source Table must be the same size and format as the current field and data importation is not possible from that particular Source Table.

Source Field Name

The Source Field name identifies a field within the Source Table that is used to validate data or as the source of import, depending on the action code specified (Y, N, R, E, D or I). This field must be the entire key field of the Source Table for actions Y, N, R and E. If a name is entered here, it indicates that a View for the Source Table will exist at table edit time.

Trigger Field Name

The Trigger Field name is required when Data Importation (I) is specified. The field name entered may be any field of the current View that has an action code of Y, R or E.

Field Exit Name

Together with the Field Exit Version, Field Exit Input Indicator and Field Exit Output Indicator, this field controls the calling of a user exit program from the tablesONLINE editor. See the chapter on *tablesONLINE/CICS Exit Programming* in the tableBASE Programming Guide for information on the design and use of exit programs.

This field allows you to specify the name of the exit program to be called by tablesONLINE.

Note: To activate the Field Exit program you must set the Field Exit Name and at least one of the following indicators: Field Exit Input Indicator and/or Field Exit Output Indicator.

Field Exit Version

This option is for backward compatibility and allows you to indicate whether the exit program specified was compiled with a particular tableBASE release.

- Enter **4** to indicate the exit program was compiled with copybooks provided with Release 4
- Enter **5** to indicate the exit program was compiled with copybooks provided with Release 5.0.x
- Enter **6** to indicate the exit program was compiled with copybooks provided with releases 5.1 or later

Field Exit Input Indicator
Field Exit Output Indicator

The indicator fields tell tablesONLINE when to call the exit program. The valid values for these indicator fields are listed in Table 4-16.

Table 4-16: Field Exit Indicators

Value	Exit Program Called
(blank)	Not called
B	Before the action
A	After the action
Y	Both before and after the action

For example, when the editor is getting data for display, if the exit input indicator is set to B for a field, then the corresponding exit program will be called before data is taken from the row.

If this indicator is set to A, the exit program is called after the data is taken from the row and converted ready for display. A decision about whether or not to display the data can be made within the exit program.

Exit processing based on field accesses as described above is one of the most powerful features of tablesONLINE. Exits are also provided at the row level and the table level, making it possible to invoke an exit program when the table is opened or each time a row is retrieved. Row and table level exit information is provided in the Supplementary View discussed later in this chapter.

Data Display Extended Attributes
Field Name Extended Attributes

The two EXTATT fields define the extended attributes to be used for special effects when the associated fields are displayed. Data Display EXTATT controls display of data from the field being defined, while Field Name EXTATT applies to the field name when it is displayed, as a column heading in a select row screen. This allows users with 3270-series color terminals to control the colors and allows any user to control features such as blinking or reverse video display.

The User Profile (for additional information see, "Edit Profile" on page 115) also contains the Row Data EXTATT and Field Name EXTATT fields. If non-zero data is present, the field-specific extended attribute information from the View overrides the User Profile extended attributes.

Extended attribute usage is highly system dependent. What can be done with this field will depend both on what your terminal can accommodate and on how the local CICS installation's attribute handling has been set up. All the extended attribute fields are initialized to low values that do not affect standard displays.

Due to the system-dependent nature of these codes, tablesONLINE itself does not perform any error-checking on them.

Define View Supplementary

Once you have specified the row layout information that defines your table View, you may wish to specify some of the table access controls that are described below. The screen required to define these controls is accessed using Option 2 - DEFINE VIEW SUPPLMT from the Define Table and View menu (Figure 4-1). Selecting Option 2 will bring up the **Supplementary View Information** screen (see Figure 4-4).

```
tablesONLINE 6.0.2 Administrator ------ Edit Supplementary Information for View
COMMAND ===>

                            Enter Supplementary Information for View
                TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DICTIONARY.LIBRARY
     VIEW NAME            : EXAMPLE
     DEFN VERSION         : 5
     WRITE PASSWORD       :

     USER COMMENTS        :
     USER ID              : DKLS02
     DATA TABLE NAME      :
     HELP TABLE NAME      : EXAMHELP
     MULTIPLE USER UPDATE :
     DUPLICATE KEYS IND   : N
     PROTECT KEYS IND     :
     HASH RE-ORG CODE     :
     FREEZE KEYS COUNT    : 2
     DUPLICATE PRIME KEYS :
     DYNAMIC SUFFIX LOCN  : 0
     DYNAMIC SUFFIX LNGTH : 0
     VIEW ITEM SIZE       : 90
     VIEW KEY LOCATION    : 1
     VIEW KEY SIZE        : 20
     PRIMARY KEY LOCATION : 0
     PRIMARY KEY SIZE     : 0
     ITEM EXIT NAME       : DSPLSTIF
     ITM/TBL EXIT VERSION :
     ITEM EXIT INPUT IND  :
     ITEM EXIT OUTPUT IND :
     ITEM EXIT XFIELD IND :
     ROW EXCLUDE IND      :
     TABLE EXIT NAME      : DSPLSTIF
     TABLE EXIT OPEN IND  :
     TABLE EXIT STORE IND :
     TABLE EXIT CLOSE IND :
```

Figure 4-4: Supplementary View Information

Note: For ease of reading all the fields of the **Supplementary View Information** screen (Figure 4-4) are shown as one screen. However, in reality, the fields are spread over multiple screens; to view each screen use the scroll keys, <PF7> and <PF8>, to move back and forth between the screens.

Each field of the Supplementary View Information screen is described below.

Library Name

Enter the name of the library where the View is stored. You may alternatively enter any alias that is defined in TBOLLIBR, a number from 1-7 to select a relative library number, a DDNAME or full dataset name. To display a list of all authorized libraries, place the cursor in the library field and press <PF1>. To select a library from the displayed list, place the cursor beside the row and press <Enter> or use the S line command.

If you do not know in which library the View is stored, clear the name in this field and all authorized libraries will be searched.

There is a safety feature to prevent illogical changes from this screen. If Library Name and/or View Name has been changed, then tablesONLINE rejects the first attempt to store the modified trailer with the <PF2> key. Instead, it reads in the data for the new table so that it can be modified. It will not allow you to store supplementary information you have not read.

View Name

Enter the name of the View for which you wish to define supplementary information. To display a list of all authorized Views using the List Facility, place the cursor in the View Name field and press <PF1>. A list of all libraries containing a View with the specified name will be displayed. If you clear the View Name field, a list of all authorized Views will be displayed. To select a View from the displayed list, place the cursor beside the row and press <Enter> or use the S line command.

Definition Version

This field will display an entry indicating whether the selected View was created for Release 4.0 or a subsequent release of tablesONLINE. An entry will automatically appear in this field when this screen is displayed. A 5 in this field reflects all releases from 5.0 to the current release.

Write Password

A write password may be set for the View from the tablesONLINE Utilities menu. If a write password has been set for the View, then it must be specified in this field before you can edit the supplemental information for the selected View.

User Comments

This field allows you to insert a comment up to 16 characters long, into the View supplemental information. This comment field may be used to give a brief description of the table.

User ID

This field will display a user identification. The User ID is placed into this field automatically by tablesONLINE based on the Sign-on ID of the individual that created or last updated the tablesONLINE View.

Data Table Name

If the Data Table and the View share a common name, you do not need to complete this field. If a different View is used to access the Data Table and its name is not the same as the Data Table, then the Data Table name should be specified in this field to indicate that tablesONLINE should resolve the relationship accordingly. The List Facility can also be used here to find and select a Data Table name. Place the cursor in the field and press <PF1>.

Help Table Name

Enter the name of the XXXXHELP table you wish to use with the View being defined. The Help table name specifies a table containing help information specific to the Data Table or the View being defined. If this field is filled in, the named table is searched before the application's Help table when help is requested during an edit using this View. The List Facility can also be used here to find and select a Help table name. Place the cursor in the field and press <PF1>.

Multiple User Update

Multiple User Update controls how multiple users can access table data. There are three options available:

- (blank) - this is the default. Multiple user access to the Data Table is turned off
- B - all users will be restricted to browse-only access to the Data Table.
- M - this allows simultaneous editing of the Data Table by multiple users.

Option M requires the following additional settings:

When multiple users are editing a table, duplicate prime keys are not allowed as they cannot be referenced properly.

If this is a View for the Data Table, the Duplicate Keys Indicator field must be set to Option N (no duplicate prime keys in the Data Table).

If this is a View for an Alternate Index, the Data Table key(s) must be unique (see “Enforcing Unique Data Table Keys with an Alternate Index” on page 91). and the Duplicate Prime Keys field must be set to N. The Duplicate Keys Indicator field can then be set to either blank, Y or N (see “Duplicate Keys Indicator” on page 74).. The definition of the Duplicate Keys Indicator field will apply only to the keys of the Alternate Index View, while setting Duplicate Prime Keys field to N will ensure that the prime keys of the Data Table are unique.

The multiple-user update feature can be defined for Data Table Views and/or Alternate Index Views. Users entering a multi-user updateable table are notified that multiple user access is in effect.

When entering the **Edit-Table** screen, an asterisk beside the row identifies the rows that are currently being edited by another user. To test whether a row is still in use attempt to edit it. If a row is being edited by another user, you will only be allowed to browse the row.

Duplicate Keys Indicator

This option indicates whether duplicate keys can be created within a table and whether or not the display screen will contain a prompt to allow users to enter duplicate keys in those cases where it may be allowed.

If this field is set to N, no two table rows can have the same value in their key fields. If duplicate keys is set to Y or left blank, the user has the choice of whether or not to allow duplicate keys in the table (see Table 4-17).

Table 4-17: Duplicate Key Indicator Values

Value	Display Option on Table ID Screen?	Initial Value of Option	Allow user to Alter?	Duplicate Keys Actually Permitted?
N	No	---	No	Not with this View
Y	Yes	Y	Yes	User's choice
(blank)	Yes	N	Yes	User's choice

Protect Keys Indicator

The Protect Keys Indicator specifies whether or not the keys in the table are protected. This option can be used to limit the editing process to fields that are not key fields. If Protect Keys is set to Y, the key fields in the table are protected and cannot be edited. As well, rows cannot be added to, or deleted from, the table.

Hash Reorganization Code

This code controls whether tablesONLINE reorganizes Hash true tables for editing. Codes of S (sequential) and D (descending) in this field cause the table to be reorganized as ascending sequential or descending sequential, respectively. This feature should only be used when you are certain that no other application will attempt to access the table during editing.

When using the Hash re-organization code you must also set the duplicate keys indicator in the View to N. This feature applies only to Hash true tables. Hash pointer tables are always accessed via an index that makes them appear to be in ascending sequential order.

Freeze Keys Count

The Freeze Keys Count specifies the number of fields that are to be locked. The specified number of fields will not scroll when the table is displayed. The value set here is used as a default. The user may change it once the application is running.

Duplicate Prime Keys

An N in this field indicates that unique Data Table keys should be enforced. This field is blank by default.

Note: If multiple-user access is enabled and you are defining an Alternate Index View, the Duplicate Prime Keys field must be set to N.

Dynamic Suffix Location

The Dynamic Suffix Location is automatically inserted in to this protected field by tablesONLINE. This field will contain a non-zero value if a field in the table has been defined as a dynamic suffix field with an S, B, Q or A key field indicator. The value is the location of that field within the row.

Dynamic Suffix Length

The Dynamic Suffix Length is automatically inserted in to this protected field by tablesONLINE. This field will contain a non-zero value if a field in the table has been defined as a dynamic suffix field with an S, B, Q or A key field indicator. The value is the length of that field.

View Item Size
View Key Location
View Key Size

The View Item Size, View Key Location and View Key Size are all protected fields maintained by tablesONLINE. They show table parameters that have been calculated by tablesONLINE from the View information provided. When tablesONLINE is asked to open a table, it obtains the tableBASE definition of the table from the library and checks that the Row Size, Key Location and Key Size match those in the View. If they do not, an error message is generated and the open fails.

Primary Key Location
Primary Key Size

These are protected fields maintained by tablesONLINE. They show table parameters that have been calculated by tablesONLINE from the View information provided.When a

table is opened using an Alternate Index and the Duplicate Prime Keys indicator is set to N, tablesONLINE obtains the tableBASE definition of the table from the library and checks that the Prime Key Location and Prime Key Size match those in the View.

Item Exit Name

This field allows you to specify the name of the item exit program that is to be called. Item exit programs may be used to access data outside tableBASE or to perform cross-field validation.

An item-level exit program will be called whenever the current row has been changed and the user hits <Enter> or takes any action that requires that the current row be either input, output and/or cross-field validated.

Note: To activate the Item Exit program you must set the Item Exit Name and at least one of the following indicators: Item Exit Input Indicator, Item Exit X-Field Indicator and/or Item Exit Output Indicator.

Item/Table Exit Version

This option is for backward compatibility and allows you to indicate whether the exit program specified was compiled with a particular tableBASE release.

- Enter **4** to indicate the exit program was compiled with copybooks provided with releases 4.0 to 4.2
- Enter **5** to indicate the exit program was compiled with copybooks provided with release 5.0.x
- Enter **6** to indicate the exit program was compiled with copybooks provided with releases 5.1 or later

Item Exit Input Indicator
Item Exit Output Indicator
Item Exit X-Field Indicator

Item exit indicators may be set to apply to input, output, or cross-field validation: input applies to data movement from table to screen; output applies to movement from screen to table, and X-Field controls calls to the item exit program for cross-field data validation.

The set of permitted values for the item exit indicators are listed in Table 4-18:

Table 4-18: Item Exit Indicators

Value	When Exit Program Called
(blank)	Never
B	Before the action
A	After the action
Y	Both before and after the action

Note: The cross-field validation indicator may only be set to Y or blank. If it is set to Y, then the exit program is invoked when the current row has been changed and the user presses <Enter> or takes any action that requires that the current row be output back to the table.

Row Exclude Indicator

The Row Exclude Indicator is used in conjunction with user exit programs that prevent certain rows from being displayed.

If you have an exit program whose purpose is to suppress rows from being displayed and you perform a block delete in the **Edit-Table** screen, you may be affecting rows you are not aware exist, since they are not displayed. In order to prevent the deletion of hidden rows, set the row exclude indicator to Y to exclude all hidden rows from the operation being performed.

Table Exit Name

This field allows you to specify the name of the table exit program that is to be called. Depending on the indicator that is selected, a table-level exit program will be called when one of three events occurs - a table is opened, stored and/or closed.

Note: To activate the Table Exit program you must set the Table Exit Name and at least one of the following indicators: Table Exit Open Indicator, Table Exit Store Indicator and/or Table Exit Close Indicator.

Table Exit Open Indicator
Table Exit Store Indicator
Table Exit Close Indicator

Table exit indicators (see Table 4-19) may be set to apply to open, store or close. If the Table Exit Open Indicator is set, then the exit program specified above is invoked when the table is opened. If the Table Exit Store Indicator is set, then the exit program specified is invoked when the table is stored. If the Table Exit Close Indicator is set, then the exit program specified is invoked when the table is closed.

Table 4-19: Table Exit Indicators

Value	When Exit Program Called
(blank)	Never
B	Before the action
A	After the action
Y	Both before and after the action

For example, if the Table Exit Open Indicator is set to B, then tablesONLINE will run the exit program before attempting to open this table.

Define Data Table

Once the row layout and the table access controls have been completed, the next task is to define the physical table attributes such as organization and search method. This is done on the **Table Definition** screen (see Figure 4-5).

```
tablesONLINE 6.0.2 Administrator ------ Edit the Data Table Definition --------
COMMAND ===>


                          Edit tableBASE Table Definition (DT Block)
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DATA.TABLE.LIBRARY
     TABLE NAME           : EXAMPLE
     GENERATION           : 0
     PASSWORD             :

     TABLE ORGANIZATION   : S
     SEARCH METHOD        : B
     TRUE OR POINTER      : T
     STORAGE MODE CODE    : R

     ROW LENGTH           : 64
     KEY LENGTH           : 34
     KEY LOCATION         : 1
     NO OF ROWS           : 30        Enter a good estimate for new table.

     GENERATIONS TO KEEP  : 8
     EXPANSION FACTOR     : 20.0
     LOW DENSITY (HASH)   : 50.0
     HIGH DENSITY (HASH)  : 80.0
     DEFAULT VIEW NAME    :
     USER COMMENT AREA    :

     DATE (YYYYMMDD)      : 1998-10-25
     TIME (HHMM)          : 18:49
     LIBRARY DATASET NAME : DKLTBT.V5R1M0.TST.O27.MAINLIB
     ABSOLUTE GENERATION  : 38
     RELATIVE GENERATION  : 0
     GENERATIONS PRESENT  : 8
     MAX. ITEMS BEFORE EX : 37
     DDNAME               : MAINLIB
     USER ID              : DKLS02
     PHYSICAL DATA TABLE  :
     TABLE OPEN STATUS    : X
     ALTERNATE INDEX IND  : N
```

Figure 4-5: Table Definition Screen

Note: For ease of reading all the fields of the **Table Definition** screen (Figure 4-5) are shown as one screen. However, in reality, the fields are spread over multiple screens; to view each screen use the scroll keys, <PF7> and <PF8>, to move back and forth between the screens.

In order to prevent accidental changes from occurring, the following precautions have been put in place:

- The <End> key has no effect on this screen except to exit from it.
- The <Enter> key updates the screen only. If the library name and/or View name have been changed, then the definition of the newly specified table is displayed.

Note: The <PF2> key is required to change the table definition. Before pressing <PF2>, press <Enter> and to verify that the screen has no errors.

Library Name

If you are creating a new Data Table, enter the name of the library where you wish to store the Data Table. This field will default to the name of the last library you used in tablesONLINE. If this is the library where you wish to store your new Data Table, leave the default library name as displayed. If you wish to specify a different library, replace the default entry with the new library name. You may wish to store the Data Table in the same location as the associated Views.

To view a list of authorized libraries place the cursor in the Library Name field and press <PF1>.

Table Name

Enter the name that you wish to assign to the new Data Table. This name must not already exist in the library you specified in the field above.

If you are editing an existing Data Table and wish to view a list of the authorized tables use the List Facility press <PF1>. To select a Table from the displayed list, place the cursor beside the row and press <Enter> or use the S line command. If you are creating a new Data Table, this list facility can be used to display the names of the existing tables.

Generation

When you are creating a new table definition, this field will default to 0. If you are editing an existing table definition, you should specify the generation of the table you wish to edit. See "Specifying Table Generations" on page 21 for detailed information on the use of generations.

Password

A write password may be set for a table from the tablesONLINE Utilities menu. If a write password has been set for a table, then it must be specified in this field before you can edit the table definition for the selected table.

Table Organization

The table organization indicates how table entries are to be ordered within the table. You can choose from the options listed in Table 4-20.

Table 4-20: Table Organization Options

Option	Table Organization
R	Random
U	User Ordered
S	Ascending Sequential
D	Descending Sequential
H	Hash

For a list of the permitted organization/search method combinations, see "Table Organization/Search Method Combinations" on page 84.

Random (R)

With a random organization, new rows are added at the end of the table. Since there is no systematic order to the table, searching must be done using the serial method.

User Ordered (U)

In a table with this organization, the ordering of rows is controlled entirely by the user. As a result, the system has no way of recognizing where a row may be stored, so the serial search must be used.

Sequential (S)

A sequentially organized table is sorted by key in ascending sequence. This organization allows for the use of various search techniques that exploit ordered keys. Insertion and deletion tends to be slower since other rows in the table must be moved to maintain the sequence.

Descending (D)

A table organized in descending order is sorted by key in descending sequence. This organization also allows various search techniques that exploit ordered keys. Insertion and deletion also tends to be slower since other rows in the table must be moved to maintain the sequence.

Hash (H)

In a Hash table, rows are positioned by an arithmetic operation performed on the key by tableBASE. Selecting a Hash organization causes the data to take up more space than

other organization options, but allows rows to be retrieved and updated very quickly. The only search method that can be used with the Hash organization is a Hash search.

Search Method

The search method specifies how the system will look for data stored in the table when that table is used by a program. You can choose from the following options:

- S - Serial
- Q - Queued
- B - Binary
- C - Address Tree Binary
- H - Hash

Serial (S)

The serial search starts with the first row in the table and moves consecutively through the rows until a match is found or the end of the table is reached. The average retrieval time for a random set of keys increases linearly with table size. In a user ordered table, where rows can be ordered by frequency of access, the serial search may be the fastest search.

Serial search is also the fastest search method for small tables (less than 15 rows) where more complex searches do not justify extra computational requirements.

Queued (Q)

The queued search method is a variation of the serial search. Each subsequent search begins where the previous one left off. The search finishes when a match is found, the key is out of range or the end of the table is reached. This method works only with sequential or descending sequential tables.

When attempting to align table data for an ordered data source or for a merge operation with two tables that are organized in the same way, this search method may be the most efficient available. Queued is the only method that allows you to take advantage of the ordering of both tables.

Binary (B)

This search method compares the key of a row to the middle row of an ascending sequential or descending sequential table to determine which half of the table contains the row. The search then divides that half into half once again. It continues this process of dividing the remainder in half until the row is found. This method provides fast access and updating of sequentially organized tables.

Address Tree Binary (C)

The Address Tree Binary search process searches an index to determine if the search key is in the index.

If the search key is found outside of the index then the system returns a not found message.

If the search key is found in the index then the Address Tree Binary search process looks at the index as a tree with branches. The process then selects the correct branches to determine the location of the search key. The search process continues to search the branches until a minimum key value is reached. At that point the process performs a sequential search of the index to find the search key.

Note: Address Tree Binary is faster than Binary if there is not more than one insertion or deletion for every five table accesses.

Hash (H)

A Hash search method is only used to search a table with a Hash organization. Since the rows in a Hash table are stored based on an arithmetic operation, that same operation is used in the search to determine the location where the row will be found. The Hash search method is very fast if you have a large table.

Table Organization/Search Method Combinations

Table 4-21 shows which search methods may be used with each type of table organization. Illegal combinations are indicated by dashes (---).

Table 4-21: Search Method and Table Organization

		Search Method				
Organization		**S - Serial**	**B - Binary**	**C - Address Tree Binary**	**Q - Queued**	**H - Hash**
R	Random	Default	----	----	----	----
U	User Ordered	Default	----	----	----	----
S	Sequential	----	Default	OK	OK	----
D	Descending Sequential	----	Default	OK	OK	----
H	Hash	----	----	----	----	Default

In general, you will choose the organization and search method as a pair. For many applications, you can specify the organization and tablesONLINE will fill in the default search method automatically. For random, user ordered or hashed tables, there is only one search method.

Ascending sequential and descending sequential tables offer a choice of search methods. For these, Binary search is the reliable default choice. Tree binary and queued searches perform better in special cases, but are also subject to greater performance degradation if their conditions (fairly consistent size for tree binary and an ordered data source for queued) are not met.

True/Pointer Tables

Here you must specify whether the table is to be a T (True) or a P (Pointer) table. The default value for this field is P.

A true table stores and retrieves the actual table rows in the manner specified by the table organization and search method assigned to it. A pointer table stores the actual rows in a randomly ordered table that is accessed indirectly through pointers. Organization and search method actually apply to the index. The appearance of the table to the user is identical whether the table is a true table or a pointer table.

In Release 6, internally, all tables are maintained through indexes (pointer table).

In cases where multiple Indexes are to be created for a Data Table, the table must be defined as a pointer table.

Note: In order to create an Alternate Index, the Data Table must be defined as a pointer table.

Storage Mode Code

This option controls how a table is to reside in memory. In Release 6 the setting here is R — Resident, which is the only option.

Paged tables are no longer supported in Version 6.

Row and Key Parameters

These parameters — row length, key length, key location — are used to allocate storage for rows and to locate keys within the rows (see Table 4-22). If you have defined your table using Option D - Define All in the Define Table and View menu, these numbers will be generated for you by tablesONLINE based on the information provided in previous screens.

Table 4-22: Row and Key Length Limits

Field Name	Minimum Value	Maximum Value	Default Value
Row Length	1	32,767 bytes	1
Key Length	1	Smaller of 256 or Row Size	1
Key Location	1	Row Size - Key length+ 1	1

Caution: Although you may enter your own data here, or alter the existing data, it is generally good practice to use the numbers generated by tablesONLINE.

Number of Rows

This field allows you to set the number of rows that your table will contain. The maximum limit on the number of rows will depend on the row size and on the maximum table size your system configuration can support.

If the table is already defined, this field contains the actual number of rows in the table.

Generations to Keep

tableBASE allows up to nine generations of a table to be kept in the tableBASE library. Disk space usage increases in proportion to the number of generations kept.

Table 4-23: Generations to Keep

Field Name	Minimum Value	Maximum Value	Default Value
Generations To Keep	1	9	1

Expansion Control

These three fields are the parameters used by tableBASE to control the automatic expansion of tables as they increase in size.

Table 4-24: Expansion Control

Field Name	Minimum Value	Maximum Value	Default Value
Expansion Factor	0.1	99.9	20.0
Lower Density	0.1	99.9	50.0
Upper Density	0.1	99.9	80.0

Non-hash tables will be set up initially with an allowance of free space for growth as indicated by the expansion factor. If a table is set up with the number of rows set to 100 and the expansion factor set to 20.0 percent, the table can grow to 120 rows before it requires expansion. When the table reaches 120 rows, it will be expanded by 20% to 144 rows, and so on. The density limit is ignored for non-hash tables.

The initial table size for a hashed table is set so that the expected number of rows in the table will represent the percentage set for the lower density. If the table is expected to contain 100 rows and the table is set with a lower density of 50%, then the initial space allocated for the table will be for 200 rows.

As successive additions are made to the table, the upper density limit may be reached. If the upper density is set to 80% then this will occur when there are 160 rows in the table. When this limit is reached, the table will be expanded to 320 rows in order to bring the density of the table back to the lower density that was set to 50%.

Default View Name

This field allows you to specify a View that will be used as the default View for this Data Table. This field may be overwritten by the user when the Data Table is opened for Edit or Browse (see "Define Many To Many (M2M)" on page 98).

User Comment Area

This field allows you to insert a comment, up to 16 characters long. This comment field can be used to insert a brief description of the table.

Generated Table Definition Information

The balance of the screen contains fields that have been generated by tableBASE. These are protected fields and cannot be modified by the user. The following is a brief description of each of these fields.

Table 4-25: Generated Table Definition Fields

Field Name	Description
Date	This field contains the date that the table generation was stored.
Time	This field contains the time that the table generation was stored.
Library Dataset Name	This field contains the name of the tableBASE dataset that contains the table.
Absolute Generation	This field contains the absolute generation number for the table.
Relative Generation	This field contains the relative generation number for the table.
Generations Present	This field contains the number of generations of the table that currently exist.
Max Rows before Expansion	This field indicates the maximum number of rows that the table may contain before it is expanded by tableBASE based on the number of rows and the expansion factor.
DDNAME	This field contains the DDNAME in the CICS start-up JCL for the tableBASE library.
User ID	This field contains the user ID of the person who last changed or stored the table.
Physical Data Table *	Not Active
Table Open Status *	Not Active
Alternate Index Indicator *	Not Active
* Since the Table Definition displayed is from the library and is not the definition of the table that may be open in the TSR, these fields only reflect the values of the library definitions.	

Edit Display Order

Once you have completed the field definitions for your table, you may wish to change the order in which the fields will be displayed. Only the order of non-key fields may be modified since all key fields must appear in contiguous order at the beginning of the table.

To edit the display order of the fields in your table, use Option 4 - EDIT DISPLAY ORDER on the Define Table and View menu, (see Figure 4-1). This will display the **Identify Table** screen with the last table you worked on in tablesONLINE. If the information provided is not correct, make any required changes. Press <Enter> to open the **Edit Field Display Order** screen (see Figure 4-6).

```
tablesONLINE 6.0.2 Administrator ------ Edit View --------- Field Display Order
COMMAND ===>

View Name : EXAMPLE                          Field Display Position : 1

          FIELD          KEY DISPL  DISPL  DISPL DISPL FIELD FIELD  FIELD  ACT
           NAME          FLD LENGTH FORMAT ATTR  FEAT  LOCN  LENGTH FORMAT
_____________________    IND _______ _______ _____ _____ _____ _______ _______ ___
LAST NAME                 Y    20     X                    1    20     X
FIRST NAME                Y    14     X                   21    14     U
DIVISION                  N     8     X                   35     8     U
DEPARTMENT                N     8     X                   43     8     U
HOW OFTEN                 N     4     N                   51     2     P
DATE OF CONTRIBUTION      N     8     A      V            58     8     A
CHARITABLE DONATION       N     6     2                   53     5     N      Y
```

Figure 4-6: Edit Field Display Order Screen

The **Edit Field Display Order** screen displays all of the field names for the table with the row layout. Each field definition occupies one line on the screen. Use the editor line commands described in Chapter 3, to re-order the fields as required. Key fields for a table must remain contiguous and in the same order in which they were created. As such, tablesONLINE will not allow you to re-order the key fields.

It is also possible, from this screen, to perform some additional editing functions. New fields can be added to the table, using this option, only if they are comment fields. In addition, edits to existing rows may be made, provided they do not alter the physical layout of the row. Changing the display length or display format of some fields is possible.

Browse View

Option 5 - BROWSE VIEW on the Define Table and View menu simply allows you to open a View and examine the information stored in it. It is not possible to make any modifications to the stored information using this option.

Create an Alternate Index

An Alternate Index permits access to the Data Table using a different key, organization and/or search method than those originally defined in the physical table attributes. Using this feature, many different indexes on the Data Table can be created. Since there is a single copy of the Data Table, any Views based on Alternate Indexes reflect any changes made to the Data Table. Conversely, changes made in an Alternate Index are also made in the Data Table.

To edit or browse a table using an Alternate Index:

1. Create a View in the Define Table and View menu (see "Table Defining Options" on page 45).

or

1. Copy and modify an existing View (see "Copy View" on page 109).

2. Define the fields of Alternate Index View (select the key(s), edit the display order, and other fields as appropriate as you would to define any View)

3. Select Option 6 - CREATE ALTERNATE from the Define Table and View menu.

Note: To create an Alternate Index for a table, the Data Table must have been defined as a pointer table.

On the **Create an Alternate Index** screen (see Figure 4-7), enter the name of the library where the Alternate Index for the table will be stored, the name to be assigned to the Alternate Index, and the name of the table to which the Alternate Index applies. Next, specify an Index Organization, Search Method, Key Location and Key Length. To complete the creation of the Alternate Index, press <PF2>. A new Alternate Index and an Alternate Index View have now been created.

```
tablesONLINE 6.0.2 Administrator ------ Create an Alternate Index ------------
COMMAND ===>


                          Create or Edit an Alternate Index
              TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : TBDEV.V600.O27.TBAPPLB
     ALTERNATE INDEX NAME : FIRSTNAM

     TABLE NAME           : EXAMPLE

     INDEX ORGANIZATION   : S
     SEARCH METHOD        : B
     KEY LOCATION         : 1
     KEY LENGTH           : 44
```

Figure 4-7: Create Alternate Index Screen

Note: The key location and key size of the Alternate Index View can be obtained by selecting Option 2 - DEFINE VIEW SUPPLMT from the Define Table and View menu and copying the values from the View Key Location and View Key Size fields.

Modifying an Existing Alternate Index

To update an existing Alternate Index, open it by inserting the library name and the Alternate Index name in the appropriate fields and pressing <Enter>. Make any required changes to the fields in the Create Alternate Key Definition. Once your changes are complete, enter UPDATE on the command line and type <PF2>.

Note: Copying an Alternate Index causes you to create a table with the attributes set in the Alternate Index.

Enforcing Unique Data Table Keys with an Alternate Index

The default set up of an Alternate Index allows the possibility that duplicate Data Table keys are created in the Data Table. This can occur when an Alternate Index is created and the key that is used is not the same as the Data Table key. When data is entered using this Alternate Index, a duplicate Data Table key can be created that while it does not conflict with the key of the Alternate Index, does conflict with the existing Data Table key.

To illustrate this potential conflict, examine the following example shown in Figure 4-8 and Figure 4-9).

```
tablesONLINE 6.0.2 Administrator ------ Edit View ---------- Table's Row Layout
COMMAND ===>

View Name : DATAVIEW                                  Field Position : 1

            FIELD          KEY DISPL  DISPL  DISPL DISPL FIELD  FIELD
             NAME          FLD LENGTH FORMAT ATTR  FEAT  LENGTH FORMAT
_____________________      IND (DATA) ______ _____ _____ _______ ______
EMPLOYEE NUMBER             Y    10     X                  10    X
LAST NAME                   N    20     X                  20    X
FIRST NAME                  N    20     X                  20    X
```

Figure 4-8: Example - DATAVIEW

```
tablesONLINE 6.0.2 Administrator ------ Edit View ---------- Table's Row Layout
COMMAND ===>

View Name : INDXVIEW                                  Field Position : 1

            FIELD          KEY DISPL  DISPL  DISPL DISPL FIELD  FIELD
             NAME          FLD LENGTH FORMAT ATTR  FEAT  LENGTH FORMAT
_____________________      IND (DATA) ______ _____ _____ _______ ______
EMPLOYEE NUMBER             N    10     X                  10    X
LAST NAME                   Y    20     X                  20    X
FIRST NAME                  N    20     X                  20    X
```

Figure 4-9: Example - INDXVIEW

Assume that no duplicate keys are permitted in either table.

If a user were to update DATAVIEW (see Figure 4-8) with an Employee Number that already existed in the data, then the update would be rejected.

If a user were to update INDXVIEW (see Figure 4-9) with a Last Name that already existed in the data, then that update would be rejected.

However, if while using INDXVIEW a user enters data that contains an Employee Number that already exists but with a Last Name that is unique, this update will be accepted and the Data Table now contains an entry with a duplicate Employee Number. A duplicate Data Table key has been created and permitted using the Alternate Index View.

To enforce a unique Data Table key(s) while using an Alternate Index, extra steps need to be taken in defining the Alternate Index View. The definition for the Alternate Index View needs to include not only its own key(s) but also the Data Table key(s) of the Data Table, and the Duplicate Prime Keys field must be set to N. This can be done when creating the Alternate Index View or by editing an existing Alternate Index View:

1. Identify the fields that are key(s) for the Alternate Index View and Data Table key(s) for the Data Table.

2. Edit the field definitions of each of the fields identified in Step 1 and set the Key Field Indicators to the appropriate values (see Table 4-2).

There are eight possible options for the definition of the Key Field Indicator, but only four (see Table 4-26) are relevant to enforcing a unique Data Table key (see "Key Field Indicator" on page 50). By using these options, you can indicate whether a field is a key in the Alternate Index and/or a Data Table key.

Table 4-26: Data Table Key Field Indicators for an Alternate Index View

Option	Description
P	Indicates that the field is a Data Table key.
O	Indicates that the field is a Data Table key and a key of the Alternate Index.
Q	Indicates that the field is both a Data Table key and is to be used as the Dynamic View Suffix.
A	Indicates that the field is a Data Table key, a key of the Alternate Index. and is to be used as the Dynamic View Suffix.

In addition to defining the keys of the Alternate Index by identifying the fields that are keys of the Data Table keys, unique Data Table Keys can be enforced.

3. Set the Duplicate Prime Keys field on the **Edit Supplementary Information** screen, Option 2 - DEFINE VIEW SUPPLMT on the Define Table and View menu, to N.

Restructure Table

Once you have modified a View by inserting new fields, expanding or reducing the length of existing fields, or deleting existing fields, the Restructure Table option will restructure the data accordingly. If you have expanded a date field to accommodate the century by changing a field's format from YY to YYYY, the following rules apply:

1. If the value of YY is 82 or greater, the field will be converted to 19YY.

2. If the value of YY is less than 82, the field will be converted to 20YY.

To modify an existing table, you must use the Define All option from the Define Table menu. Once you have modified the fields in your View, you will automatically be moved to the **Data Restructuring Utility** screen (see Figure 4-10). This is only necessary if the View relates to a Data Table that already exists.

```
tablesONLINE 6.0.2 Administrator ------ --- Data Restructuring Utility --------
COMMAND ===>




                          Restructure data in a table
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DATA.TABLE.LIBRARY
     TABLE NAME           : SAMPLE
     PASSWORD             :
     ALL GENERATIONS      : Y  Enter N to restructure most current generation.


     RESTRUCTURE TABLE    :           If blank, enter a table name if the
                                      restructuring table is to be used later.
                                      If not blank, change as desired.
```

Figure 4-10: Data Restructuring Utility Screen

When this screen is displayed, the library name and table name will already be inserted based on the View you have just edited. If the table is password protected, you must supply the write password before this operation can be performed. You may also specify whether you want to restructure all generations of the table or only the current generation using the All Generations field. The default value for this field is Y to restructure all generations.

If you do not wish to restructure the Data Table associated with the modified View immediately, you may insert a restructure table name to identify the restructure table so that it may be used at a later date. Inserting a name in the Restructure Table field will create a table containing the rules required by tablesONLINE to restructure your Data Table at a later date.

Note: If you have multiple Data Tables that are associated with the same View, you should save the Restructure Table for later use. In this case, you will need to perform the restructuring process, using this Restructure Table, for each Data Table associated with the modified View.

Once you have completed these fields in the **Data Restructuring Utility** screen, press <PF2> to initiate the restructuring process.

Using a Saved Restructure Table

To use a saved Restructure Table to restructure a Data Table, select option 7-RESTRUCTURE TABLE from the Define Table and View menu. This will open the **Data Restructuring Utility** screen. Insert the name of the library containing the table to be restructured, the name of the table to be restructured and the write password for that table, if there is one. You must, once again, indicate whether you wish to restructure all generations of the table or only the most recent. Finally, insert the name of the saved Restructure Table and press <PF2>. This will restructure the data in your table to accommodate the new row layout. This process should be repeated for all Data Tables associated with the modified View.

Generate Copybook

The Generate Copybook option of the Define Table and View menu allows you to generate COBOL copybooks when a new View is created or an existing View is modified. To generate a copybook for a View, select Option 8 - GENERATE COPYBOOK from the Define Table And View menu to display the following screen and complete the fields described in Table 4-27.


```
tablesONLINE 6.0.2 Administrator ------ Generate Copybook from View ----------
COMMAND ===>




                        Generate Copybook from VIEW table
              TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DATA.TABLE.LIBRARY
     VIEW NAME            : EXAMPLE

     INCLUDE COMMAND AREA : N
     FIELD NAME PREFIX    : W-

     COPYLIB DATASET NAME : DKLTBT.V5R1M0.O27.CPY
     MEMBER NAME          : EXAMPLE

     JOB NAME             : DKLD01TP
     ACCOUNTING INFO      : DKLD01,DKLSCS090
     NOTIFY               : DKLD01
     JCL TO MAKE COPYBOOK : TBOLJCL1
```

Figure 4-11: Generate Copybook Screen

Table 4-27: Generate Copybook Field Descriptions

Field Names	Description
Library Name	Insert the name of the library where the View is kept. You may also enter a number from 1-7 to select a relative library number, a DDNAME or a full dataset name. Pressing <PF1> for help in this field displays a list of all authorized libraries.
View Name	Insert the name of the View for which you wish to create a copybook. Pressing <PF1> for help in this field displays a list of all authorized Views.
Include Command Area	This field indicates whether or not the command area is to be included at the end of the COBOL copybook. It can be set to Y to include the command area or N to exclude it.
Field Name Prefix	This field contains a prefix that will be placed in front of all field names in the COBOL FD (for example, `WS-TBXX-`)
Copylib Dataset Name	This field contains the COPYLIB dataset name that contains the FD copybooks available to application developers for the compilation of application programs. This is where the copybook will be placed.

Table 4-27: Generate Copybook Field Descriptions (Continued)

Field Names	Description
Member Name	This field contains the name of the member in the Copylib that will contain the created copybook.
Job Name	This field contains the job name that will appear on the job statement of the batch job that creates the a COBOL copybook.
Accounting Info	This field contains the accounting information that will appear on the job statement of the batch job that creates the COBOL copybook.
Notify	This field contains the notify (User-ID) that will appear on the job statement of the batch job that creates the COBOL copybook.
JCL To Make Copybook	This is the name of the table that contains the skeleton JCL to be submitted to the internal reader when a COBOL FD is created. The table that is delivered for this purpose is TBOLJCL1. It can be modified for your own installation.

Generate C Struct

The Generate C Struct option allows you to generate C structure when a new View is created or an existing View is modified. To generate a C structure for a View, select Option 8C - GENERATE C STRUCT from the Define Table menu to display the **Generate C Struct** screen (see Figure 4-12), and complete the fields described in Table 4-28.

```
tablesONLINE 6.0.2 Administrator ------ Generate C Structure from View --------
COMMAND ===>




                          Generate C Structure from View
                TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : TBDEV.CUST.DKL.V600.MAINLIB
     VIEW NAME             : EXAMPLE

     STRUCTURE NAME        : EXAMPLE-C

     INCLUDE FILE NAME     : TBSUP.CUST.NWML.CNTL
     MEMBER NAME           : EXAMPLE

     JOB NAME              : DKLD01
     ACCOUNTING INFO       : DKLD01,DKLSCS090
     NOTIFY                : &SYSUID
     JCL TO MAKE C STRUCT  : TBOLJCL4
```

Figure 4-12: Generate C Struct Screen

Table 4-28: Generate C Struct field Description

Field Name	Description
Library Name	Insert the name of the library where the View is kept. You may also enter a number from 1 to 7 to select a relative library number, a DDNAME or a full dataset name. Pressing <PF1> for help in this field will display a list of all authorized libraries.
View Name	Insert the name of the View for which you wish to create a C structure. Pressing <PF1> for help in this field displays a list of all authorized Views.
Structure Name	This field contains a prefix that will be placed in the C structure.

Table 4-28: Generate C Struct field Description (Continued)

Include File Name	This field contains the Include File dataset name that contains the C structures available to application developers for the compilation of application programs. This is where the C structure will be placed.
Member Name	This field contains the name of the member in the Include File that will contain the created C structure.
Job Name	This field contains the job name that will appear on the job statement of the batch job that creates the C structure.
Accounting Info	This field contains the accounting information that will appear on the job statement of the batch job that creates the C structure.
Notify	This field contains the notify (User-ID) that will appear on the job statement of the batch job that creates the C structure.
JCL To Make C structure	This is the name of the table that contains the skeleton JCL to be submitted to the internal reader when a C structure is created. The table that is delivered for this purpose is TBOLJCL4. It can be modified for your own installation.

Define Many To Many (M2M)

Using the Define Many To Many (M2M) option it is possible to define multiple relationships between tables and Views within tablesONLINE. The M2M facility allows you to do this, and to subsequently make use of these defined relationships. Defining an M2M relationship involves making an entry in the M2M table to associate a specific Data Table with a specific View and assign a unique Table Object name to that Data Table/ View combination. Once that combination has been defined, a user does not have to know the name of the View or the name of the Data Table, it can simply be referred to using the Table Object name. tablesONLINE automatically searches the M2M table for an entry for the Table Object name, which indicates the table to be accessed and the appropriate View to be used.

To define an M2M relationship, select Option 9 - DEFINE M2M from the Define Table and View menu (see Figure 4-1). The system displays the Edit M2M Table-Identify Screen (see Figure 4-13).


```
tablesONLINE 6.0.2 Administrator ------ Edit -------------- Identify Table/Row
COMMAND ===>

Please indicate the table required by entering the parameters below.
 Data Table Library          ==> DICTIONARY.LIBRARY
 View Library (If different) ==> SYSTEM.TABLES
 Table Object==> TBOLM2M
                                                    Generation ==> 5

Enter row key for direct search or positioning on the edit table screen.

TABLE OBJECT         ===>
```

Figure 4-13: The M2M Table - Identify Screen

This screen will already contain the Data Table Library, View Library and Table Object names for the table that contains the M2M information. If you wish to edit the information for an existing entry in the M2M table, you can insert the Table Object name in the Table Object field. This displays the **Edit-Row** screen for the selected Table Object. If no Table Object is specified in this field, the following **Edit-Table** screen will be displayed (see Figure 4-14).

From this screen (see Figure 4-14) you may select a row to edit or create a new Table Object. To create a new Table Object, place the N line command to the left of the row you wish to use as a template for the new entry and press <Enter>. This will display the **Edit-Row** screen. Here you must insert the name of the Table Object to be defined, as well as the names of the Data Table and associated View that are to be identified by that Table Object name. Once these fields have been completed, press <PF2> or <End>. You may continue to define additional entries, or you can use the <End> key to exit the **Edit-Row** screen and return to the menu.

```
tablesONLINE 6.0.2 Administrator ------ Edit Table --------------------------
COMMAND ===>

Table Name : TBOLM2M                                    Row Location : 1

      TABLE     VIEW     DATA
      OBJECT    NAME    /INDEX
     ________ ________ __NAME__
     OBJ1     TBOLMENU PAYRMENU
     OBJ2     EMPLOYEE EMPLOYE2
     OBJ3     EMPLOYEE EMPLOYE3
```

Figure 4-14: The M2M Table - Edit-Table Screen

Note: 1. An entry is not required in the M2M table if: the relationship between the View and the Data Table is one-to-one, and the View and Data share the same name. This is the most common use of tablesONLINE.
2. Entries in the M2M table to describe many Views for one Data Table are not needed. One can simply specify the same Data Table Name in each Supplementary View Information described earlier in this chapter.

M2M Name Resolution

When the M2M facility is turned on (the default setting) the name entered in the Table Object field is resolved in the following sequence:

1. The M2M table is searched. If an entry in this table is found that matches the entered Table Object name, then the Data Table and View names are obtained from this entry.

2. Otherwise, the available tableBASE libraries are searched for a View with the same name as the entered Table Object name.

3. If a View is found in Step 2, and that View contains the name of the Data Table it is explicitly associated with, then that name is used to locate the Data Table.

4. Otherwise, if a View is found in Step 2, but it does not explicitly contain the associated Data Table name, then the Table Object name entered by the user is used to locate the Data Table.

5. Otherwise, the available tableBASE libraries are searched for a Data Table with the same name as the entered Table Object name. If the View name is specified within the Data Table definition, then that name is used to locate the View. Otherwise, an error message will be issued.

When M2M is turned off, separate fields appear so that both the View name and the Data Table name may be specified explicitly by the user.

1. However, if the View name is cleared, then, the Data Table is located using the entered Data Table name.

   If the Data Table's definition names the associated View, then that name is used to locate the View. Otherwise, the View with the same name as the Data Table is located.

2. If the Data Table name is cleared, then, the View is located using the entered View name.

   If the View explicitly names the associated Data Table, then that name is used to locate the Data Table. Otherwise, the Data Table with the same name as the View is located.

Print View

By selecting Option P in the Define Table and View menu (see Figure 4-1) you will be presented with the Print screen (see Figure 4-15). As is indicated, the screen enables you to obtain a hard copy report of a particular View.

```
tablesONLINE 6.0.2 Administrator ------ Print a View -------------------------
COMMAND ===>




                          Submit a Batch Job to Print a VIEW
                TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : DATA.TABLE.LIBRARY
     VIEW NAME             : EXAMPLE

     ALPHABETIC ORDER?     : N   Enter 'Y' to print the fields alphabetically.

     JOB NAME              : DKLD01TP
     ACCOUNTING INFO       : DKLD01,DKLSCS090
     NOTIFY                : DKLD01
     JCL TO PRINT VIEW     : TBOLJCL3
```

Figure 4-15: Print View Screen

Table 4-29 provides a description of the fields of the Print View screen.

Table 4-29: Print View Screen Field Definitions

Field Name	Usage Description
Library Name	Insert the name of the library where the View is kept. You may also enter a number from 1-7 to select a relative library number, a DDNAME, or a full dataset name. Pressing <PF1> in this field will display a list of all authorized libraries.
View Name	Insert the name of the View for which you wish to create a copybook. Pressing <PF1> in this field will display a list of all authorized Views.
Alphabetic Order?	If you wish the fields defined in the View to be printed in alphabetic sequence enter Y. The default sequence, or N, is to print the fields in the sequence in which they appear in the View.
Job Name	This field contains the job name that will appear on the job statement of the batch job to print the View.

Table 4-29: Print View Screen Field Definitions (Continued)

Field Name	Usage Description
Accounting Info	This field contains the accounting information that will appear on the job statement of the batch job to print the View.
Notify	This field contains the notify (User-ID) that will appear on the job statement of the batch job to print the View.
JCL To Print View	This is the name of the table that contains the skeleton JCL to be submitted to the internal reader to print the View. The table that is delivered for this purpose is TBOLJCL3. It can be modified for your own installation.

5 Utilities

The tablesONLINE/CICS software provides you with a variety of utilities that can be used with your tableBASE tables. These utilities allow you to copy, delete and rename both Data Tables and Views. Utilities are also provided to allow you to change the password on a table and edit your User Profile.

To display the menu containing the available utility functions, select the Option U - UTILITIES, from the Application Developer's menu. This will display the **Utility Menu** screen (see Figure 5-1).

```
tablesONLINE 6.0.2 Administrator ------ Utility Menu -------------------------
COMMAND ===>


To select, enter number/symbol on command line:

P    PRINT TABLE          - Submit a Batch Job to Print contents of a table
1    COPY TABLE           - Copy a Data Table (To Another Library - Optional)
2    COPY VIEW            - Copy a View (To Another Library - Optional)
3    DELETE TABLE         - Delete a Generation of a Table
4    DELETE VIEW          - Delete a View
5    RENAME TABLE         - Change the Data Table Name
6    RENAME VIEW          - Change the View Name
7    CHANGE PASSWORDS     - Change Either the Read or Write Password
8    WRITE PROTECT VIEW   - Place a Write Password on a View
9    EDIT PROFILE         - Edit User Profile



Enter HELP at any stage for help within tablesONLINE.
Enter PF for program function key assignments.
Enter X to suspend tablesONLINE and return to CICS.
```

Figure 5-1: Utility Menu Screen

Each of these options is described in detail on the following pages. You should be aware of the following points before proceeding:

- Wherever a Password field appears on a screen, a password must be given if the table has one. Whether the read or write password is required will depend on the operation to be performed.
- Several of the utility screens have a Library Name field. You are authorized to operate only in libraries that are included on your library list. This list is retrieved from the Application Control Table when you sign on. The only way to change the list is to edit that table, often a privilege reserved for the system administrator. If changes are made to your library list, you must leave tablesONLINE and sign on again to invoke the new list.
- For any Library field, you can display a list of the available libraries by moving the cursor into the library field and pressing <PF1> for help. If you do not know the name of the library where a specific table or View resides, insert the table or View name in the appropriate field, clear the library field, and press <Enter> to automatically insert the appropriate library name in the field.
- In addition to the List Facility described above, there are four ways to specify a library name:
    - insert a DDNAME
    - insert a Dataset Name
    - specify a library alias
    - specify a number from 1-7

While a library can be identified by a DDNAME or Dataset Name, tablesONLINE provides two additional methods to specify a library. You can set up aliases for your tablesONLINE libraries that can then be used to reference that library. An alias is simply another name for a library that can be used in tablesONLINE by a user more easily than the library Dataset Name. These aliases are stored in the Application Driving Table TBOLLIBR.

Or, you can identify a library in a Library Name field by inserting a number from 1-7 in the field. The number refers to the position of the library in the library list for the application, as defined in the Application Control Table (ACT). When an application is initially set up by the tableBASE administrator, an entry is made in the ACT. The ACT includes seven fields that are used to identify the libraries that will be used by that application. The relative position of a library in that list can be used to identify a library in the Library Name field.

Print Table

To print a formatted report of a table or View via a batch job, select P - PRINT TABLE, from the Utilities menu. This will display the **Print Table** screen (see Figure 5-2).

```
tablesONLINE 6.0.2 Administrator ------ Print the contents of a Table --------
COMMAND ===>




                          Submit a Batch Job to Print a Table
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : DATA.TABLE.LIBRARY
     TABLE NAME            : EXAMPLE
     PASSWORD              :
     GENERATION            : 0
                                  If VIEW is in same library, leave blank.
     VIEW LIBRARY (OPT)    :
     VIEW NAME             :           Used when a different View is required.

     REPORT TITLE          : HYPHENS-DISAPPEAR-ON-REPORT
     FIELDS & OVERRIDES    :
     NO OF REF FIELDS      :

     START KEY             :
     END KEY               :

     START ROW NUMBER      : 0
     NUMBER OF ROWS        :

     JOB NAME              : DKLD01TP
     ACCOUNTING INFO       : DKLD01,DKLSCS090
     NOTIFY                : DKLD01
     JCL TO PRINT TABLE    : TBOLJCL2
```

Figure 5-2: Print Table Screen

Identify the Table

Insert the name of the table you wish to print and the name of the library in which it resides. If the table is protected by a password, you must insert the password in the Password field. Specify the table generation to print; by default, the current generation is used.

If you wish to print a table using a particular View specify the View name and if the View is not in the current library, the name of the View Library.

Format the Report

The next three fields — Report Title, Fields & Overrides, and No of Ref Fields — control the format of the report. If you wish to include embedded spaces in your report title, enter hyphens (-) in the appropriate place. A report title has a maximum length of 50 characters.

You control which fields are printed and in what sequence using the Fields & Overrides parameter. In this parameter you supply the field name(s) to be selected from the View for printing. The default is to print all fields. Use double quotes to enclose field names that contain embedded blanks.

Example: FIELDS=FIRST, SECOND, "NEW CODE", "OLD CODE"

Field names may be selected with wildcards by placing an asterisk (*****) after the significant characters of the field name. All fields beginning with the significant characters (generic portion of the field name) will be printed. An asterisk (*****) by itself will select all fields.

Print all fields beginning with D.

Example: FIELDS=D*

Print all fields and repeat the two keys again.

Example: FIELDS=*,KEY1,KEY2

The first occurrence of a generic field name may be printed by using an exclamation mark (!) after the significant character(s). To select a number of occurrences repeat the significant character(s) and the exclamation character a number of times.

Print the first two fields beginning with D.

Example: FIELDS=D!,D!

Note: All fields in a View with ATTRIBUTE=SUPPRESS may be printed only by being explicitly named or by generic selection.

In order to identify the fields that overflow onto a second or subsequent page, the No of Ref Fields is used to specify the number of fields to be reprinted on the left side of the second page containing the overflow field columns that do not fit on page one. The default is zero, indicating that no fields are to be repeated.

Print a Subset of a Table

You can print sections of the Data Table or Alternate Index by specifying the starting and ending key fields. If you specify a start key and no end key, all rows beginning with

the starting key are printed. You may also specify the row in a Data Table — from which reporting is to begin — and the number of rows to print.

Specify JCL

The remaining four fields are used to specify the JCL to be submitted to the internal reader as follows:

Table 5-1: Specify JCL

Field Names	Description
Job Name	This field contains the job name that will appear on the job statement of the batch job to print the table or View.
Accounting Info	This field contains the accounting information that will appear on the job statement of the batch job to print the table or View.
Notify	This field contains the Notify UserID that will appear on the job statement of the batch job to print the table or View.
JCL To Print Table	This is the name of the table that contains the skeleton JCL to be submitted to the internal reader when a table or View is to be printed. The table that is delivered for this purpose is TBOLJCL2. It can be modified for your own installation.

Copy Table

The Copy Table utility allows you to copy an existing table within the same library or to another library. To copy a table, select Option 1 - COPY TABLE from the **Utility** menu to proceed to the **Copy Table** screen (see Figure 5-3).

```
tablesONLINE 6.0.2 Administrator ------ Copy Table ---------------------------
COMMAND ===>




                          Copy a Table to the same or a different Library
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : APPLICATIONS.DRIVING.TABLES
     TABLE NAME           : TBOLMENU
     PASSWORD             :
     GENERATION           : 0

                               If copy is to same library,
                              leave new library name blank.

     NEW LIBRARY NAME     :
     NEW TABLE NAME       :           Enter new name if copy to same library.

     REPLACE OPTION       : N
     PASSWORD (TARGET)    :
```

Figure 5-3: Copy Table Screen

Insert the name of the table you wish to copy and the name of the library in which it resides. If the table is protected by a password, you must insert the password in the Password field. Only one generation of the table is copied to the new table. By default this is the current generation. If you wish to copy a previous generation, simply change the Generation field to the number of the desired generation. To copy the table within a library, leave the New Library Name field blank and enter a New Table Name. To copy the table to another library, enter the new library name. You do not have to insert a new table name if you wish the copied table to have the same name as the original table.

If you wish to change the table name, simply insert the new name in the New Table Name field. If the new table name already exists in the library you are copying the table to, you have the option of replacing the existing table. To do this, you must set the Replace Option to Y and supply the password for the target table, if it is password protected.

Note: If you have inserted a table name and relied on tablesONLINE to insert the proper library name in the Library Name field, you should press <Enter> before pressing <PF2>. This allows you to verify that the defaults have been correctly resolved before the copy command is executed.

Once you have completed all of the necessary information in this screen, press <PF2>.

Copy View

The Copy View utility allows you to copy an existing View to the same library or to another library. To copy a View, select Option 2 - COPY VIEW from the **Utility** menu. This will display the **Copy View** screen (see Figure 5-4).

```
tablesONLINE 6.0.2 Administrator ------ Copy View -----------------------------
COMMAND ===>




                          Copy a VIEW to the same or a different Library
              TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DATA.TABLE.LIBRARY
     VIEW NAME            : EXAMPLE
     PASSWORD             :

                                If copy is to same library,
                               leave new library name blank.

     NEW LIBRARY NAME     : DATA.TABLE.LIBRARY
     NEW VIEW NAME        : EXAMFRST  Enter new name if copy to same library.

     REPLACE OPTION       : Y
     PASSWORD (TARGET)    :
```

Figure 5-4: Copy View Screen

Insert the name of the View you wish to copy and the name of the library in which it resides. If the View is protected by a password, you must insert the password in the Password field. To copy the View within a library, leave the New Library Name field blank and enter a new View name. To copy the View to another library, enter the new library name. You do not have to insert a new View name if you wish the copied View to

have the same name as the original View. If you wish to change the View name, simply insert the new name in the New View Name field.

If the new View name already exists on the library you are copying the View to, you have the option of replacing the existing View. To do this, you must set the Replace Option to Y and supply the password for the target View if it is password protected.

Note: If you have inserted a View name and relied on tablesONLINE to insert the proper library name in the Library Name field, you should press <Enter> before pressing <PF2>. This allows you to verify that the defaults have been correctly resolved before the copy command is executed.

Once you have completed all of the necessary information in this screen, press <PF2>.

Delete Table

The Delete Table utility allows you to delete one generation of a table. If you wish to delete all generations of a table, you must delete each generation individually using this utility. This utility is often used to remove a corrupt generation of a table and revert to an earlier version.

To delete a table, select Option 3 - DELETE TABLE from the **Utility** menu. This will display the **Delete Table** screen (see Figure 5-5).

```
tablesONLINE 6.0.2 Administrator ------ Delete Table -------------------------
COMMAND ===>




                          Delete a generation of a Table
                TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : DATA.TABLE.LIBRARY
     TABLE NAME            : EXAMPLE
     PASSWORD              :
     GENERATION            : 0
```

Figure 5-5: Delete Table Screen

Insert the name of the table you wish to delete and the name of the library in which it resides. If the table is protected by a password, you must insert the password in the Password field. Insert the generation of the table you wish to delete. If a generation is not specified, tablesONLINE will delete the current generation.

Once you have completed all of the necessary information in this screen, press <PF2> to delete the specified table generation.

If you wish to delete more than one generation of a table, successive presses of <PF2> will delete generations, one deletion for each use of <PF2>, until no further generations exist.

Note: If you have inserted a table name and relied on tablesONLINE to insert the proper library name in the Library Name field, you should press <Enter> before pressing <PF2>. This allows you to verify that the defaults have been correctly resolved before the delete command is executed.

Delete View

The Delete View utility allows you to delete a View. If you wish to delete all generations of a View, you must delete each generation individually using this utility. This utility is often used to remove a corrupt generation of a View and revert to an earlier version.

To delete a View, select Option 4 - DELETE VIEW from the **Utility** menu. The system displays the **Delete View** screen (see Figure 5-6).

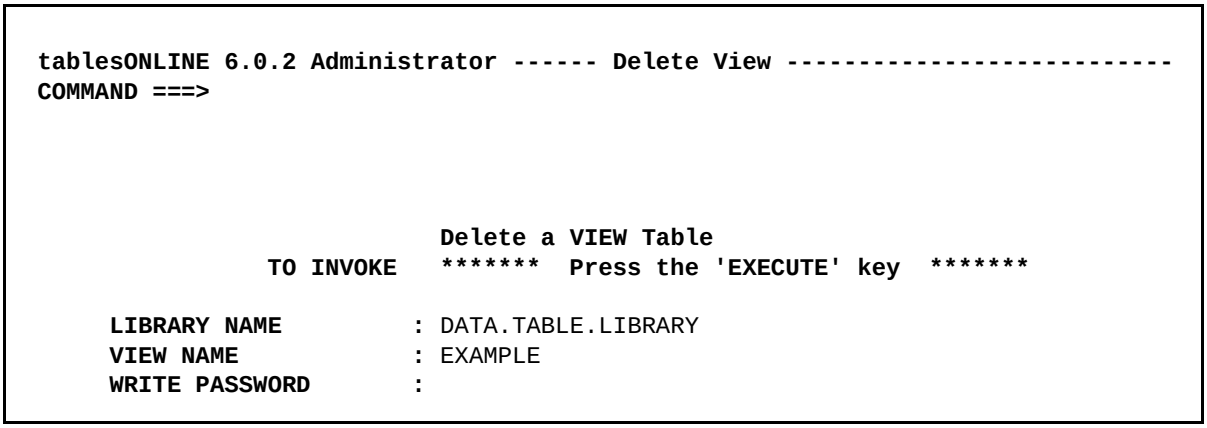

Figure 5-6: Delete View Screen

Insert the name of the View you wish to delete and the name of the library in which it resides. If the View is protected by a password, you must insert the password in the password field.

Once you have completed all of the necessary information in this screen, press <PF2> to delete the specified View.

Note: If you have inserted a View name and relied on tablesONLINE to insert the proper library name in the Library Name field, you should press <Enter> before pressing <PF2>. This allows you to verify that the defaults have been correctly resolved before the delete command is executed.

Rename Table

The Rename Table utility allows you to rename a Data Table. To rename a table, select Option 5 - RENAME TABLE from the **Utility** menu, to proceed to the **Rename Table** screen (see Figure 5-7).

```
tablesONLINE 6.0.2 Administrator ------ Rename a Table -----------------------
COMMAND ===>




                         Change the Data Table Name
                TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : DATA.TABLE.LIBRARY
     TABLE NAME            : EXAMPLE
     PASSWORD              :

     NEW NAME              : EXAMPLE
```

Figure 5-7: Rename Table Screen

Insert the name of the table you wish to rename and the name of the library in which it resides. If the table is protected by a password, you must insert the password in the Password field. Insert the new name that you wish to assign to the table.

Once you have completed all of the necessary information on this screen, press <PF2> to rename the specified table generation.

Rename View

The Rename View utility allows you to rename a View. To rename a View, select Option 6 - RENAME VIEW from the **Utility** menu. This will display the **Rename View** screen (see Figure 5-8).

```
tablesONLINE 6.0.2 Administrator ------ Rename View --------------------------
COMMAND ===>




                          Change the View Name
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : DATA.TABLE.LIBRARY
     VIEW NAME             : EXAMPLE
     WRITE PASSWORD        :

     NEW NAME              : NEWEXAM
```

Figure 5-8: Rename View Screen

Insert the name of the View you wish to rename and the name of the library in which it resides. If the View is protected by a password, you must insert the password in the Password field. Insert the new name that you wish to assign to the View.

Once you have completed all of the necessary information on this screen, press <PF2> to rename the specified View.

Change Passwords

The Change Password utility allows you to change the read and write passwords for a Data Table. To change the password for a table, select Option 7 - CHANGE PASSWORDS from the **Utility** menu. This will display the **Change Table Password** screen (see Figure 5-9).

```
tablesONLINE 6.0.2 Administrator ------ Change Table Password -----------------
COMMAND ===>




                          Change either the READ or WRITE password
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME          : DATA.TABLE.LIBRARY
     TABLE NAME            : EXAMPLE
     WRITE PASSWORD        :

                                Enter * to Remove Passwords
     NEW READ PASSWORD     :
     NEW WRITE PASSWORD    :
```

Figure 5-9: Change Table Password Screen

Insert the name of the table you wish to change the password for, and the name of the library in which it resides. If the table is already protected by a write password, you must type the password in the Write Password field. Insert the new read password or the new write password, or both, in the appropriate fields. If you wish to remove an existing password from the table, insert an asterisk (*) in the appropriate field.

Once you have completed all of the necessary information on this screen press <PF2> to change the password.

A new generation of the table is created which can be accessed using the new passwords.

Write Protect View

The Write Protect View utility allows you to assign a write password for a View. It is not possible to assign a read password to a View. To assign a write password for a View, select Option 8 - WRITE PROTECT VIEW from the **Utility** menu (see Figure 5-1). The system displays the **Change View Password** screen (see Figure 5-10).

```
tablesONLINE 6.0.2 Administrator ------ Change View Password ------------------
COMMAND ===>




                          Place a WRITE password on a View
               TO INVOKE   *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DATA.TABLE.LIBRARY
     VIEW NAME            : EXAMPLE
     WRITE PASSWORD       :

                                Enter * to Remove Passwords
     NEW WRITE PASSWORD   :
```

Figure 5-10: Change View Password Screen

Insert the name of the View you wish to assign a password to, and the name of the library in which it resides. If the View is already protected by a write password, you must insert the current password in the Write Password field. Insert the new write password in the appropriate field. If you wish to remove an existing password from the table, insert an asterisk (*) in the New Write Password field.

Once you have completed all of the necessary information on this screen, press <PF2> to assign the write password to the View. A new generation of the View is created which can be accessed using the new password.

Edit Profile

The User Profile table contains an entry for each user, that specifies the environmental variables for that user. Initially the values in this table are taken from a default entry created by your system administrator. Each time you use tablesONLINE, the information stored in this table is updated by tablesONLINE based on the operations performed during your session. It is the information stored in this table that allows tablesONLINE to start your next session using the same environment as your previous session. You can change various entries in your own profile using the Edit Profile utility.

To edit your User Profile, select Option 9 - EDIT PROFILE from the **Utility** menu (see Figure 5-1). This will display the **Update User Profile** screen.

```
tablesONLINE 6.0.2 Administrator ------ Update User Profile ------------------
COMMAND ===>




                          Edit your User Profile

     TABLE CONFIRM SAVE   : Y
     ROW CONFIRM DELETE   : N
     TRANSFER TBLNAME IND : Y
     SUPPRESS INFOMSG IND : N
     TABLE COPY REPL IND  : Y
     SUPPRESS MSG ID IND  : N

     JOB NAME             : DKLD01TP
     ACCOUNTING INFO      : DKLD01,DKLSCS090
     NOTIFY               : DKLD01
     COPYLIB DATASET NAME : DKLTBT.V5R1M0.O27.CPY
     MEMBER NAME          : TBUTILSL
     FIELD NAME PREFIX    : W-
     INCLUDE COMMAND AREA : N
     JCL TO MAKE COPYBOOK : TBOLJCL1
++++ JCL TO PRINT TABLE   : TBOLJCL2
```

Figure 5-11: Update User Profile Screen

As indicated by the ++++ at the bottom of the screen, the profile table spans more than one screen. Navigate through the screens by using the <PF7> and <PF8> keys to go up and down. Once you have completed all of the necessary information on this screens, press <PF2> to update the User Profile.

Each of the fields contained in these screens is described in . Many of the fields in this table are set automatically by tablesONLINE based on the actions of the user. In the following subparagraphs, the fields that are automatically set by tablesONLINE are indicated by an asterisk.

Table Confirm Save

The first field controls whether tablesONLINE asks for confirmation before saving a table. If the switch is set to Y, users will be asked for confirmation before the save operation is performed. If the switch is set to N, users will not be asked for confirmation before the table is saved.

Row Confirm Delete

The second field controls whether tablesONLINE asks for confirmation before deleting a row. If the switch is set to Y, users will be asked for confirmation before the delete operation is performed. If the switch is set to N, users will not be asked for confirmation before the row is deleted.

Transfer TBLNAME Indicator

This field controls whether tablesONLINE changes its internal record of the current table name when a rename or copy operation is done. For example, if you are editing table AAA, move to the utilities to rename table AAA to BBB and then return to the editor, the initial table name that appears on the **Select Table/Row** screen will be BBB if this indicator is Y and AAA if this indicator is N.

Suppress INFOMSG Indicator

This field is used to suppress the display by tablesONLINE of messages whose message type field in the message table is I, indicating an information message. If this field is set to Y, informational messages are suppressed. Messages whose message type is error, warning, or abend are not affected by this indicator. These messages are displayed regardless of the setting of this indicator.

Table Copy/Replace Indicator

This field indicates whether or not existing tables are replaced during a copy operation that specifies an existing table. If this indicator is set to Y, it is possible to copy a table to a location where a table with the same name exists and overwrite the existing table with the copied table. If this indicator is set to N, an operation that copies a table to a location where a table with the same name exists will result in an error and the table will not be overwritten.

Suppress Message ID Indicator

This field is used to suppress all message identification. If the indicator is set to Y, only the text of messages will appear. If the indicator is set to N, the text and an identifying code for the message will appear.

Job Name

This field contains the job name that will appear on the job statement of the batch job to create a COBOL copybook for a View.

Accounting Info

This field contains the accounting information that will appear on the job statement of the batch job to create a COBOL copybook for a View.

Notify

This field contains the Notify UserID that will appear on the job statement of the batch job currently used for the creation of a COBOL copybook for a View.

Copylib Dataset Name

This field contains the COPYLIB dataset name that contains the FD copybooks available to application developers for the compilation of application programs.

Member Name

This field contains the name of the member in the COPYLIB that will contain the copybook.

Field Name Prefix

This field contains a prefix that will be placed in front of all field names in the COBOL FD, for example, `WS-TBXX-`.

Include Command Area

This field indicates whether or not the command area is to be included at the end of the COBOL copybook. It can be set to Y to include the command area or N to exclude it.

JCL To Make Copybook

This is the name of the table that contains the skeleton JCL to be submitted to the internal reader when a COBOL copybook is created. The delivered default is TBOLJCL1.

JCL To Print Table

This is the name of the table that contains the skeleton JCL to be submitted to the internal reader when a table is printed. The delivered default is TBOLJCL2.

JCL To Print View

This is the name of the table that contains the skeleton JCL to be submitted to the internal reader when a View is printed. The delivered default is TBOLJCL3.

Extended Attributes

There are a series of extended attribute fields for use in controlling the appearance of color terminals. These fields should not be modified if any of the following conditions apply to your system.

- Your terminals do not support these attributes
- Your CICS installation does not support these attributes
- You are uncertain about the above conditions

If you wish to make modifications to these extended attributes, experimentation is the best way to identify the function each performs, and, which effects are possible.

Caution: tablesONLINE does no error checking on these fields, it simply passes the data to CICS as extended display attributes. You must know what attributes your terminals and CICS support before attempting any experimentation. Attempting to use unsupported attributes may cause unpredictable and undesirable behavior from the terminal.

Freeze Keys Count*

This field indicates the number of fields that are currently frozen on the screen. Frozen fields will always be displayed on the screen when fields are being scrolled left and right.

Edit/Browse DBTYPE

This field indicates the type of database that is being accessed. This field will be set to TABLE.

Edit/Browse Library*

This field indicates the current tableBASE library for the Data Table that is being browsed or edited.

Edit/Browse Object*

This field indicates the current Table Object that is being browsed or edited.

Edit/Browse View Library*

This field indicates the current tableBASE library for the View table that is being browsed or edited.

Edit/Browse View*

This field indicates the current View that is being browsed or edited.

Utility To Library*

This field indicates the library used as the target library, for a table using a utility.

Utility To Table*

This field indicates the table used as the target table for a utility.

Utility To View Library*

This field indicates the library used as the target library for a View using a utility.

Utility To View*

This field indicates the View used as the target View for a utility.

Restructure Table*

This field will contain the name of a restructuring table if it has been saved for later use.

Restructure All Generations Indicator

If this field is set to Y, it indicates that all generations of a table have been restructured. If set to N, it indicates that only the most recent generation was restructured.

* Field automatically set by tablesONLINE.

6
Building Applications

tablesONLINE is both a CICS transaction program and a table-driven program. The code is fully re-entrant, enabling many users to simultaneously use one copy of the code since the system creates a separate data area for each user. tableBASE tables can readily be shared, a feature that has been used throughout the product so common data is not duplicated.

Using this type of approach offers a number of advantages. It allows you to conserve region since multiple copies of the same information are not needed. One copy of common data and one tested, centralized method of accessing it also has a positive affect on reliability and security. Finally, system changes - maintenance, customization, performance tuning and installation - are simplified.

Overview

tablesONLINE allows you to build applications that are table-driven. To create or customize an application you will take the seven tables that are used by tablesONLINE to drive an application and customize them to your needs. This chapter describes the seven Application Driving Tables and explains the editing process for each of these tables.

The following is a list of the Application Driving Tables required for an application together with a brief description of the content of each of these tables. These tables are stored in the Application Driving Tables Library.

The table names listed in Table 6-1 include a four-character application identifier, indicated by XXXX, which is unique for each application. For example, for a payroll application you, may wish to use an identifier of PAYR. In that case, your Application Driving Tables will be called PAYRMENU, PAYRPFKS, PAYRCMDS, and so on.

Table 6-1: Application Driving Tables

Application Driving Table Name	Description
XXXXMENU	Contains customized menu entries for the application. This controls the flow of your application.
XXXXPFKS	Contains customized PF Key assignments for the application.
XXXXCMDS	Contains the command translation information for the application. This enables you to define aliases for commands.
XXXXDESC	Contains screen descriptions for the application. This lets you modify some textual lines on the screens.
XXXXMSGS	Contains customized messages for the application.
XXXXHELP	Contains customized help for the application.
XXXXLIBR	Contains the DSN to DDNAME translation for the application.

Each application, when started up, will pick up the names of its controlling tables from the Application Control Table. Once started up, the application is guided by those tables.

Note: Special Views are used in the editing process for the Application Driving Tables. These Views ensure that the system knows how each table is set up. Much data validation and error-checking is performed automatically when these tables are edited.

Creating a New Application

To create an application, you must perform the following steps:

- copy an existing set of Application Driving Tables
- create entries in the Application Control Table for the application
- write any exit programs required
- edit the copied tables for the new application

In the delivered defaults of tablesONLINE, two of these functions are functions that are reserved for the tableBASE administrator. These functions are copying the Application Driving Tables and creating entries in the Application Control Table. These functions are documented in the tableBASE Administration Guide. Unless you have been granted permission to use these functions the tableBASE administrator will have to copy the Application Driving Tables and create entries for them in the Application Control Table before you will be able to begin editing the tables for your application.

The third function on this list, writing exit programs, is described in the tableBASE Programming Guide.

The final function on this list, editing the copied tables for the new application, is the focus of the balance of this chapter. We will discuss the use of each of the individual Application Driving Tables and how they are edited.

Note: For the purposes of this guide we will be using the default tablesONLINE driving tables as templates for creating a new application. Your tables may have been modified.

Selecting a Table

Once they have been copied, the Application Driving Tables are selected for edit from the **Application Developer's** menu (see Figure 6-1).

```
tablesONLINE 6.0.2 Administrator ------ --- Application Developer's Menu ------
COMMAND ===>

To select, enter number/symbol on command line:

A    EDIT TABLE           - Add/Change/Delete Rows in a Table
B    BROWSE TABLE         - Display Contents of a Table
C    TBDRIVC              - Execute TBLBASE Commands
D    DEFINE TABLE         - Define Table, View and Data Descriptions
U    UTILITIES            - Copy/Rename/Delete a Table
1    EDIT MENU TABLE      - Add/Change/Delete Application menu Items
2    EDIT PFKS TABLE      - Add/Change/Delete Application PF Keys
3    EDIT CMDS TABLE      - Add/Change/Delete Application Alias Commands
4    EDIT HELP TABLE      - Add/Change/Delete Application Help Items
5    EDIT MSGS TABLE      - Add/Change/Delete Application Messages
6    EDIT DESC TABLE      - Add/Change/Delete Application Screen Descriptions
7    EDIT LIBR TABLE      - Add/Change/Delete Application Library Names

Enter HELP at any stage for help within tablesONLINE.
Enter PF for program function key assignments.
Enter X to suspend tablesONLINE and return to CICS.
```

Figure 6-1: Application Developer's Menu

The first five options in the Application Developer's menu (A-U) were discussed in previous chapters.

Options 1 to 7 allow you to edit the driving tables for your application. Selecting one of these options will display a screen similar to Figure 6-2.


```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Menu Table -----
COMMAND ===>

Please indicate the table required by entering the parameters below.
 Data Table Library          => APPLICATIONS.DRIVING.TABLES
 View Library (If different) => SYSTEM.TABLES
 Table Object==> TBOLMENU
                                                    Generation => 0


Enter row key for direct search or positioning on the edit table screen.

SCREEN NAME          ===>
SELECT SYMBOL        ===>
```

Figure 6-2: The Application's Menu Table - Identify Screen

The Data Table Library name and the View Library name are displayed in the screen in white. The Application Driving Tables always is found in the Applications Driving Tables library and the View required to edit those tables is found in the System Tables library. As a result, these are protected fields that may not be changed.

The process of editing any of these Application Driving Tables is the same as editing any other table.

You must specify the name of the Table Object and the generation that you wish to edit. To view a list of the available Application Driving Tables, place the cursor in the Table Object field and press <PF1> for help. To select one of the Table Objects from the list, place the cursor to the left of the row and press <Enter>, or use the S line command.

Once these fields are complete, press <Enter>. This will refresh the screen so that it will display fields where you may insert a table row key. In this example, the keys are Screen Name and Select Symbol.

If you wish to begin by editing a specific row in the table, you can insert the full key or a partial key in these fields. If, however, you wish to start editing at the beginning of the table, leave these fields blank. Press <Enter> to open the table for edit.

This screen is followed by either the **Edit-Table** or the **Edit-Row** screen depending on whether or not a row key was specified. If you inserted a row key in the **Identify** screen, you will go directly to the **Edit-Row** screen. If you did not insert a row key, you will go to the **Edit-Table** screen where you can then move through the rows selecting rows to edit using line commands.

Menu Tables

The first Application Driving table that we will look at is the Menu table. Menu tables control the flow of the application. By editing this table, you can modify existing menu rows, create new menu rows or create a complete new menu screen.

To edit this table, select Option 1 - EDIT MENU TABLE from the Application Developer's menu (see Figure 6-1), leaving the key field on the table identification screen blank. The system displays a screen similar to Figure 6-3.

```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Menu Table -----
COMMAND ===>

                                                          Row Location : 1

      SCREEN  SELECT        SHORT                              LONG
       NAME   SYMBOL     DESCRIPTION                        DESCRIPTION
     _______ _______ _____________________ ____________________________________
     APPLDEVL  A     EDIT TABLE           Add/Change/Delete Rows in a Table
     APPLDEVL  B     BROWSE TABLE         Display Contents of a Table
     APPLDEVL  C     TBDRIVC              Execute TBLBASE Commands
     APPLDEVL  D     DEFINE TABLE         Define Table, View and Data Descripti
     APPLDEVL  U     UTILITIES            Copy/Rename/Delete a Table
     APPLDEVL  1     EDIT MENU TABLE      Add/Change/Delete Application menu It
     APPLDEVL  2     EDIT PFKS TABLE      Add/Change/Delete Application PF Keys
     APPLDEVL  3     EDIT CMDS TABLE      Add/Change/Delete Application Alias C
     APPLDEVL  4     EDIT HELP TABLE      Add/Change/Delete Application Help It
     APPLDEVL  5     EDIT MSGS TABLE      Add/Change/Delete Application Message
     APPLDEVL  6     EDIT DESC TABLE      Add/Change/Delete Application Screen
     APPLDEVL  7     EDIT LIBR TABLE      Add/Change/Delete Application Library
     DEFINE    D     DEFINE ALL           Define All Table Elements (View and D
     DEFINE    D2    DEFINE VIEW SUPPLMT  Define Supplementary Information for
     DEFINE    MSTR  UPDATE MASTER VIEWS
++++ DEFINE    P     PRINT VIEW           Submit a Batch Job to Print a View
```

Figure 6-3: The Application's Menu Table - Edit-Table Screen

The rows in the menu table are sequentially organized by Screen Name and Select Symbol. When the menu is displayed in an application, the rows included in the menu are displayed in sequence by select symbol. For example, the rows whose screen name is APPLDEVL form the selections in the Application Developer's menu we saw in Figure 6-1.

To Add a New Menu Item

To add a new menu item, select an existing item to use as a template and use the N line command. You may now make any required changes to the fields defining the menu item and save the new item.

To Edit an Existing Menu Row

Each line in this table displays one row and can be selected for editing by entering S or U beside it. A line specifies one entry for each menu option in the tablesONLINE menus. Once a row is selected, the **Edit-Row** screen appears (see Figure 6-4). All of the fields are described below, however, on your terminal they will be spread over two screens. Use the scroll keys, <PF7> and <PF8>, to move to these fields.

```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Menu Table -----
COMMAND ===>

                            Key(s) Protected          Row Location : 2


     SCREEN NAME          : APPLDEVL
     SELECT SYMBOL        : B
     SELECT SYMBOL EXTATT : 00000000
     DISPLAY INDICATOR    :

     SHORT DESCRIPTION    : BROWSE TABLE
     SHORT DESC.EXTATT    : 00000000

     LONG DESCRIPTION     : Display Contents of a Table
     LONG DESC.EXTATT     : 00000000

     NEXT MENU SELECTION  :

     DYNAMIC VIEW SUFFIX  : T
     PROGRAM OP MODE      : B
     FUNCTION             : BROWSE
     DB TYPE              : TABLE
     TABLE NAME           :
     TABLE LIBRARY        :
     VIEW NAME            :
     VIEW LIBRARY         :
     DUPLICATE KEYS IND.  :
```

Figure 6-4: The Application's Menu Table - Edit-Row Screen

In this example using the S line command, we have selected the BROWSE TABLE function (Select Symbol B) to be modified.

The screen is divided into sections. The first group of fields (from Screen Name to Long Description) determine menu organization and display descriptions. The second group (beginning with Next Menu Selection) control the flow of your application. The content of the screen will vary depending on what you enter in the Dynamic View Suffix field.

The first two fields on this screen make up the key for the menu table. This table does not allow duplicate keys. This ensures that the select symbol for each row in a table is unique. When creating or editing menu rows, use the fields described in Table 6-2. A brief description of each field is provided.

Table 6-2: Menu Table Field Descriptions

Field Name	Description
Screen Name	This field contains the name of the screen that contains the menu row to be edited. The screen name may be up to eight characters in length.
Select Symbol *	This field contains the symbol that will be used to select the row from the menu. The select symbol may be up to four characters in length.
Display Indicator	This field indicates whether or not the row will be displayed on the menu. This field may be set to Y for yes, display the row, or N for no, do not display the row. The default value is Y.
Short Description *	This field contains a brief description of the menu function.
Long Description *	This field contains a long description of the menu function.
Next Menu Selection	This field indicates what action is to be taken on return to a menu after this particular selection has been made and completed. If this field is blank, the menu is re-displayed. If you wish to perform another function on return to the menu, insert the select symbol for another row in the menu into this field. If the value CICS is inserted in this field, the user is sent back to native CICS on completion of the menu choice. If this is done for a menu entry that has been assigned as a user's starting menu option in the Application Control Table, that user can only execute the process defined for that menu entry before being logged out. This can be used to restrict a user to a single function, such as browsing a particular table.
* These fields all appear together with extended attribute fields which allow you to control color displays for your environment. If you are familiar with the extended attribute capabilities of your environment, you may wish to use these fields to customize your menus. However, if you are not familiar with them, it is recommended that you leave these fields as they are.	

Dynamic View Suffix

From a tablesONLINE menu, control may be passed to one of the following:

- the tablesONLINE Table Editor for editing or browsing tables, or for table definition
- another tablesONLINE menu
- a Utility Function
- another CICS program or a CICS transaction

Each of these transfer types requires a different set of parameters, yet each type of transfer is controlled by entries in a single menu table.

The Dynamic View Suffix is used to specify the type of transfer to be performed. The balance of the fields that will appear in this screen will depend on the value placed in the Dynamic View Suffix field. This field may be set to the values listed in Table 6-3.

Table 6-3: Dynamic View Suffix Values

Field Value	Transfer To
T	Table Editor
M	Menu
S	Utility
(blank)	CICS Program/CICS Transaction ID

Table Editor (T)

To create a row that transfers control to the Table Editor insert T in the Dynamic View Suffix field and press <Enter>. This will refresh the screen using a View for entering table editing parameters (see Figure 6-5).

```
DYNAMIC VIEW SUFFIX  : T
PROGRAM OP MODE      : E
FUNCTION             : EDIT
DB TYPE              : TABLE
TABLE NAME           :
TABLE LIBRARY        :
VIEW NAME            :
VIEW LIBRARY         :
DUPLICATE KEYS IND.  :
```

Figure 6-5: Table Editor Parameters

Program Opening Mode

This field is used to control which tables (Data or View) are opened and whether they are opened for read or write access. Editing and browsing operations, for example, invoke the same program but with different program opening modes. For editing, the table is opened with a program opening mode of E (open for write) and for browsing the table is opened with a program opening mode of B (open for read). The program opening mode values are listed inTable 6-4.

Table 6-4: Program Opening Mode Values

Mode	Usage
B	Browse Data Table
E	Edit Data Table
C	Browse View
D	Edit View - display order
L	Edit View - row layout
U	Utilities

The editor uses different sets of Views depending on whether it is operating on a Data Table (mode B or E), a View (mode C, D or L) or a Utility (mode U).

Function and DB Type

The FUNCTION and DB TYPE fields are used by the Editor as components of keys for lookups in the HELP and the DESCription tables. See the sections below on editing the HELP and DESCription tables for details.

Table Name and Table Library

These fields can be used to restrict a user to a particular table or library. If a table name is provided, then executing the menu entry opens the Editor with that table. In this case, the user cannot select any other table. If a library name is provided, executing the menu entry will open the **Identify Table** screen where the user may select any table that is stored on that library. In this case, the user does not have access to tables in any other library.

If these fields are left blank, the last table accessed is used as a default value in the **Identify Table** screen. It may then be changed to another library and table by the user. These fields may or may not be displayed on the **Identify Table** screen. This will depend on how the Display Table Name and Display Library Indicator fields in the Application Control Table for the user and application have been set.

View Name and View Library

These fields are used to specify the View and/or View library that is used to edit the table specified above. If you wish to use a View other than the one that is defined with the same name as the Data Table, insert the View name in this field. If this field is blank, the View name used for the edit will be the one with the same name as the Data Table. If you wish to use a View in a different library, enter the name of that library.

The ability to specify the use of a particular View of a table for a menu item allows you to provide a user with access only to the required fields in the table. This may be desirable in cases such as personnel tables containing employee records. Using this feature, it is possible to display only the necessary fields to the user.

Duplicate Keys Indicator

The Duplicate Keys Indicator field here interacts with the Duplicate Keys Allowed? field in the View for the table being edited. The following describes how these fields work together. If:

- the table is hashed;
- the View has N for duplicate keys; or
- the menu has N for duplicate keys

then:

- duplicate keys are not allowed and the user cannot change this, for the Duplicate Keys Allowed? switch does not appear on the **Table Identification** screen.

Otherwise:

- the Duplicate Keys Allowed field appears on the table identification screen;
- the Duplicate Keys Indicator menu controls the initial setting of the Duplicate Keys Allowed? switch in Table 6-5 and the user may alter the setting.

Table 6-5: Duplicate Keys Indicator Settings

Indicator Setting	Duplicate Keys Allowed?
(blank)	N
Y	Y

Menu (M)

To create a menu item that transfers control to another menu, insert M in the Dynamic View Suffix field and press <Enter>. This action refreshes the screen using a View for entering menu transfer parameters (see Figure 6-6).

```
DYNAMIC VIEW SUFFIX  : M
TRANSFER MENU        : UTILITY
TRANSFER MENU SYMBOL : A
```

Figure 6-6: Menu Transfer Parameters

Transfer Menu

This field indicates the name of the menu to which control will be passed by this menu item. tablesONLINE searches the menu table using the Transfer menu field as a key and builds the next menu from the items with a screen name matching that key.

Transfer Menu Symbol

This field may contain a Transfer menu Symbol, the selection symbol for a particular entry in the transfer menu. If this field is blank, the menu specified in the Transfer Menu field is displayed and a user may make a selection from that menu. If an entry is made in the Transfer Menu Symbol field, it is treated as a selection from the menu specified in the Transfer menu field and that entry is executed.

Through successive use of such options, it is possible to nest menus indefinitely. To return from a subordinate menu to the calling menu, press <End> or <PF12>.

Utility (S)

A menu item with a Dynamic View Suffix set to S transfers control to a utility function. The system displays the fields in Figure 6-7.

```
DYNAMIC VIEW SUFFIX  : S
PROGRAM OP MODE      : U
FUNCTION             : DEF TBL
```

Figure 6-7: Utility Invocation Parameters

The transfer to a tablesONLINE utility is actually a transfer to the Table Editor with all of the editor transfer parameters set by tablesONLINE - to values appropriate for a particular Utility function.

A program opening mode of U indicates this is a Utility and the function field indicates which one.

Table 6-6 is a list of the supported utility functions.

Table 6-6: Utility Functions

Utility Function	Description
CHG PSWD	Change either the read or write password.
COPY	Copy a table to the same or a different library.
CPY APPL	Copy all the Applications Control Table.
CPY VIEW	Copy a View to the same or a different library.
CREATE	Create or edit an Alternate Index.
DEF TBL	Edit tableBASE table definition (DT Block).
DEL	Delete a generation of a table.
DEL VIEW	Delete a View table.
EDITSUPL	Enter supplementary information for View.
GENCOPYB	Generate copybook from View table.
PRTTABLE	Print a table.
PRTVIEW	Print a View table.
RENAME-D	Change the Data Table name.
RENAME-V	Change the View name.
RESTRUCT	Restructure data in a table.
WPROT VW	Place a write password on a View.
XED PROF	Edit your User Profile.

The best way to set up menu items that call a tablesONLINE Utility is to start with an existing menu table such as TBOLMENU. Create a copy of the desired menu item in TBOLMENU (follow the procedure outlined in "To Add a New Menu Item" on page 125) and simply change the Menu Name of the copied item to the name of your menu table. You can also modify any unprotected fields such as Select Symbol and Short Description. A copy of the menu item that calls the desired tablesONLINE utility has now been created in your Menu table. This procedure ensures that the Utility is called correctly.

Blank Program/Transaction ID

To create a menu item that transfers control to a program or starting a CICS transaction, leave the Dynamic View Suffix blank and press <Enter>. This refreshes the screen using a View for entering CICS transfer Program Control parameters (see Figure 6-8).

```
DYNAMIC VIEW SUFFIX  :
TRANSFER PROGRAM     : TBDRIVC
RETAIN ENVIRONMENT?  :
TRANSFER TRANSACTION :
TRANSFER PARAMETER   :
TRANSFER PARAMETER   :
```

Figure 6-8: Transfer Program Control Parameters

Transfer Program

Enter the name of the program that is to be invoked by the menu item, in the Transfer Program field. Specifying a program name in the Transfer Program field causes that program to be invoked by a CICS XCTL (transfer control) command. If this field is blank, the transaction that is specified in the Transfer Transaction field, described below, is started.

Retain Environment

This field only applies to transfers to programs, rather than transfers to transactions. If it is set to Y, the called program receives a 4096 byte DFHCOMM area. Otherwise, the called program receives only the first 20 bytes that contain session identification information:

```
01 DFHCOMMAREA.
             05 SESSION-IDENTIFICATION          PIC X(8).
             05 PROGRAM-IN-PROCESS              PIC X(8).
             05 SUBPROGRAM-IN-PROCESS           PIC X(1).
             05 FILLER                          PIC X(3).
```

To return to tablesONLINE at exactly the point of transfer, the called subroutine must issue a CICS XCTL call to TBDKAPPL with a DFHCOMMAREA of exactly 20 bytes with the following format:

```
01 DFHCOMMAREA.
             05 SESSION-IDENTIFICATION          PIC X(8).
             05 MESSAGE-KEY                     PIC X(7).
             05 NEW-SESSION-IND                 PIC X(1).
             05 FILLER                          PIC X(3).
             05 PROGRAM-RETURN-CODE             PIC S9(4) COMP.
```


The Session-Identification field must be returned exactly as it was passed. Provisions are also made to return a condition code and a 7 byte message key. If the message key is non-blank, tablesONLINE will attempt to retrieve a message from the message table. When the message key is non-blank, the PROGRAM-RETURN-CODE must be zero, otherwise control will be returned to the transfer program.

Normally the NEW-SESSION-IND is blank when returning to the point of transfer, however a Y in this indicator causes tablesONLINE to set up a new window. A **+** or - in this indicator, along with a PROGRAM-RETURN-CODE of -1, causes tablesONLINE to attempt to move to another window created either before or after the one you are currently in.

Transfer Transaction

Enter the name of the transaction that is to be invoked by the menu item. If this field is completed, the Transfer Program field must be blank.

Transfer Parameters

A non-blank transfer parameter causes tablesONLINE to mimic terminal input when a transfer transaction is started. This allows a form of parameter passing to CICS applications that expect to find data on the screen. For example, to start a tablesONLINE application where the following is entered on the terminal,

```
TBOLPAYRUSER1,
```

TBOL is the transfer transaction and PAYRUSER1 represents the application ID and user ID and is entered as the transfer parameter. Similarly, a CEMT transaction could be started with 'INQUIRE PROGRAM(TB*)' as the TRANSFER PARAMETER.

Transferring Directly to Another Application

Normally, tablesONLINE is terminated when the command CICS or its alias X is executed, or the <Clear> key is pressed. There may be situations when you wish to terminate tablesONLINE and transfer to another menu system or to pass control to another tablesONLINE session. To do so, blank out the transfer menu name and enter the literal TRSF in the Transfer menu Symbol field. This will cause tablesONLINE to activate the terminating transaction or program specified as Transfer Program or Transaction when the Dynamic View Suffix is blank.

Using TBDRIVC as the terminating program may be useful for diagnosing user exit programming, as it uses the standard 20-character DFHCOMMAREA protocol and, when finished, returns control to the point at which it was initiated. See the tableBASE Programming Guide for a full description of TBDRIVC.

Create a New Menu

To create a new menu, create a new row in the menu table for each option that is required on the new menu using the procedures described above. The Screen Name field for each of these new rows should be set to the name of your new menu.

Once the menu has been created, you need to provide a route so that the menu can be accessed by users. There are two ways to do this:

- Add a new menu option to one of your existing menus, with the Dynamic View Suffix field set to M and the Transfer menu field set to the name of your new menu.
- Make changes to the Application Control Table so that the new menu becomes the initial menu for a user or a group of users. This function is usually restricted so that only your tableBASE administrator can perform it. For further details, refer to the tableBASE Administration Guide.

PF Key Tables

It is possible to customize the assignment of PF keys for any tablesONLINE application. This is done by editing the PF Key table. To edit this table, select Option 2 - EDIT PFKS TABLE from the Application Developer's menu and press <Enter>. This action displays a screen (an **Identify Table** screen) where you may insert a specific PFK identifier. Pressing <Enter> will bring up the **Edit-Row** screen. If this field is left blank, **The Application's PF Keys** screen is displayed (see Figure 6-9).

```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's PF Keys --------
COMMAND ===>

                                                          Row Location : 1

        PFK              PFK
    IDENTIFIER       COMMAND
    __________ ____________________
       PF 1    HELP
       PF 2    EXECUTE
       PF 3    END
       PF 4    GETCNT
       PF 5    FINDFLD
       PF 6    GETKEY
       PF 7    UP
       PF 8    DOWN
       PF 9    WINDOW
       PF10    LEFT
       PF11    RIGHT
       PF12    CANCEL
       PF13    HELP
       PF14    EXECUTE
       PF15    END
++++   PF16    GETCNT
```

Figure 6-9: The Application's PF Keys Screen

If you have not already specified a row key, you may select a row from the PF Key table using the U line command. Once you are in the **Edit-Row** screen you can edit the PFK command information.

This table translates PF keys into commands. PF keys can be set up for any command that can be given from the command line. The keys are defined for the entire application, although depending on the screen you are using and the commands available from that screen, they may not all be active at any given time.

There are 24 PF keys set up for tablesONLINE as distributed defaults. Thirteen of these keys, <PF1>-<PF12> and <PF21>, have been assigned. The commands for PF keys 13 to 24 are the same as those for 1 to 12, except for <PF21> which is set up for the freeze keys command.

You can define PF keys to the maximum number your system supports and reassign the keys the delivered defaults. If you alter the pre-assigned PF keys for tablesONLINE, they will no longer match the product documentation.

Command Tables

tablesONLINE allows you to set up command aliases for command and parameter combinations. A command alias is simply an abbreviated form of a command that can be entered on the command line. These command aliases are stored in the Command Table. To edit this table, select Option 3 - EDIT CMDS TABLE from the Application Developer's menu and press <Enter>. This displays the **Identify Table** screen where you can insert a command alias and command parameter; this will bring up an **Edit-Row** screen. If this field is left blank, **The Application's Alias Commands** screen is displayed (see Figure 6-10).


```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Alias Commands -
COMMAND ===>

                                                          Row Location : 1

           COMMAND                    COMMAND                       ACTUAL
            ALIAS                       PARM                       COMMAND
     ____________________ ____________ALIAS_____________ ____________________ _
     BROWSE                                              B
     CA                                                  CANCEL
     CAN                                                 CANCEL
     CANC                                                CANCEL
     CANCE                                               CANCEL
     CD                                                  CONFIRMDEL          Y
     CD                   N                              CONFIRMDEL          N
     CD                   NO                             CONFIRMDEL          N
     CD                   Y                              CONFIRMDEL          Y
     CD                   YE                             CONFIRMDEL          Y
     CD                   YES                            CONFIRMDEL          Y
     CS                                                  CONFIRMSAV          Y
     CS                   N                              CONFIRMSAV          N
     CS                   NO                             CONFIRMSAV          N
     CS                   Y                              CONFIRMSAV          Y
++++ CS                   YE                             CONFIRMSAV          Y
```

Figure 6-10: The Application's Command Aliases Screen

If you have not already specified a row key, you may select a row from the Command table using the U line command. Once you are in **The Application's Command Aliases** screen you can edit the command alias information.

This table allows you to assign aliases and parameters that are translated by this table into tablesONLINE commands. An alias together with its parameters is entered on the command line. tablesONLINE then goes to the Command table to find the actual command and parameters that are represented by this alias. This actual command and parameters are then executed by tablesONLINE.

If a parameter is specified on the command line and there are no parameters listed with the actual command, then the parameter specified on the command line will be used. If the actual command has a parameter listed with it, then any parameter entered on the command line will be replaced with the actual command parameter.

For example, if the user types CD NO on the command line, matching the eighth line in Figure 6-10, that entry is translated to CONFIRMDEL N and passed to tablesONLINE.

When a command alias is inserted on the command line, tablesONLINE searches the Command table. This search is attempted twice. The first search uses a key made up of the command and parameter combination as inserted on the command line. If this fails,

a second search is attempted. This second search is attempted with the same command but with a blank parameter. For example, if the user types DO 1000, the second search will match the last line in Figure 6-10, DO blank DOWN blank entry. The command is translated to DOWN 1000, the 1000 interpreted as the actual command parameter. This combination is then executed by tablesONLINE.

Disabling Commands

It is also possible to use the Command table to disable an existing tablesONLINE command. This is useful if you wish to prevent users from issuing certain sensitive commands such as `CICS`.

Note: The Clear key is equivalent to entering either `CICS` or `X` on the command line.

To disable the CICS command (<Clear> key), add a row with the command alias set to `CICS`, the actual Command field set to `END` and the other fields cleared. Now if users try to use leave tablesONLINE by using the <Clear> key or the CICS line command, they will simply be returned to the previous screen. Remember to remove the row with the command alias `X` to prevent users from accomplishing the same function using this command alias.

Help Tables

It is possible to customize help for any tablesONLINE application, by editing the Help table. To edit this table, select Option 4 - EDIT HELP TABLE from the Application Developer's menu. This refreshes the screen displaying the key fields where you may insert a View name, field name and sequence number. If you wish to edit the information for a particular help entry, you can insert the appropriate row key and that row will be opened in the **Edit-Row** screen. If these fields are left blank, the **Edit-Table** screen is displayed. From this screen you may select a help item to edit using one of the line commands. This action displays the screen shown in Figure 6-11.

```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Help/Tutorial -
COMMAND ===>

                        Update Mode                 Row Location : 33

   VIEW NAME            : EXAMPLE4
   FIELD NAME           : CHARITABLE DONATION
   SEQ                  : 1

   INDIRECT LOOKUP IND  :
   DISPL ATTR           :
   DISPL EXTATT         : 00000000

                          ....|....1....|....2....|....3....|....4....|....5
   HELP DATA            : The Charitable Donation field requires 2 decimal
                          ....|....6....|....7....|....
                        : places.
```

Figure 6-11: The Application's Help/Tutorial - Edit-Row Screen

Creating Help for Editing Tables

The View Name in this screen indicates the table to which the help information applies. The Field Name specifies the field within that table. The sequence of items in the Help table is controlled by a key composed of these two fields together with the sequence number. The Sequence Number allows you to create multiple lines of help text for any table field.

If the help entry you are defining is to apply to the entire table rather than a specific field within that table, you should leave the Field Name blank. If the help entry you are defining is to apply to any table containing the specified field name, you should leave the View Name blank. These help entries may be viewed by a user by typing TABLE on the command line and pressing <PF1> for help.

The Indirect Lookup Indicator field allows you to create help message lines that are actually retrieved from another Help table or another Field Name in the current Help table or both.

To create an indirect help entry, type an `I` in the Indirect Lookup Indicator field, clear the remaining non-key fields and press <Enter>. This displays the screen shown in Figure 6-12. The screen contains fields that are used to define an indirect help entry, like Indirect Table, Indirect Key-Table, and Indirect Key-Field.

```
tablesONLINE 5.1.0 Administrator ------ Edit the Application's Help/Tutorial --
COMMAND ===>

                            Key(s) Protected           Row Location : 18


     VIEW NAME             :
     FIELD NAME            : DATA DISPL EXTATT
     SEQ                   : 2

     INDIRECT LOOKUP IND   : I
     DISPL ATTR            : 0
     DISPL EXTATT          : 00000000

     INDIRECT TABLE        : TBOLHELP
     INDIRECT KEY - TABLE  :
     INDIRECT KEY - FIELD  : EXTENDED ATTRIBUTES
```

Figure 6-12: Retrieving Help from Another Help Table

In the Indirect Table field you must specify the name of the Help table from which the indirect help is to be retrieved. The Indirect Key-Table field allows you to specify the table name, the first key field for the help entry to be retrieved. The Indirect Key-Field allows you to specify the field name, the second key field for the help entry to be retrieved. Indirect help lookup entries can be inserted anywhere in help, but are acted on only once. It is not possible to nest or to create a recursive loop.

The remaining three fields - Displ Attr, Displ Extatt, and Help Data - are those required to display a help message. The display attributes and display extended attributes control the visual presentation of the data on your screen. How these features are used will depend on your installation. The final field will contain the help message itself. This field may contain up to 79 characters of help data. Should you wish to create a help message that is longer than 79 characters you may create multiple entries using the Sequence Number field. Help entries that have identical View name and Field Name will be displayed together in the order specified by the sequence numbers.

Creating Help for menus

Help is available for both menus and tables. When a menu is displayed and help is requested, the help table designated in the Application Control Table is searched. Table 6-7 shows the two searches that are performed in sequence.

Table 6-7: Menu Help Lookup Order

Key 1	Key 2
<menu-ID>	<Short Description>
	<Short Description>

When the cursor lies on a menu item and <PF1> is pressed, a search key is created using the menu-ID (in place of the View Name) and the Short Description (in place of the Field Name). If the search is not successful, the menu-ID is made blank and the search is attempted again.

A portion of the Short Description can be entered on the command line, if it follows the keyword MENU, as shown in this example:

MENU,<short description>

If general help is required for the entire menu, only the keyword MENU should be entered on the command line.

Lookup Order for Help Tables

When you are using the Table Editor (whether editing normal Data Tables, editing Views to define tables, or performing a utility function), an additional help table may be specified in the View controlling the edit function. This table is searched by View Name and Field Name. This allows help material to be associated with the data, or even with a particular View of the data.

Help tables of this type are optional. If the View's Help table field is blank, there is no specific help for this View. When this field is non-blank, the data-specific Help table is searched before the application's Help table.

The lookup order performed for contextual help for data fields are listed in Table 6-8.

Table 6-8: Contextual Help Lookup Order

Help Table Name	Search Done	Key 1	Key 2
From View	If present	<view name >	<field name>
			<field name>
From ACT	If above search fails	<view name>	<field name>
			<field name>

The results of the first successful search are given as the response to a user's help request. If all searches fail, a message is returned.

The first two searches are tried only if the Editor is in use and there is a Help table name in the relevant View. Data-specific help is given precedence over application-specific help.

The second pair of searches is done when the first pair do not apply, or fail. Application help should always be available. Developers may choose to build their own Help tables or to use the tablesONLINE Help table TBOLHELP with or without customization, but a Help table name must be specified in the Application Control Table entries for their applications, and, the named table must exist.

Within each pair, a specific search is tried first, followed by a more general search.This avoids much potential redundancy in help tables. For example, if a Balance field appears in many tables, then one help item keyed by field name alone and found by the second search described above, can provide help for that field in any table. This might be something as simple as a reminder that Balance is a signed number and needs two digits after the decimal. If different help information is needed for the Balance field in certain tables, then more specific entries that will be found by the first search can be constructed.

Messages Tables

It is possible to customize the messages available for any tablesONLINE application, by editing the Messages table. To edit this table, select Option 5 - EDIT MSGS TABLE from the Application Developer's menu.

```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Messages table -
COMMAND ===>


                         Update Mode                Row Location : 146

     MSG GROUP           : TB-
     MSG NUMBER          : 5934
     MSG SEQ             : 0

     MSG TYPE            : E

     DISPLAY ATTR.       : 0
     DISPLAY EXTATT      : 00000000

                           ....|....1....|....2....|....3....|....4....|....5
     MESSAGE TEXT        : DATA Table 'XXXXXXXX' cannot be stored, library 'X
                           ....|....6....|....7
                         : XXXXXXX' is full.

     1ST VARIABLE OFFSET : 13
     2ND VARIABLE OFFSET : 50
```

Figure 6-13: The Application's Message Table - Edit-Row Screen

A tablesONLINE user exit program may return a message code such as TB-5934 to indicate a problem. This message code is made up of a message group, in this case TB- and a message number 5934. These fields are the key fields for the messages table. When a message code is returned, tablesONLINE searches the message table for a row with the specified key and displays the appropriate message.

tablesONLINE is delivered with two message groups: TB- and TBS. These are tablesONLINE and tableBASE message groups respectively and should not be deleted by the user. When creating customized messages for an application, simply add to the existing Message table, to create a unique message group for your application and to create new messages within that group.

As with a Help table, it is possible to create a multi-line message, using the message sequence number. Messages are retrieved using the message group and message number as the key, and are displayed in the order indicated by the sequence number.

You can create four types of messages (see Table 6-9)

Table 6-9: Message Types

Type	Description
I	Information
W	Warning
E	Error
A	Abend

A message type must be specified for each entry in the message table. Each type of message is handled differently by tablesONLINE.

After delivering an information or warning message, tablesONLINE attempts to continue processing as if there had been no message. These message types differ in that information messages may be suppressed for a user using the SUPPRESSINFO Y command. This option may be modified by the User by updating the Profile (see "Edit Profile" on page 115).

When an error message occurs, tablesONLINE attempts to restore the system to the state that existed before the program generating the message was called, so that the user can perform a different action. If the message type is abend, the task abends and no further actions may be carried out.

Display Attribute and Display Extended Attributes are used to assign the screen attributes to be used when displaying the message text. How and whether these features can be used will depend on your own installation.

It is possible to create messages that can accommodate additional strings when invoked by a user exit program. The Variable Offset fields tell the message-handler where to put these additional strings. In the example, a first string overlays the Xs starting at the character position specified by the first variable offset, while a second string will overlay the Xs starting at the character position specified by the second variable offset. A value of zero for either of these offsets will suppress the substitution of variable data into the message.

Description Tables

Description tables provide part of the text that appears on the various screens in tablesONLINE. There is only one Description table per application Id. By editing a Description table you can alter some of the text that appears on the screens. Not all of the screen content is provided by the Description table. For example, in the case of a menu screen, text also comes from the Menu table.

In Figure 6-14, the highlighted text indicates descriptions that can be edited using a Description table. To edit a Description table, select Option 6 - EDIT DESC TABLE from the Application Developer's menu.

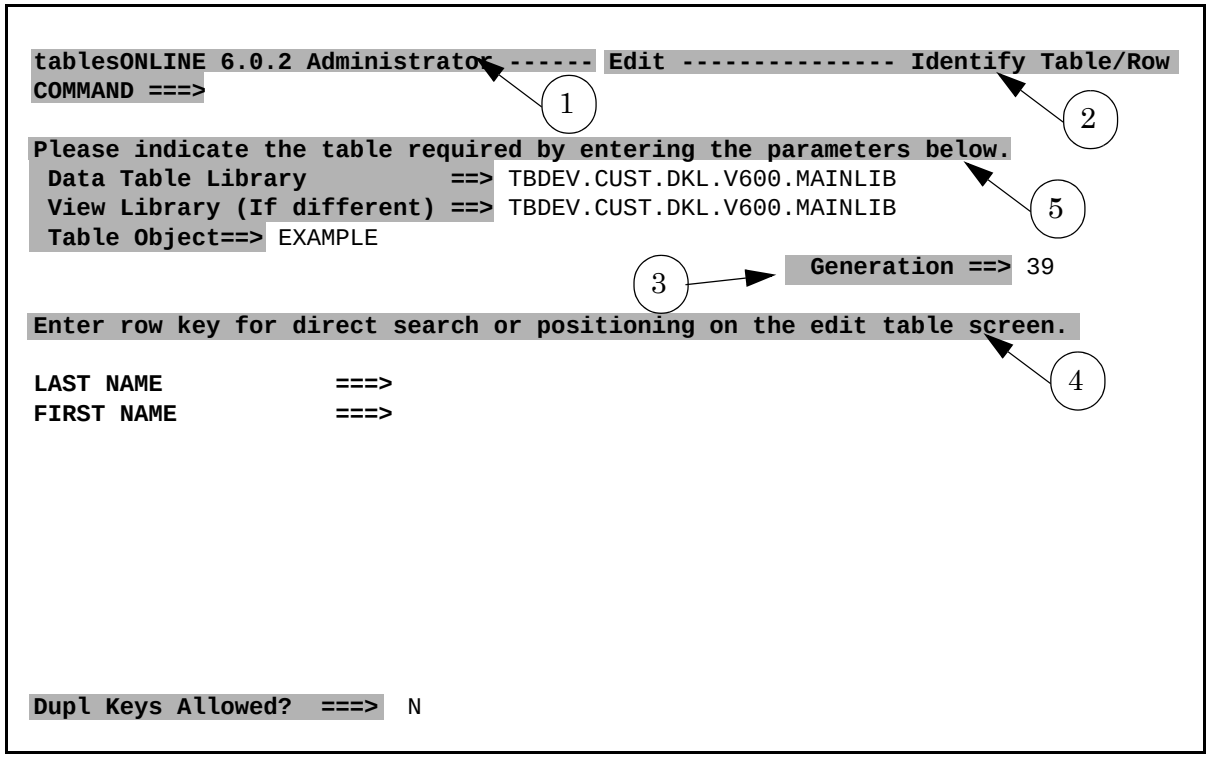

Figure 6-14: Modifiable Parts of a Typical Screen

There are nine fields used to define each row in the Description table (see Figure 6-15).

```
tablesONLINE 6.0.2 Administrator ------ Edit the Application's Screen Constants
COMMAND ===>

                           Key(s) Protected          Row Location : 144


     PROGRAM /DBTYPE      : PFKS
     SCREEN-ID /VIEWNAME  : EDIT-ROW

     FUNCTION             : BROWSE
     FIELD NAME           : FUNC-ID

     DESC DISPL LENGTH    : 39
     DESC DISPL OFFSET    : 1
     DESC DISPL ATTR      : 8
     DESC DISPL EXTATT    : 00000000

                            ....|....1....|....2....|....3....|....4....|....5
     DESCRIPTION DATA     : Browse Rows ------------ PFKS/COMMANDS
                            ....|....6....|....7....|....8....|....9....|...10
                          :
```

Figure 6-15: The Application's Descriptions Table - Edit-Row Screen

The first four fields form the key for this table and provide the information that indicates the type of screen to which the description text applies. The other fields provide the information about the display and contents of the description text.

The first key field can be one of two classes: Program or DBtype. This key determines how the remaining fields will be handled (the variations are detailed later in this section).

The Program class has seven possible values and the DBtype class has one value (see Table 6-10)

Table 6-10: Program Class and DBtype Class Values

Class Values		Description
Program Class Values	TBONLINE	Description text for all the tablesONLINE screens. For more detail see “TBONLINE” on page 148.
	HELP	Description text for **Help** and **Tutorial** screens. For more detail see “HELP” on page 151.
	MENU	Description text for **Menu** screens. For more detail see “MENU” on page 154.
	MSGR	Description text for **Message** screens. For more detail see “MSGR (Message Screen)” on page 156.
	PFKS	Description text for **PF Key** functions. For more detail see “PFKS (PF Key Screens)” on page 158.
	ERRORCHK	Description text for data conversion error messages. For more detail see “ERRORCHK” on page 160.
	NO-DESCR	Description text for the message that is displayed after a failed description table lookup. For more detail see “NO-DESCR (No Description)” on page 160.
DBtype Class Values	TABLE	Description text for **Table Identification, Edit-Table** and **Edit-Row** screens. For more detail see “Program/DBtype for DBtype Class Value” on page 160.

Description Table Lookup Order

When tablesONLINE presents a screen, it calls the Description table to provide some of the text to be displayed. In order to identify the appropriate text for the screen, tablesONLINE searches the rows of the Description table. The lookup is conducted in a hierarchical order with the most specific and applicable case being examined first and then defaulting down the line to the most general definition that applies. The most specific case found that applies to the screen being presented will be used. Table 6-11 illustrates the lookup order from the most specific to the most general case.

Table 6-11: Screen Descriptions Lookup Order

Lookup Order	Search Keys			
1	<Program/DBtype>	<Screen>	<Function>	<Field Name>
2	<Program/DBtype>	<Screen>		<Field Name>
3	<Program/DBtype>			<Field Name>

Program/DBtype for Program Class Values

This section details the settings of the Description Key hierarchy for each of the Program Class Values listed in Table 6-10.

TBONLINE

Rows in the Description table that have TBONLINE as the Program/DBtype are used to display text on the tablesONLINE table editing/browsing screens.

The fields of the Description table row should be completed as described in Table 6-12.

Table 6-12: Description Table Field Values for TBONLINE

Program	**Description**
Program/DBtype	The Program name TBONLINE is entered in this field.
Screen-ID/ View Name	This field may have the following values: APPL-ID, EDIT-TBL, EDIT-ROW or IDENTIFY. If the Field Name contains the literal `SIGN OFF TABLEBASE`, the Screen-ID field must be blank.
Function	The Function field may contain the name of any function that is currently defined in the XXXXMENU table. If the Field Name contains the literal `SIGN OFF TABLEBASE`, the Function field must be blank.
Field Name	This field may have the following values: APPL-ID, FUNC-ID, SIGN OFF TABLEBASE or DESCn-xxx. See Table 6-13 for a description of each of these values.
Description Display Length	Indicates the total length of the description text.
Description Display Offset	Indicates the location where the first character of the text is to be positioned on the line. The offset is effective only with the following field name values: DESC0-BRM, DESC0-KPM, DESC0-NWM, DESC0-UPM, DESC0-ROW and DESC0-TBL.
Description Display Attributes	Controls the visual presentation of the text on the screen.
Description Display EXTATT	Controls the visual presentation of the text on the screen.
Description Data	The description text that will appear on the screen.

The Field Name value determines which text on the screen is affected. Table 6-13 lists the available Field Name values when the Program/DBtype is TBOLINE.

Table 6-13: Field Name Values for Program/DBtype TBONLINE

Field Name Value	Description
APPL-ID	Literal that appears on the left half of the top line of every tablesONLINE screen (see Figure 6-14 #1). To describe an application, you must specify the literal `APPL-ID` in the Screen-ID and in the Field Name fields. The Function field must contain the four-character application name that is currently defined to tablesONLINE.
FUNC-ID	Literal that appears on the right half of the top line of every tablesONLINE screen (see Figure 6-14 #2). The Screen-ID field must contain one of the following literals `EDIT-TBL`, `EDIT-ROW` or `IDENTIFY`. The Function field may contain the name of any function that is currently defined in the XXXXMENU table.
DESC0	Literal for Table Entry Instruction that appears on the Table Identification screen (see Figure 6-14 #5).
DESC0-BRM	Literal to identify Browse mode that appears on the Edit-Row screen. This literal only appears when another user is editing the row that you selected.
DESC0-KPM	Literal to identify Key Protect mode that appears on the Edit-Row screen when the row has been selected using the S line command.
DESC0-NWM	Literal to identify New Row mode that appears on the Edit-Row screen when the row has been selected using the N or I line commands.
DESC0-UPM	Literal to identify Update mode that appears on the Edit-Row screen when the row has been selected using the U line command.
DESC0-ROW	Literal to identify the Row Location when editing or browsing a table. The literal appears in both the Edit-Table screen and the Edit-Row screens. Literal to identify the Field Position when editing or browsing a View. The literal appears in both the Edit-Table screen and the Edit-Row screens.

Table 6-13: Field Name Values for Program/DBtype TBONLINE (Continued)

Field Name Value	Description
DESC0-TBL	Literal to identify the Table Name when editing or browsing a table. The literal appears in both the Edit-Table screen and the Edit-Row screens. Literal to identify the View Name when editing or browsing a View. The literal appears in both the Edit-Table screen and the Edit-Row screens. **Note:** This literal does not appear when a customized FUNC-ID literal is used for a specific table.
DESC1-LIB	Literal to identify the Data Table Library that appears on the Table Identification Screen.
DESC2-LIB	Literal to identify the View Library that appears on the Table Identification Screen.
DESC3-OBJ	Literal to identify the Table Object name that appears on the Table Identification Screen when the M2M option is turned on. This is the default value (see "Specifying a Data Table and a View" on page 19).
DESC3-TBL	Literal to identify the View name that appears on the Table Identification Screen when the M2M option is turned off.
DESC4-TBL	Literal to identify the Data Table that appears on the Table Identification Screen when the M2M option is turned off.
DESC5-PSW	Literal to identify the Password that appears on the Table Identification Screen.
DESC6-PSC	Literal to identify that the password entered in the field above (DESC5-PSW) must be the Write Password.
DESC7-GEN	Literal to identify the Generation number that appears on the Table Identification Screen (see Figure 6-14 #3).
DESC8	Literal to identify the Row/Key Entry Instruction that appears on the Table Identification Screen (see Figure 6-14 #4).

Table 6-13: Field Name Values for Program/DBtype TBONLINE (Continued)

Field Name Value	Description
DESC9-DUP	Literal to identify the Duplicate Keys Allowed? switch that appears on the Table Identification Screen. Whether this literal appears is determined by the setting of Duplicate Indicator (see Table 6-5).
SIGN OFF TABLEBASE	The sign off message that displays when exiting from tablesONLINE to native CICS. Both the Screen-ID and the Function fields for this row must be blank and the Field Name contains the literal `SIGN OFF TABLEBASE`.

You may wish to browse through the tablesONLINE Description table (TBOLDESC) to see how these fields are used.

HELP

Rows in the Description table that include HELP as the Program/DBtype are used to display text on the **Help** and **Tutorial** screens. The balance of the text that makes up a **Help** screen comes from the specific Help table associated with the View used to edit the Data Table, or if a table-specific Help table is not available, the XXXXHELP table. The balance of the text that makes up a **Tutorial** screen comes from the XXXXTUTR table.

Whether the description text appears on a Help or Tutorial screen is determined by the Field name value DESC0-INFO or DESC0-TUTR. These Field Name values can only be used to display text that will appear on all **Help** or **Tutorial** screens. Consequently, when using these Field Name values, the Screen-ID and Function fields must be left blank.

The remaining Field Name value is FUNC-ID. When this Field Name value is used the Screen-ID must be completed, and it must contain the one of the following literals: `IDENTIFY`, `EDIT-TABLE`, `EDIT-ROW` or a menu name that currently exists on XXXXMENU table. The Function field may contain the name of a function that currently exists in the XXXXMENU table. If the Screen-ID points to a menu name, then the Function field simply contains the literal `MENU`.

The fields of Description table row should be filled in as listed in Table 6-14.

Table 6-14: Description Table Field Values for HELP

Field Name	Description
Program/DBtype	This field indicates that the description text appears on a Help screen. Enter `HELP` in this field.
Screen-ID/ View Name	When the Field Name is DESC0-INFO or DESC0-TUTR, this field must be blank. When the field name is FUNC-ID, this field must contain one of the following literals: `IDENTIFY`, `EDIT-TABLE`, `EDIT-ROW` or a menu name that currently exists in the XXXXMENU table.
Function	When the Field Name value is DESC0-INFO or DESC0-TUTR this field must be blank. When the Field Name value is FUNC-ID this field may contain the name of any function that is currently defined in tablesONLINE, or, if the Screen-ID points to a menu name, then the Function field simply contains the literal `MENU`.
Field Name	This field may have the following values: FUNC-ID, DESC0-INFO or DESC0-TUTR. See below for more information on each of these options.
Description Display Length	Indicates the total length of the description text.
Description Display Offset	Indicates the location where the first character of the text is to positioned on the line. The offset is effective only with DESC0-INFO and DESC0-TUTR as the Field Name values.
Description Display Attributes	Controls the visual presentation of the text on the screen.
Description Display EXTATT	Controls the visual presentation of the text on the screen.
Description Data	The description text that will appear on the screen.

The Field Name value determines which text on the screen is affected. Table 6-15 lists the available values.

Table 6-15: Field Name Values for Program/DBtype HELP

Field Name Value	Description
FUNC-ID	Literal that appears on the right half of the top line of every Help screen (see Figure 6-16 #1). The Screen-ID field must contain one of the following literals `EDIT-TBL`, `EDIT-ROW` or `IDENTIFY`. The Function field may contain the name of any function that is currently defined in the XXXXMENU table.
DESC0-TUTR	Literal that appears on the fourth line from the top of the Tutorial screen (see Figure 6-16 #2)
DESC0-INFO	Literal that appears on the fourth line from the top of the Help screen (see Figure 6-16 #2)

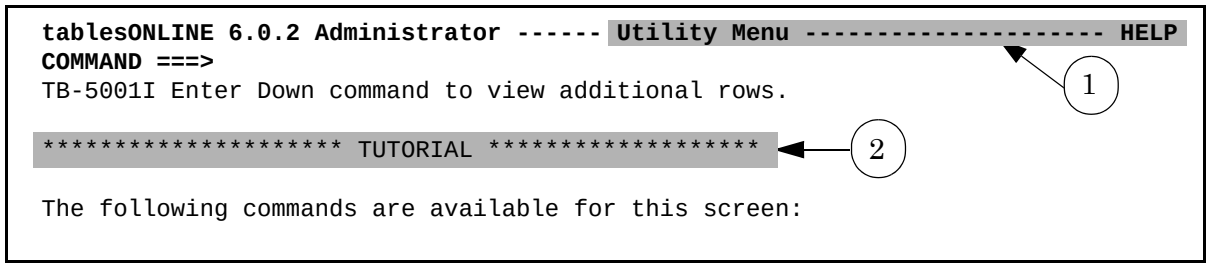

Figure 6-16: Sample Help Screen

MENU

Rows in the Description table that include MENU as the Program/DBtype are used to display text on the **Menu** screens. The remaining text in the **Menu** screen, including the options that make up the menu, comes from the Menu table defined for the application.

The fields of the Description table row should be filled in according to the information listed in Table 6-16.

Table 6-16: Description Table Field Values for MENU

Field Name	Description
Program/DBtype	This field indicates that the description text appears on a **Menu** screen. Enter `MENU` in this field.
Screen-ID/ View Name	Except when the field name value is FUNC-ID, you can create an entry that is displayed on all **Menu** screens by leaving the Screen-ID field blank. To define menu text for a particular menu, you must complete the Screen-ID field. The Screen-ID value must be a Menu name that currently exists in the XXXXMENU table.
Function	The Function field should be left blank.
Field Name	This field may contain the following values: FUNC-ID, DESC0, DESC1, DESC2, DESC3, DESC4, or DESC5. See below for more information on each of these options.
Description Display Length	Indicates the total length of the description text.
Description Display Offset	This field is ignored when the Program/DBtype is MENU.
Description Display Attributes	Controls the visual presentation of the text on the screen.
Description Display EXTATT	Controls the visual presentation of the text on the screen.
Description Data	The description text that will appear on the screen.

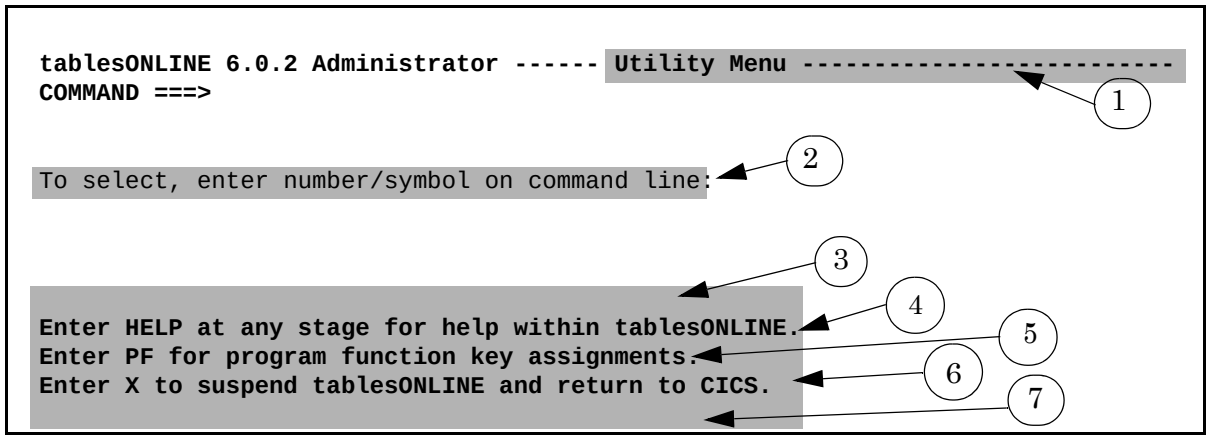

Figure 6-17: Sample Menu Screen

The Field Name value determines which text on the screen is affected. Table 6-17 lists the available field values are listed.

Table 6-17: Field Name Values for Program/DBtype MENU

Field Name Value	Description
FUNC-ID	Literal that appears on the right half of the top line of every Help screen (see Figure 6-17 #1). The Screen-ID field must contain one of the following literals `EDIT-TBL`, `EDIT-ROW` or `IDENTIFY`. The Function field may contain the name of any function that is currently defined in the XXXXMENU table.
DESC0	Literal that appears on the fourth line from the top of the Menu screen (see Figure 6-17 #2).
DESC1	Literal that appears on the fifth last line from the bottom of the Menu screen (see Figure 6-17 #3).
DESC2	Literal that appears on the fourth last line from the bottom of the Menu screen (see Figure 6-17 #4).
DESC3	Literal that appears on the third last line from the bottom of the Menu screen (see Figure 6-17 #5).
DESC4	Literal that appears on the second last line from the bottom of the Menu screen (see Figure 6-17 #6).
DESC5	Literal that appears on the last line from the bottom of the Menu screen (see Figure 6-17 #7).

MSGR (Message Screen)

Rows in the Description table with MSGR as the Program/Dbtype are used to display text on the **Message** screens. The balance of the text that makes up a **Message** screen comes from the Messages table defined for the application.

The fields of the Message Description table row should be filled in according to the information listed in Table 6-18.

Table 6-18: Description Table Field Values for MSGR

Field Name	Description
Program/DBtype	This field indicates that the description text appears on a MSGR screen. Enter `MSGR` in this field.
Screen-ID/ View Name	If you wish to create an entry that is displayed on all message screens, you must leave the Screen-ID field blank. To define description text for a particular message screen, you must specify the Screen-ID. The Screen-ID field may contain one of the following literals: `IDENTIFY`, `EDIT-TABLE`, `EDIT-ROW` or a Menu name that currently exists in the XXXXMENU table.
Function	If you wish to create an entry that is displayed on all message screens, you must leave the Function field blank. The Function field may contain the name of any function that currently exists in the XXXXMENU table, or if the Screen-ID points to a menu name, then the Function field simply contains the literal `MENU`.
Field Name	This field may have the following values: FUNC-ID or DESC0. See below for more information on each of these options.
Description Display Length	Indicates the total length of the description text.
Description Display Offset	This field is ignored when the Program/DBtype is MSGR.
Description Display Attributes	Controls the visual presentation of the text on the screen.
Description Display EXTATT	Controls the visual presentation of the text on the screen.
Description Data	The description text that will appear on the screen.

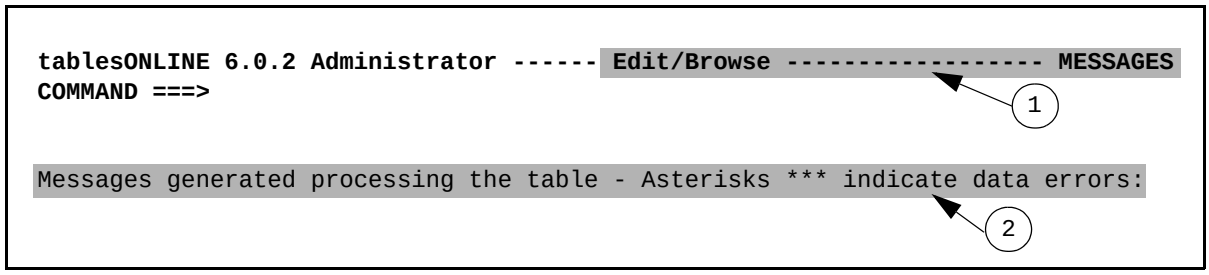

Figure 6-18: Sample Message Screen

The Field Name value determines which text on the screen is affected (see Table 6-19)

Table 6-19: Field Name Values for Program/DBtype MSGR

Field Name Value	Description
FUNC-ID	Literal that appears on the right half of the top line of every Message screen (see Figure 6-18 #1). The Screen-ID field must contain one of the following literals `EDIT-TBL`, `EDIT-ROW` or `IDENTIFY`. The Function field may contain the name of any function that is currently defined in the XXXXMENU table.
DESC0	Literal that appears on the fourth line from the top of the Message screen (see Figure 6-18 #2).

PFKS (PF Key Screens)

Rows in the Description table with PFKS as the Program/DBtype are used to display text on **PF Key** screens. The remaining text in the PFKS screen, including the options that make up the list of available PF KEYs, is controlled by the XXXXPFKS table.

The fields of the PF Key Description table row should be filled in according to the information listed inTable 6-20.

Table 6-20: Description Table Field Values for PFKS

Field Name	Description
Program/DBtype	This field indicates that the description text appears on a PFKS screen. Enter `PFKS` in this field.
Screen-ID/View Name	If you wish to create an entry that is displayed on all **PF Key** screens, you must leave the Screen-ID field blank. To define description text for one particular PF KEY screen, you must specify the Screen-ID. The Screen-ID field may contain one of the following literals: `IDENTIFY`, `EDIT-TABLE`, `EDIT-ROW` or the Menu name.
Function	If you wish to create an entry that is displayed on all **PF Key** screens or if the Screen-ID contains the literal `MENU`, you must leave the Function field blank. The Function field may contain the name of any function that currently exists in the XXXXMENU table.
Field Name	This field may have the following values: DESC0, DESC1, DESC2, DESC3, DESC4,DESC5, FUNC-ID or any command name that is defined in the XXXXPFKS table, such as CANCEL or END. See below for more information on each of these options.
Description Display Length	Indicates the total length of the description text.
Description Display Offset	This field is ignored when the Program/DBtype is PFKS.
Description Display Attributes	Controls the visual presentation of the text on the screen.
Description Display EXTATT	Controls the visual presentation of the text on the screen.
Description Data	The description text that will appear on the screen.

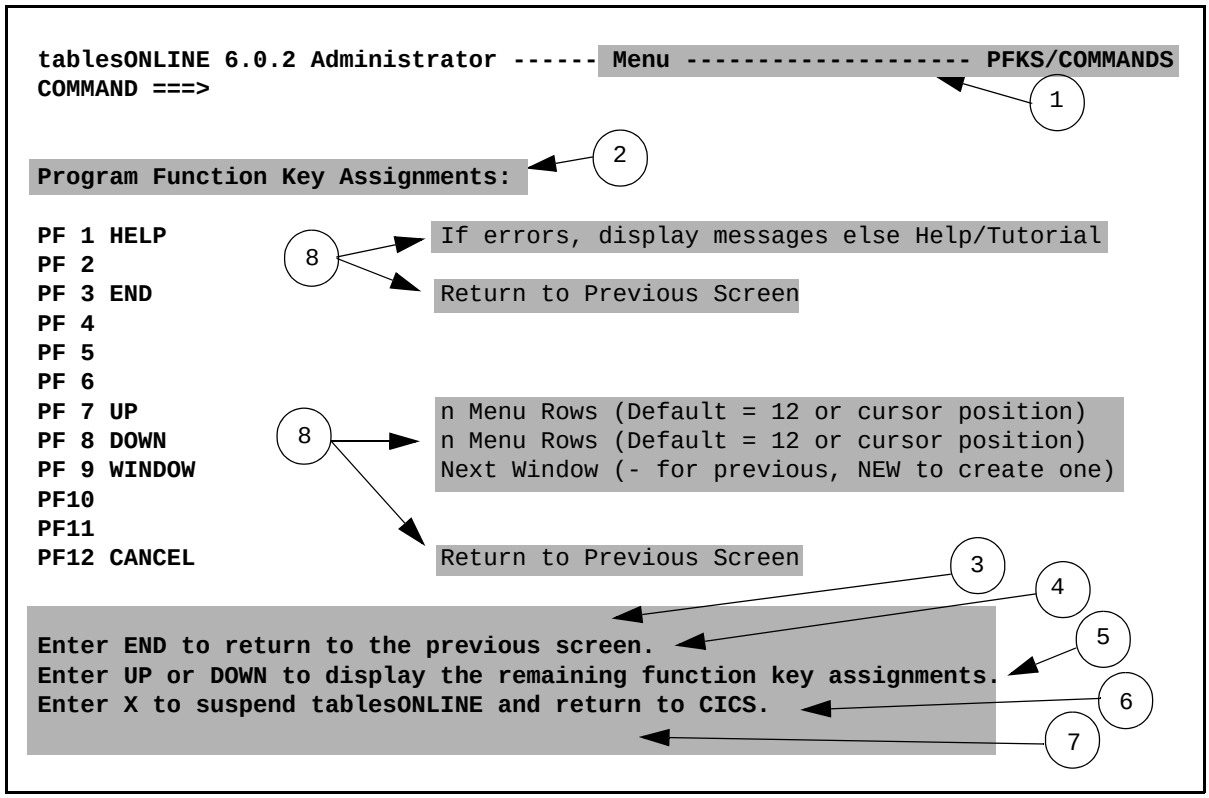

Figure 6-19: Sample PF Key Screen

The Field Name value determines which text on the screen is affected (see Table 6-21).

Table 6-21: Field Name Values for Program/DBtype PFKS

Field Name Value	Description
FUNC-ID	Literal that appears on the right half of the top line of every PF Keys screen (see Figure 6-19 #1). The Screen-ID field must contain one of the following literals `EDIT-TBL`, `EDIT-ROW` or `IDENTIFY`. The Function field may contain the name of any function that is currently defined in the XXXXMENU table.
DESC0	Literal that appears on the fourth line from the top of the PF Keys screen (see Figure 6-19 #2).
DESC1	Literal that appears on the fifth last line from the bottom of the PF Keys screen (see Figure 6-19 #3).
DESC2	Literal that appears on the fourth last line from the bottom of the PF Keys screen (see Figure 6-19 #4).

Table 6-21: Field Name Values for Program/DBtype PFKS (Continued)

Field Name Value	Description
DESC3	Literal that appears on the third last line from the bottom of the PF Keys screen (see Figure 6-19 #5).
DESC4	Literal that appears on the second last line from the bottom of the PF Keys screen (see Figure 6-19 #6).
DESC5	Literal that appears on the last line from the bottom of the PF Keys screen (see Figure 6-19 #7).
Any command that is defined on the XXXXPFKS table.	Literal that appears beside the corresponding command (see Figure 6-19 #8).

ERRORCHK

When an error occurs in data conversion at the field level, text retrieved from this type of Description table item is added to the error message to provide additional information. To define an ERRORCHK item in the Description table, the only additional key field that must be defined is the Field Name which in this case contains an error code. The error number is the numeric value returned by the data conversion routine.

Note: The description text for the ERRORCHK items are likely to be changed only in the case of localization for a particular language.

NO-DESCR (No Description)

When a Description table lookup fails for the field name values FUNC-ID or APPL-ID, the Description Data from the NO-DESCR row is displayed on the screen in place of the expected description. There is currently only one item of this type in the Description table. It displays the "No Description Found" message.

Note: During initialization, tablesONLINE checks for the NO-DESCR item on the Description table for the application. If the item is not found, it will not run that application.

Program/DBtype for DBtype Class Value

This section details the setting of the Description Key hierarchy for the DBtype Class listed in Table 6-10. There is only one DBtype Class value - TABLE.

TABLE

Rows in the Description table that have TABLE as the Program/DBtype are used to display specific text on the **Table Identification, Edit-Table** and **Edit-Row** screens. This includes the display of text on screens used to edit the tablesONLINE System tables and on **Utility** screens, like **Edit the Description Table** and the **Copy Table** screen.

The content of these screens is largely determined by the values of the table that is being edited. However, description text can be associated with any field by entering into the Field Name field the name of the field as defined in the View (see Figure 6-20).

The following **Copy Table** screen is an example of a screen that is displayed using the TABLE Program/DBtype.

The fields of Description table row should be filled in according to the information in Table 6-22.

Table 6-22: Description Table Field Values for TABLE

Field Name	**Description**
Program/**DBtype**	This field indicates that the description text appears on a Table screen. Type TABLE in this field.
Screen ID/**View Name**	To define description text for a specific table, this field must contain the name of the View of the table being edited or browsed.
Function	The Function field may contain the name of any table editing function that is currently defined in the XXXXMENU table. When the Field Name field is not blank, this field must be blank.
Field Name	This field may have the following values: FUNC-ID or the name of a field in a View. See below for more information on each of these options.
Description Display Length	Indicates the total length of the description text.
Description Display Offset	This field is ignored when the Program/DBtype is TABLE.
Description Display Attributes	Controls the visual presentation of the text on the screen.
Description Display EXTATT	Controls the visual presentation of the text on the screen.
Description Data	The description text that will appear on the screen.

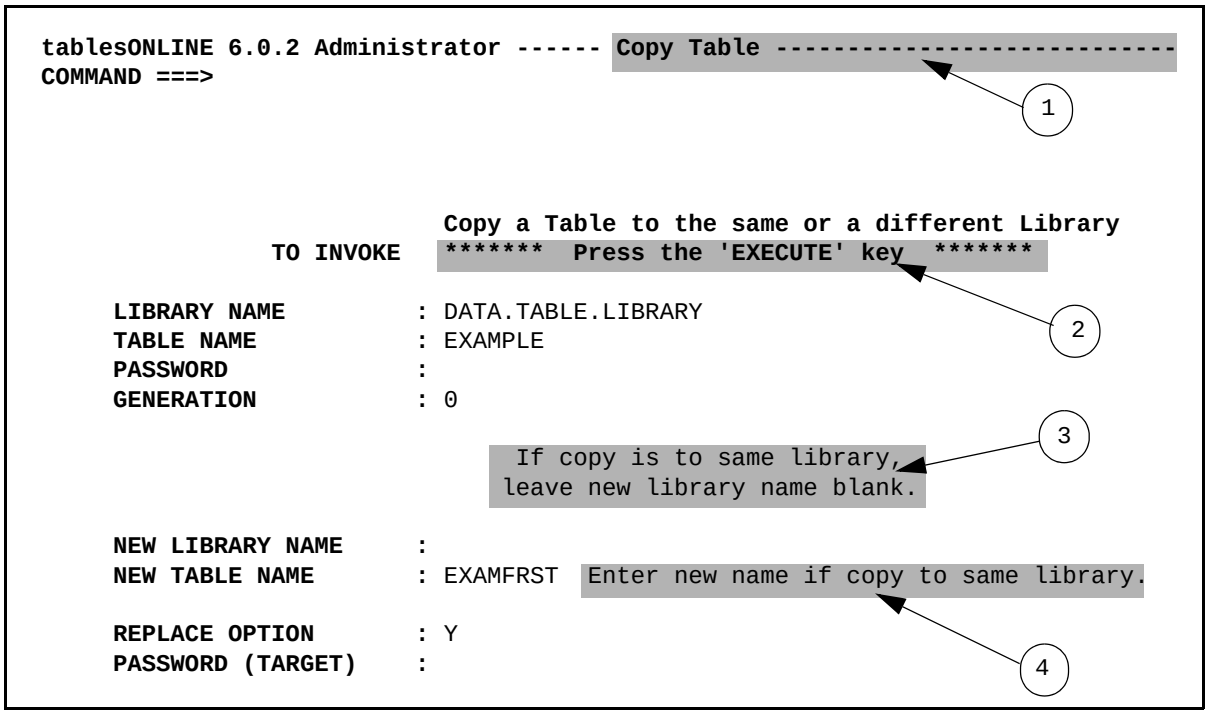

Figure 6-20: Sample Copy Table Screen

The Field Name value determines which text on the screen is affected (see Table 6-23)

Table 6-23: Field Name Values for Program/DBtype TABLE

Field Name Value	Description
FUNC-ID	Literal that appears on the right half of the top line of every table editing screen (see Figure 6-20 #1). The Function field may contain the name of any function that is currently defined in the XXXXMENU table.
The name of a field in a View	Location of the literal on the screen is determined by the location of field with which the description text is associated. For examples, see #2, #3 #4.

Note: The description text may be associated with a concealed field, that is, a field that does not appear on the screen (see Figure 6-20 #3). To create a concealed field, that is, to specify that a field name is not to appear on the Edit-Row screen, see "Display Features" on page 56.

Library Table

tablesONLINE allows you to set up aliases for your tablesONLINE libraries. An alias is simply another name for a library that may be used in tablesONLINE more easily than the library dataset name. These aliases are stored in the Library table. To edit this table, select Option 7 - EDIT LIBR TABLE from the Application Developer's menu and press <Enter>. This displays a screen where you can insert an alias. If this field is left blank, Figure 6-21 is displayed.

```
tablesONLINE 6.0.2 Administrator ------ Edit Table ---------------------------
COMMAND ===>

Table Name : TBOLLIBR                                      Row Location : 1

      DDNAME                    LIBRARY
     /DTFNAME                   DATASET
     ________ _____________________NAME_____________________
     TBSYSLB  SYSTEM.TABLES
     TBSYSLB  SYS
     TBSYSLB  S
     TBAPPLB  APPLICATIONS.DRIVING.TABLES
     TBAPPLB  APP
     MAINLIB  DATA.TABLE.LIBRARY
     MAINLIB  DATA
     MAINLIB  D
     TBACTLB  SECURITY.CONTROL.LIBRARY
     TBACTLB  SECURITY
     TBACTLB  ACT
     TBDICLB  DICTIONARY.LIBRARY
     TBDICLB  DICT
```

Figure 6-21: The Library Table Screen

Entries in this table allow the use of simpler or descriptive names that you may use to identify libraries. Once an alias has been set up in this table, you may access that library using the alias. In the example above, there are three entries for MAINLIB. The first of these entries is echoed back on the **Identification** screen, once a library has been selected. The remaining entries provide alternative library names that may be entered on the library selection field to avoid the need to enter long library names.

Libraries may also be identified by DDNAME or DTFNAME, depending upon whether you are working in an MVS or VSE environment. You may also enter a number to access the appropriate library in your library authorization list in the Application Control Table.

Note: Any DDNAME in the Application Control Table must have at least one corresponding entry in the TBOLLIBR table. If not done, a TB-5719 warning message results.

7 tablesONLINE/CICS Messages

Chapter 7 lists all the tablesONLINE/CICS messages.

Table 1: tablesONLINE / CICS messages and error codes

MSG #	Text	Meaning / Instructions
TB-2000 I	The program TBDRIVC terminated successfully.	Information only.
TB-2010 E	Transfer Error: Transfer transaction TRIN was not found.	TRIN is delivered as part of the TBOL product. Consult the tableBASE installation instructions.
TB-2011 E	Transfer Error: Invalid transaction.	Target transaction is not defined properly to CICS.
TB-2012 E	Transfer Error: User not authorized to execute the transaction.	Target transaction is not defined properly to CICS.
TB-2013 E	Transfer Error: Transaction is currently disabled.	Target transaction is not defined properly to CICS.
TB-2014 E	Transfer Error: TWA size too small.	Target transaction is not defined properly to CICS.
TB-2015 E	Transfer Error: Cannot retrieve PCT entry (Lock/Busy condition).	Target transaction is not defined properly to CICS.
TB-2016 E	Transfer Error: Target program not found or is currently disabled.	Target transaction is not defined properly to CICS.
TB-5000 E	Unexpected tableBASE error XXXX occurred while processing 'XX'.	Refer to the tableBASE error return codes for more information about errors processing command 'XX'. If problem not explained, contact Data Kinetics tableBASE support.
TB-5001 I	Enter Down command to view additional rows.	Information only.
TB-5002 I	Enter Up command to view additional rows.	Information only.
TB-5003 W	Enter Help command for a display of # lines of messages.	Information only.
TB-5004 I	Enter Up or Down command to view additional rows.	Information only.
TB-5005 W	You only have the authority to browse. Edit capability disabled.	Information only.
TB-5006 W	You do not have the authorization to perform this function.	Information only. Contact tableBASE administrator.

MSG #	Text	Meaning / Instructions
TB-5007 E	Indirect help could not be done because table could not be opened.	See tablesONLINE/CICS User's Guide for further information.
TB-5008 E	Indirect help could not be done because key could not be found.	See tablesONLINE/CICS User's Guide for further information.
TB-5009 E	Command 'XXXXXXXXXXXXXXXXXXX' is not supported for hash organization.	See tablesONLINE/CICS User's Guide for further information.
TB-5010 E	Duplicate keys allowed accepts only a Y or N response.	
TB-5011 E	The generation number supplied is incorrect.	Occurs when the generation does not exist.
TB-5012 E	Library: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx not on XXXXXXXX table. Contact your system administrator.	Contact your tableBASE administrator.
TB-5013 E	Library selection failed. You are not authorized to modify 'xxxxxxxx'. Please contact the tableBASE system administrator.	Contact your tableBASE administrator.
TB-5014 E	Table open failed, you may not access tableBASE library: xxxxxxxx.	
TB-5015 E	Indirect help not followed. Indirect points to indirect help.	Only one level of indirect help is supported.
TB-5016 E	Invalid COMMAND: 'XXXXXXXXXXXXXXXXXXX' for multi-user update mode.	See tablesONLINE/CICS User's Guide for further information.
TB-5017 W	There are no other windows established, enter 'NEW' to create another Window.	Or you can enter NW on the command line.
TB-5018 W	This View restricts you to browse only. Edit capability disabled.	Contact your tableBASE administrator.
TB-5019 E	Table 'xxxxxxxx ' is accessed from a Read Only VTS region. Try Browse.	You accessed a table residing on VTS as a result of the VTS name appearing in the tableBASE library list. If you want to EDIT this table, your profile on the TBOLACT table must be changed. Contact your tableBASE administrator.
TB-5020 W	The last field has been searched. To repeat from start press FINDFLD.	The fields are searched from left to right or top to bottom starting from the cursor position. This message appears when the end of the fields have been reached.
TB-5021 W	The field name starting with ' ' is not found.	
TB-5023 E	View key field only accepts an 'Y' or 'N' entry.	
TB-5029 I	Changes to the view 'xxxxxxxx' have been cancelled.	Information only.
TB-5030 E	The count entered is outside the table. Count of last row is xxxxxxxx.	TBOL returns the last available row.
TB-5031 I	Changes to table 'xxxxxxxx' have been cancelled.	Information only.
TB-5032 I	The previous row examined has been deleted.	Information only.
TB-5033 I	The row has been added to the table.	Information only.
TB-5034 I	A row with this key is already on the table. The row has been added.	Information only.
TB-5035 I	The row has been updated.	Information only.
TB-5036 I	A row with this key is already on the table. The row has been updated.	Information only.
TB-5037 E	A row with this key is already on the table. Update rejected.	
TB-5038 W	The count entered is outside the table. You are positioned at bottom.	
TB-5039 E	The key entered does not match any row key on the table.	Re-enter command as required.
TB-5040 E	The exit program '********' could not be loaded.	Exits from TBOL are defined in one of two places: in the View of the table being edited/browsed or in the TBOLACT table if processing commands.

MSG #	Text	Meaning / Instructions
TB-5041 W	The count entered is outside the table. You are positioned at the top.	Re-enter command as required.
TB-5042 W	The first available row is:	Warning occurs when attempting to locate a row before the previously displayed rows and a user exit program suppresses the display of rows according to the exit logic.
TB-5043 W	The last available row is:	Warning occurs when attempting to locate a row after the previously displayed rows and a user exit program suppresses the display of rows according to the exit logic.
TB-5044 I	Table already in use. Multiple user access in effect.	Information only. Multi user access is a View attribute.
TB-5045 W	This row is currently in use by 'xxxxxxxx' in the 'aaaa' application.	When attempting to edit a row in use by another user xxxxxxxx.
TB-5046 I	Table accessed by multiple users has been closed.	Information only. Multi user access is a View attribute.
TB-5047 I	Multiple user access in effect. Others may be browsing or editing.	Information only. Multi user access is a View attribute.
TB-5049 E	Cannot close table 'xxxxxxxx'. The table is open for write.	Another TBOL Window , or an unrelated process is holding this table opened for write. Contact your system administrator.
TB-5050 E	The password supplied is invalid.	
TB-5051 W	Changes have been made to table 'XXXXXXXX'. To confirm that you indeed wish to cancel the changes: Select 'CANCEL' To avoid having all the changes lost: Select 'ENTER' or 'END'.	Information only. Follow message instructions as required.
TB-5052 W	Changes have been made to table 'XXXXXXXX'. To confirm that you indeed wish to save the changes and create a new generation: Select 'ENTER' or 'END' To cancel the changes that have been made: Select 'CANCEL'. To go back to editing without saving or canceling Enter 'RESHOW'.	Information only. Follow message instructions as required.
TB-5053 E	Changes have been made to table 'XXXXXXXX'. Changing the library, table, generation or password can only be done after work on the present generation is complete. In order to save the changes that have been made: Select 'ENTER' or 'END'. To cancel the changes that have been made: Select 'CANCEL'.	Information only. Follow message instructions as required.
TB-5054 W	A request to delete row(s) from the table has been entered. To confirm that you indeed wish to delete the rows: Select 'ENTER' or 'END'. To cancel the delete request: Select 'CANCEL'.	Information only. Follow message instructions as required.
TB-5055 I	Table 'XXXXXXXX' has been updated.	Information only.
TB-5056 I	View 'XXXXXXXX' has been updated.	Information only.

MSG #	Text	Meaning / Instructions
TB-5057W	You have made changes to table 'XXXXXXXX' which may have been updated by other users. Several users may be updating this table simultaneously. The 'CANCEL' command may not have any effect because another user may have saved the table. To undo any changes you made… In order to save your changes and any changes made by other users: Select 'ENTER' or 'END'.	Information only. Follow message instructions as required. Multi user access is a View attribute.
TB-5058 W	You have made changes to table 'XXXXXXXX' which may have been updated by other users. In order to save your changes and any changes made by other users: Select 'ENTER' or 'END' To avoid saving the changes and cycling another generation: Select 'CANCEL'. To go back to editing without saving: Enter 'RESHOW'.	Information only. Follow message instructions as required. Multi user access is a View attribute.
TB-5059 W	You have made changes to table 'XXXXXXXX' which may be being browsed by other users. Changing the library, table, generation or password can only be done after work on the present generation is complete. In order to save your changes and any changes made by other users: Select 'ENTER' or 'END'. To avoid saving the changes and cycling another generation: Select 'CANCEL'.	Information only. Follow message instructions as required. Multi user access is a View attribute.
TB-5060 W	You are not authorized to use the M2M command. Contact Data Kinetics tableBASE support.	Contact Data Kinetics tableBASE support.
TB-5061 W	A range of nnnnnn row(s) has been selected for deletion. Because some rows may be suppressed from view, the actual number to be deleted may be less. To confirm that you indeed wish to delete the rows: Select 'ENTER' or 'END'. To cancel the delete request: Select 'CANCEL'.	Information only. Follow message instructions as required.
TB-5062 E	Listing the contents of the VTS attached to xxxxxxxx is not supported.	Contact Data Kinetics tableBASE support.
TB-5063 E	Internal error '9999' creating a directory listing.	Contact Data Kinetics tableBASE support.
TB-5064 E	You are trying to create a duplicate primary key. Update rejected. Primary key fields are hi-lighted with a different color.	View Attribute
TB-5065 E	Delete enqueue error. Non-existing key: xxxxxxxxxxxxxxxxxxx	Record the Key. It may help pinpoint an issue.
TB-5066 E	The row with primary key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx has been deleted or moved by another user.	Record the Key. It may help pinpoint an issue.
TB-5500 W	Exit program called with invalid indicators:	User Exit error. Contact tableBASE administrator.
TB-5501 W	Field name is spaces.	
TB-5502 W	To indicate a key field code Y. Otherwise leave blank or code N. If the field is a Dynamic View Suffix, code S. If it is also a key field as well as the Dynamic View Suffix, code B. If this is a View for Editing a table using an Alternate Index and the Primary Key of the table is in a different location, code P. If a Primary Key Field and an Alternate Index Key Field overlap, code X.	

MSG #	Text	Meaning / Instructions
TB-5503 E	Invalid display or internal table format.	With Cursor on field, press Help (PF1).
TB-5504 W	Display length must be xxxx and has been substituted.	
TB-5505 E	Display length must be in the range xxxx through xxxx.	With Cursor on field, press Help (PF1).
TB-5506 W	Internal table length must be xxxx and has been substituted.	
TB-5507 E	Internal table length must be in the range xxxx through xxxxx.	With Cursor on field, press Help (PF1).
TB-5508 E	Invalid edit length code 'X' on TBOLHKFM table. Valid values are: blank, D, E, H, N, P, Q, 0.	
TB-5509 E	Invalid attribute. Pick N, C, F, D, Q, S, P, p, M, m, V, v, or blank.	With Cursor on field, press Help (PF1).
TB-5510 E	Invalid feature, must be D, N, Y, ':' or blank.	With Cursor on field, press Help (PF1).
TB-5511 E	Invalid action code. It must be Y, N, R, E, I, D, C, U, or blank.	With Cursor on field, press Help (PF1).
TB-5512 W	Program name 'xxxxxxxx' does not exist.	
TB-5513 E	Invalid direction, must be I for inbound to display or O for outbound.	With Cursor on field, press Help (PF1).
TB-5514 E	Invalid timing, must be B for before system action or A for after.	With Cursor on field, press Help (PF1).
TB-5515 E	Key fields must be contiguous. Check field ====>	
TB-5516 E	Display length must equal internal table length.	With Cursor on field, press Help (PF1).
TB-5517 E	Display length should equal table length or table length + 1.	With Cursor on field, press Help (PF1).
TB-5518 E	Display length should equal (2 * table length). Subtract 1 if no sign.	With Cursor on field, press Help (PF1).
TB-5519 E	Display length must equal 2 * table length.	With Cursor on field, press Help (PF1).
TB-5520 E	Display length should equal table length + 1 or table length + 2.	With Cursor on field, press Help (PF1).
TB-5521 E	Display length should equal (2 * table length). Add 1 if signed.	With Cursor on field, press Help (PF1).
TB-5522 W	Exit program entered without setting exit indicators.	User Exit error. Contact tableBASE administrator.
TB-5523 E	You may not move a field to a position before a key field.	In display mode of View edit the keys are always grouped first .
TB-5524 E	You cannot add new physical fields in display mode.	
TB-5525 E	You may not move a key field.	
TB-5526 E	You cannot delete a field in display mode.	
TB-5527 E	The number of key fields must be between 1 and 50 inclusive.	
TB-5528 E	The total length of the key fields must not be greater than 256.	
TB-5529 W	Application specific help table 'XXXXXXXX' not found on any library.	
TB-5530 E	Invalid field location for field: View may be corrupt - contact your technical support person.	If the error cannot be explained, contact Data Kinetics tableBASE support.
TB-5531 E	The Data Table is currently in use.	Not used in Version 6 TBOL
TB-5532 W	Data/Index 'xxxxxxxx' requires RSZ, KSZ, KLOC: nnnnnnnnnnnnnnnnnn, in order to conform to the View just defined. Change the values of the row size, key size & location of the physical Data Table to conform to the View. If this is a View for an Alternate Index the warning above may not apply. Please note the KEY SIZE, and KEY LOCATION so that these values can be used to verify the Alternate Index Definition - Option 6 on the Define Table and View menu.	
TB-5533 E	Data Table 'xxxxxxxx' is in use - try again later.	Occurs when attempting to edit a table.
TB-5534 E	You cannot change the keys or suffix in display mode.	For some applications, a different display order is possible. Contact Data Kinetics tableBASE support.

MSG #	Text	Meaning / Instructions
TB-5535 E	You cannot change fields to be key fields while in display mode.	Go to DEFINE VIEW on the "Define Table and View" menu to change key fields.
TB-5536 E	The total length of the key fields must be greater than zero.	
TB-5537 E	Number of delimiters must be equal the display length.	
TB-5538 E	The display mask must include a valid sign symbol.	This applies to numeric Display Formats N and 0 through 9.
TB-5539 E	The display mask must have 'X' decimal places.	This has to match the number of decimals places of the Display Format
TB-5540 E	The number of delimiters must be one less than the display length.	
TB-5541 E	The trigger field name supplied is not valid. A trigger field must be defined with an action code of Y, R or E.	Verify the spelling of the Trigger field name. It must be a field already defined in the View.
TB-5542 E	Invalid decimal point encountered at position 'XX'.	
TB-5543 W	Edit pattern only allowed for display format of 'X'.	
TB-5544 W	Invalid or incomplete pattern delimiter encountered at position 'XX'.	
TB-5545 E	Number of edit pattern symbols must be equal to display length.	
TB-5546 E	View 'XXXXXXXX' is in use by another application.	Someone else is editing the View.
TB-5547 E	The field format and/or length differs from source key field.	
TB-5548 E	Source field format and/or length differs from importation field.	
TB-5549 E	Source field name required for duplication action.	
TB-5550 E	The source field name entered already has a duplication action defined	If you want multiple duplicates, always take the value from a single source field.
TB-5551 E	Source field format and/or length differs from duplication field.	
TB-5552 E	Create date or update date action codes apply only to date fields.	
TB-5553 E	The source field is not a key field as required for action Y, R or E.	
TB-5554 E	The source field specified is not the entire key of the corresponding source Data Table.	
TB-5555 E	Source field name not defined in the view.	
TB-5556 E	Importation action requires a previously defined field with an action code of Y, R or E.	
TB-5557 E	Internal view alternate table access failed.	Contact Data Kinetics tableBASE support.
TB-5558 E	Source field must be the entire key of the Source Table.	
TB-5559 E	A FIELD FORMAT of P, F, or H cannot be used as a dynamic View suffix.	Only formats that define displayable characters are supported.
TB-5560 E	The source field name entered must not be the same as the target field	
TB-5561 A	A row on view 'xxxxxxxx' cannot be found. Internal error. Abort task.	Contact Data Kinetics tableBASE support.
TB-5562 E	tableBASE error 'XXXX' occurred creating internal table 'RSXXXXXX'.	You may have too small a TSR (tableSPACE Region).
TB-5563 E	tableBASE error 'XXXX' occurred storing internal table 'RSXXXXXX'.	Target library is Read Only.
TB-5564 E	The total length of all fields: 99999 exceeds 32,763 bytes.	32,763 bytes is the maximum row size.
TB-5565 W	The attribute V or v cannot be used with display format Y. The attribute has been set to a space.	
TB-5566 E	The target field must be 8 bytes to receive a DDNAME from the list or it must be 44 bytes to receive a DATASET NAME.	

MSG #	Text	Meaning / Instructions
TB-5567 E	The target field must be 8 bytes to receive a table or view name.	
TB-5568 E	More than one dynamic suffix. Check field ====>	
TB-5569 E	The total length of the Primary key fields must be less than 257.	The sum of the lengths of all fields comprising a key must be 256 or less.
TB-5570 E	The FIELD LENGTH/FORMAT must be '8' 'X' or 'U' respectively for the insert USERID action code 'B'	
TB-5571 E	'xxxxxxxx' is not a valid TBOL view.	Verify view, and re-issue.
TB-5600 W	Exit called for a table it doesn't know about. Table is xxxxxxxx.	
TB-5601 E	You cannot delete your own session. Action ignored.	Information only.
TB-5700 W	Utility processing is bypassed as the current operating mode is 'X'.	Contact Data Kinetics tableBASE support. Very likely the XXXXMENU table has errors.
TB-5701 W	Invalid action. Use S to select a utility.	Information only. Follow message instructions as required.
TB-5702 W	Table 'xxxxxxxx' has been copied as 'xxxxxxxx'.	Information only.
TB-5703 W	Table 'xxxxxxxx' has been renamed to 'xxxxxxxx'.	Information only.
TB-5704 W	Table 'xxxxxxxx' has been deleted.	Information only.
TB-5705 W	View 'xxxxxxxx' has been copied as 'xxxxxxxx'.	Information only.
TB-5706 W	View 'xxxxxxxx' has been renamed to 'xxxxxxxx'.	Information only.
TB-5707 W	View 'xxxxxxxx' has been deleted.	Information only.
TB-5708 W	A generation of table 'xxxxxxxx' has been deleted.	Information only.
TB-5709 W	User profile has been updated.	Information only.
TB-5710 W	Application table 'xxxxxxxx' has been copied as 'xxxxxxxx'.	Information only.
TB-5711 E	Function cannot be performed: Table 'xxxxxxxx' not found.	
TB-5712 E	Utility failed: Table 'xxxxxxxx' is currently in use.	Try again later, or contact tableBASE administrator.
TB-5713 E	Utility failed: Table 'xxxxxxxx' already exists on library.	
TB-5714 E	Utility: View 'xxxxxxxx' parameters not currently defined.	
TB-5715 E	Utility failed: view 'xxxxxxxx' already exists on the library.	
TB-5716 E	Utility cannot be performed: View 'xxxxxxxx' not found.	
TB-5717 E	Utility failed: Prefix required to generate a new set of tables.	
TB-5718 E	Utility failed: Invalid tableBASE library. DDNAME: xxxxxxxx.	
TB-5719 W	DDNAME xxxxxxxx not on 'xxxxxxxx' - contact your system administrator.	Contact your tableBASE administrator.
TB-5720 E	Utility failed: Invalid password supplied.	Try again, or contact tableBASE administrator.
TB-5721 W	Table 'xxxxxxxx' has been defined.	Information only.
TB-5722 W	Table 'xxxxxxxx' definition has been updated.	Information only.
TB-5723 W	Supplemental information of view 'XXXXXXXX' has been updated.	Information only.
TB-5724 E	Reducing row size may result in data loss. Press execute to continue.	Information only. Follow message instructions as required.
TB-5725 E	The passwords for table 'xxxxxxxx' have been changed.	Information only.
TB-5726 W	Alternate Index 'xxxxxxxx' has been defined for table 'xxxxxxxx'.	Information only.
TB-5727 E	Table 'xxxxxxxx' from a different library is currently in use.	
TB-5728 E	Illegal To prefix 'xxxx'.	
TB-5729 E	Blank table name. This is a required entry.	
TB-5730 E	Invalid generation.	

MSG #	Text	Meaning / Instructions
TB-5731 E	Function not performed - tableBASE library XXXXXXXX is full.	
TB-5732 E	tableBASE library must be entered.	
TB-5733 E	The RSZ, KSZ & KLOC have been updated to conform to an updated view. If these values are not required and the original values of the data are to be used, select ENTER again immediately upon return. Press EXECUTE to change the Data Table to conform to the VIEW. This could result in data loss if you are reducing the row size.	Information only. Follow message instructions as required.
TB-5734 W	Table is not an alternate. Fields filled in from the Data Table.	
TB-5735 E	Table 'xxxxxxxx' is an alternate for Data Table 'xxxxxxxx'.	
TB-5736 W	The RSZ, KSZ & KLOC have been filled in from the view.	
TB-5737 W	View 'xxxxxxxx' could not be found and was not renamed.	
TB-5738 W	Alternate Index 'xxxxxxxx' has been updated for table 'xxxxxxxx'.	
TB-5739 E	New name cannot be the same as old name.	
TB-5740 E	This operation is not supported for tables from/to the VTS server.	
TB-5741 E	Alternates should not be updated with this option. Use option 6.	The name used was that of an Alternate Index.
TB-5742 W	Utilities function error. There is no processing logic for row ID 'X'.	Contact Data Kinetics tableBASE support.
TB-5743 E	Utility failed: Invalid password supplied for target.	
TB-5744 E	The SYSIN parameter starting with ' xxxxxxxx . . . . ' cannot be found.	There is an error in the TBOLJCLx template table. Cannot identify the JCL statement. Contact Data Kinetics tableBASE support.
TB-5745 E	The JES spool data set could not be opened, reason code '9999'.	Contact your CICS system administrator.
TB-5746 E	The JES spool data set could not be closed, reason code '9999'.	Contact your CICS system administrator.
TB-5747 E	The JES spool data set could not be written to, reason code '9999'.	Contact your CICS system administrator.
TB-5748 E	The JCL template table, 'xxxxxxxx' does not exist. Please re-enter.	Re-enter command as required.
TB-5749 E	The variable ' ' in JCL statement 00000000 was not replaced.	There is an error in the TBOLJCLx template table. The template JCL statement is missing a search for value. Contact Data Kinetics tableBASE support.
TB-5750 E	The JCL statement starting with '//xxxxxxxx . . . . ' cannot be found.	There is an error in the TBOLJCLx template table.
TB-5751 I	Job 'xxxxxxxx' has been submitted to generate copybook 'mmmmmmmm'.	Information only.
TB-5752 E	The generated JCL from template 'xxxxxxxx' for table/view 'tvtvtvtv'has been submitted with an error. Please examine the JES queue to ensure that the generated JCL contain the correct program name and that the keyword substitutions have been accurately performed for this function.	Examine the JES queue and determine the error before re-submission. You may need to change the accounting information.
TB-5753 I	Job 'xxxxxxxx' has been submitted to print table 'ttttttt'.	Information only.
TB-5754 E	Alternate definition failed: Table 'xxxxxxxx' already exists. Use Execute with 'UPDATE' on command line to replace alternate table.	Information only. Follow message instructions as required.
TB-5755 E	Copy cannot be performed: Table 'xxxxxxxx' exists on target library.	

MSG #	Text	Meaning / Instructions
TB-5756 E	Update of alternate failed: Alternate table 'xxxxxxxx' does not exist.	
TB-5757 E	Copy cannot be performed: View 'xxxxxxxx' exists on target library.	
TB-5758 I	Job 'xxxxxxxx' has been submitted to print View definition 'vvvvvvvv'.	Information only.
TB-5759 I	The alternate definition name 'xxxxxxxx' is unavailable - (not found).	Information only.
TB-5760 E	Enter the name of a table that contains the restructuring rules.	Follow message instructions as required. This table would have been saved from a prior visit to the Data Restructuring Utility.
TB-5761 E	Re-enter a different name.	Re-enter a different name as required.
TB-5762 E	The restructuring table, 'xxxxxxxx' does not exist. Please re-enter.	Re-enter restructuring table name as required.
TB-5763 E	Enter Y or N to indicate the generations you want to restructure.	Information only. Follow message instructions as required.
TB-5764 I	The restructuring table, XXXXXXXX is not a valid restructuring table.	Information only. The restructuring table contains information to restructure tables. It can only be processed by TBOL.
TB-5765 E	Restructuring tbl: XXXXXXXX requires RSZ/KSZ/ KLOC:nnnnnnnnnnnnnnnnnn. The Data Table that has been entered cannot be restructured with this restructuring rules table.	The Restructuring table does not match the basic information of the data table that you are attempting to restructure.
TB-5766 E	Generation (-x) of the data has RSZ/KSZ/KLOC: nnnnnnnnnnnnnnnnnn. This does not conform to the latest generation. Restructuring aborted. Restructuring is possible with the most recent generation. Set the All Generations flag to N so only the most recent is restructured.	Older generations of the data table do not have the same layout as the latest generation from which the restructuring data was taken from.
TB-5767 I	N generation(s) of the table 'XXXXXXXX' has been restructured.	Information only.
TB-5768 I	The restructuring rules are saved in the table 'xxxxxxxx'.	Information only.
TB-5769 E	If the restructuring rules are to be saved for future use with other tables, enter a restructuring table name.	Information only. Follow message instructions as required. The name should be recorded so it can be used at a later time or deleted if it is never used again.
TB-5770 I	The data is already restructured. Use END or CANCEL to exit utility.	Information only.
TB-5771 E	Although you have elected not to restructure the Data Table(s) at this time, it is essential to enter a name to save the restructuring table for use at a later date. If this is not a requirement, enter CANCEL to avoid a repeat of this message.	Information only. Follow message instructions as required.
TB-5772 E	FLD: xxxxxxxxxxxxxxxxxxxx Error: One of the rows from the existing table cannot be converted.	
TB-5773 E	Data Table 'XXXXXXXX' cannot be stored, library 'XXXXXXXX' is full. Free up some space on the target library, return to this option (7), and try again.	Open up another Window and delete an unused table to free up some space

MSG #	Text	Meaning / Instructions
TB-5774	The name 'XXXXXXXX' entered is either an alternative index or a paged table and cannot be restructured directly. For alternate indexes restructure the underlying base Data Table. For a paged table, change it to an in-memory table using the utility TBEXEC.	For alternate indexes: Restructure the underlying base Data Table. For a paged table: Change it to an in-memory table using the utility TBEXEC.
TB-5775 E	The variable ' ' in SYSIN statement 00000000 was not replaced.	There is an error in the TBOLJCLx template table. The template JCL statement is unable to replace a search for value. Contact Data Kinetics tableBASE support.
TB-5777 W	The Multi-User table update option requires unique primary keys. If this is the VIEW for a table (not an Alternate Index), set the DUPLICATE KEY IND to 'N'. Press HELP for more details.	Follow message instructions as required.
TB-5778 E	The Multi-User table update option requires unique primary keys.	With Multi-User updates, TBOL cannot process tables that have duplicate primary keys.
	If this is the VIEW for an Alternate Index, set the DUPLICATE PRIME KEYS to 'N' when MULTIPLE USER UPDATE is set to 'M'.	With Multi-User updates, TBOL cannot process tables that have duplicate primary keys.
TB-5801 E	## - Action Error:	The ## identifies the field that has the Error: xxx.. Often pressing help gives you the list of values to pick from.
TB-5802 E	## - Import Error:	The ## identifies the field that has the Error: xxx..
TB-5803 E	## - Duplication Error:	The ## identifies the field that has the Error: xxx..
TB-5804 E	## - Mandatory Error: An EOF erase does not constitute a valid key stroke. Spaces are valid.	The ## identifies the field that has the Error: xxx..
TB-5805 E	## - Pattern: 'XXXXXXXX' does accept data as entered. Refer to legend below: Z - alphabetic; A - alphabetic or blank; 9 - numeric; I - numeric or blank; Y - alphabetic or numeric; X - alphabetic, numeric or blank; B - blank; C - any character (no validation); Literals - any set of characters between a pair of (!) marks. Use Help (F1) in Define View Option for more details.	The ## identifies the field that has the Error: xxx..
TB-5806 E	## - Verification Error:	The ## identifies the field that has the Error: xxx..
TB-5811 E	The source View 'xxxxxxxx' not found.	The Source View is required to locate the import data.
TB-5812 E	The source Data table 'xxxxxxxx' not found.	Data importation is not possible.
TB-5813 E	The source table 'xxxxxxxx' is password protected.	Data importation is not possible.
TB-5814 E	tableBASE error 'xxxxx' occurred opening Source Table 'xxxxxxxx'.	Refer to the tableBASE error return codes for more information.
TB-5815 E	Source Table, 'xxxxxxxx' must be in descending order for x' action.	Data importation is not possible.
TB-5816 E	The length of field 'xxxxxxxxxxxxxxxx' and Source Table 'xxxxxxxx' key field length do not match.	Data importation is not possible.
TB-5817 E	Source field 'xxxxxxxxxxxxxxxx' not found in table 'xxxxxxxx'.	Data importation is not possible.

MSG #	Text	Meaning / Instructions
TB-5818 E	Source field 'xxxxxxxxxxxxxxxx' in table 'xxxxxxxx' is not a key.	Data importation is not possible.
TB-5819 E	Source field 'xxxxxxxxxxxxxxxx' in source view 'xxxxxxxx' has a different length than the field defined in the view.	Data importation is not possible.
TB-5820 E	Source field 'xxxxxxxxxxxxxxxx' in source view 'xxxxxxxx' is not the entire key of the corresponding source Data Table.	Data importation is not possible.
TB-5821 E	Source field 'xxxxxxxxxxxxxxxx' in source view 'xxxxxxxx' has a different format than the field defined in the view.	Data importation is not possible.
TB-5822 E	tableBASE error 'xxxx' occurred creating internal table 'SPxxxxx'.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5823 E	tableBASE error 'xxx' occurred emptying internal table 'ECxxxxxx'.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5824 E	tableBASE error 'xxxx' occurred creating internal table 'ECxxxxxx'.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5825 E	tableBASE error 'xxxx' occurred deleting internal table 'ECxxxxx'.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5826 E	tableBASE error 'xxxx' occurred deleting internal table 'SPxxxxx'.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5827 E	Upper range 'xxxxxxxxxxxxxxxx' in source view 'xxxxxxxx' has a different length than the lower range field.	Data importation is not possible.
TB-5828 E	Upper range 'xxxxxxxxxxxxxxxx' in source xxxxxxxx' has a different format than the lower range field.	Data importation is not possible. Fix the format mismatch first.
TB-5829 E	Lower range 'xxxxxxxxxxxxxxxx' in source view 'xxxxxxxx' has no corresponding upper range field for action R.	Data importation is not possible. Fix the
TB-5830 E	Internal get storage failed for data importation processing.	Data importation is not possible.
TB-5832 E	View data importation pointers corrupted.	Contact Data Kinetics tableBASE support.
TB-5833 E	View data duplication pointers corrupted.	Contact Data Kinetics tableBASE support.
TB-5834 E	Upper range in source Data Table 'xxxxxxxx' is too short or missing.	Data importation is not possible.
TB-5835 E	Initial value does not conform to the display format or edit pattern.	
TB-5836 E	Highest bound does not conform to the display format or edit pattern.	
TB-5837 E	Lowest bound does not conform to the display format or edit pattern.	
TB-5838 E	The Lowest bound value is greater than Highest bound value.	
TB-5839 E	Both the Highest and Lowest bound values must be specified.	
TB-5840 E	The Initial value is not between the Highest and Lowest bounds.	
TB-5841 E	Initial value has longer length than the display length.	
TB-5842 E	Lowest bound value has longer length than the display length.	
TB-5843 E	Highest bound value has longer length than the display length.	
TB-5844 E	## - Value Lower than bound: 'lllllllllllllllllllllllllllllllll'. Please enter a value greater than the lower bound indicated.	The ## identifies the field that has the Error. Enter a value greater than the lower bound indicated.

MSG #	Text	Meaning / Instructions
TB-5845 E	## - Value Greater than bound: 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'. Please enter a value lower than the upper bound indicated.	The ## identifies the field that has the Error. Enter a value lower than the upper bound indicated.
TB-5897 E	QIDERR error: command code 'xxxx' response code 'xxxxxxxxxxxx'. A CICS related operational resource is unavailable. The table directory listing was aborted. Please try HELP again (PF1).	The table directory listing was aborted. Please try HELP again (PF1).
TB-5898 E	QIDERR error: command code 'xxxx' response code 'xxxxxxxxxxxx'. A CICS related operational resource is unavailable. You cannot continue to edit the table. Table rows may have been updated. In order not to lose any changes, You must exit the table editor and save any changes. After exiting this message screen, press END (PF3) from the Identify Table/Row screen.	Exit the table editor and save any changes. After exiting this message screen, press END (PF3) from the Identify Table/Row screen.
TB-5899 E	QIDERR error: command code 'xxxx' response code 'xxxxxxxxxxxx'. A CICS related operational resource is unavailable. You cannot continue to edit/browse the table. After exiting this message screen, press END (PF3) from the Identify Table/Row screen and try editing or browsing the table at a later time.	Try editing or browsing the table at a later time.
TB-5900 W	Help message unavailable, message 'xxxxxxx' not on table 'xxxxxxxx'.	
TB-5901 E	Unable to open table 'xxxxxxxx', reason code 9999.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5902 E	Unable to invoke alternate table 'xxxxxxxx', reason code 9999.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5903 W	Help information for table 'xxxxxxxx' is not found on table 'xxxxxxxx'	
TB-5904 W	DESC: 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx' not on 'xxxxxxxx'	This could occur when a DESCription with key has been deleted.
TB-5905 A	Error creating table 'XXXXXXXX', reason code 9999. Job aborted.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5906 W	Tutorial data 'xxxxxxxxxxxxxxxxxxxxxxxx' not found on table 'xxxxxxxx'.	
TB-5907 E	This PF key has not been defined on the PFK table.	
TB-5908 W	This application does not have items defined on menu table 'xxxxxxxx'.	Refer to the tablesONLINE/CICS User's Guide for creation of Menu Items.
TB-5909 W	Help data 'xxxxxxxxxxxxxxxxxxxxxxxx' not on table 'xxxxxxxx'	Refer to the tablesONLINE/CICS User Guide for creation of Menu Items.
TB-5910 E	Transfer program 'xxxxxxxx' of menu option 'xxxx' not found/ disabled.	Refer to the tablesONLINE/CICS User Guide for creation of Menu Items.
TB-5911 E	Blank transfer parameters for menu option 'XXXX'	Refer to the tablesONLINE/CICS User Guide for creation of Menu Items.
TB-5912 E	Invalid COMMAND or OPTION: 'xxxxxxxxxxxxxxxxxxxxxxxx', please re-enter.	Re-enter command as required.
TB-5913 E	Transfer option 'xxxxxxxxxxxxxxxx' is not found on menu table 'xxxxxxxx'.	Refer to the tablesONLINE/CICS User Guide for creation of Menu Items.
TB-5914 W	The value entered for this command should be numeric.	Re-enter command as required.
TB-5915 W	Additional tutorial information is not found on table 'xxxxxxxx'.	

MSG #	Text	Meaning / Instructions
TB-5916 E	The keys are protected. The row cannot be deleted in this mode.	
TB-5917 E	Command or option 'xxxxxxxxxxxxxxxx' not supported for browse.	See tablesONLINE/CICS User's Guide for further information.
TB-5918 E	Command or option 'xxxxxxxxxxxxxxxx' unsupported for hash tables.	See tablesONLINE/CICS User's Guide for further information.
TB-5919 E	There are no rows on the table.	
TB-5920 E	View 'xxxxxxxx' cannot be found on the library searched.	
TB-5921 E	View 'xxxxxxxx' is empty.	
TB-5922 E	Unable to open view 'xxxxxxxx', reason code 9999.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5923 W	Dynamic open of view 'xxxxxxxx' failed, reason code 9999.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5924 W	Dynamic open failed. View 'xxxxxxxx' not found.	
TB-5925 E	tablesONLINE system table 'xxxxxxxx' is not on any library searched.	
TB-5926 E	Unable to open tablesONLINE system table 'xxxxxxxx', reason code 9999.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5927 E	tablesONLINE system table 'xxxxxxxx' is empty.	
TB-5928 E	tablesONLINE system table 'xxxxxxxx' is in use by another application.	
TB-5929 W	Dynamic open failed. View 'xxxxxxxx' is empty.	
TB-5930 E	## - Conversion Error:	The ## identifies the field that has the Error: xxx..
TB-5931 E	## - Conversion Error:	The ## identifies the field that has the Error: xxx..
TB-5932 E	Key field ## - Conversion Error:	The ## identifies the field that has the Error: xxx..
TB-5933 E	View 'xxxxxxxx' is in use by another application.	
TB-5934 E	Data Table 'xxxxxxxx' cannot be stored, library 'xxxxxxxx' is full. In order not to lose the work just completed, enter another Window and free up some space by deleting unnecessary tables. Once this is done: Select 'ENTER' or 'END'. To cancel the changes that have been made: Select 'CANCEL'.	Follow message instructions as required.
TB-5935 E	Data Table 'xxxxxxxx' is in use by another application. Try browse.	
TB-5936 E	View 'xxxxxxxx' cannot be stored, library 'xxxxxxxx' is full. In order not to loose the work just completed, enter another Window and free up some space by deleting unnecessary tables. Once this is done: Select 'ENTER' or 'END'. To cancel the changes that have been made: Select 'CANCEL'.	Follow message instructions as required.
TB-5937 E	View defines row size of 00000 which differs from table RSZ of 00000.	In order to use this View, you must make the Key size, Row size and Key location match the Data table.
TB-5938 E	View defines key location 00000 which differs from tbl location 00000.	In order to use this View, you must make the Key size, Row size and Key location match the Data table.

MSG #	Text	Meaning / Instructions
TB-5939 E	View defines key size of 0000 which differs from table key of 0000.	In order to use this View, you must make the Key size, Row size and Key location match the Data table.
TB-5940 E	Data Table 'xxxxxxxx' is not on any library searched.	
TB-5941 E	Default Data Table 'xxxxxxxx' is not on any library searched.	
TB-5942 E	You cannot create 'xxxxxxxx'. The table already exists.	
TB-5943 W	Dynamic open failed. View row size 00000 does not match table 00000.	In order to use this View, you must make the Key size, Row size and Key location match the Data table.
TB-5944 E	View 'xxxxxxxx' cannot be found. Enter NEW command to create view.	Enter NEW on the command line to create an View.
TB-5945 W	The table in memory is not the latest generation.	This occurs if another User or application has the table Open for Read in the TSR and you are trying to open the latest copy from the library, that was updated by another application since the table was opened in this TSR.
TB-5946 E	The table in TSR is not the latest generation. Update access denied.	This occurs if another user or application has the table Open for Read in the TSR and you are trying to open the latest copy from the library, that was updated by another application since the table was opened in this TSR. Close the table in the TSR and try the edit again. The tableBASE administrator can go to option 2 on the Administrators Menu to find the User that has the table open.
TB-5947 E	Table is in use by another application. Changes cannot be cancelled.	
TB-5948 W	Help information for menu 'xxxxxxxx' is not found on table 'xxxxxxxx'.	
TB-5950 E	The keys are protected. A new row cannot be inserted in this mode.	
TB-5951 A	This application is now continued at another terminal.	
TB-5952 A	This application is now suspended by the tableBASE administrator.	Contact your tableBASE administrator.
TB-5953 E	View defines Primary Key at 00000 which differs table location 00000.	In order to use this View, you must make the Key size, Row size and Key location match the Data table.
TB-5954 E	View defines Primary Key size of 0000 which differs table key of 0000.	In order to use this View, you must make the Key size, Row size and Key location match the Data table.
TB-5957 W	Line command 'xxxx' at row count ' ' requires target A or B.	
TB-5958 W	Line command 'xxxx' at row count ' ' requires action command.	
TB-5959 E	A hash table organization cannot support a block row delete.	
TB-5960 E	The line commands are conflicting. Row location(s): , .	Examine the conflicting location and place the line command appropriately.
TB-5961 I	Block command 'xxxx' at row count ' ' requires target A or B.	Information only.
TB-5962 I	Block command 'xxxx' at row count ' ' is incomplete.	Information only.
TB-5963 E	Invalid line command 'xxxx', please re-enter.	Re-enter command as required. Typically this occurs if you are using an update line command such as U, N or D while in BROWSE mode.

MSG #	Text	Meaning / Instructions
TB-5964 E	A sequential or hash organization of 'X' cannot support a row move operation. The organization must be U or R. This can be accomplished with a change definition command.	Use option 3 on the Table Define menu to change the table to be user ordered.
TB-5965 I	The number of rows deleted is	Information only.
TB-5966 I	The number of rows moved is x to location y.	Information only.
TB-5967 I	Line commands have been cleared.	Information only.
TB-5968 W	Line commands following the selected row have been deleted.	Under certain organizations multiple row line commands are not supported.
TB-5969 E	A table organization of 'X' cannot support a NEW nn command (insert at location nn operation). The organization must be 'U' (User ordered, Pointer or True) or the organization must be 'R' (Random ordered, True which is changed to User ordered by tablesONLINE). NOTE: A Random Pointer table is edited using a sequential index. In order to change the table organization, use option 3 on the Define Table menu.	
TB-5970 W	Table is empty. Enter data here. To leave table empty select CANCEL.	
TB-5971 E	Line command 'XXXX' invalid in multiple user table access. If you are trying to Delete a row, select the row for Update using the 'U' line command. Once the row is selected, type DELETE in the command line. (DEL or DE will also work.)	Because other users may be editing items, selecting an item first ensures that you have complete control first before deleting the selected row.
TB-5972 E	Multiple line commands are not supported for hash tables.	The rows are physically in random order.
TB-5973 E	tableBASE error 'xxxx' occurred storing table/view 'xxxxxxxx'.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5974 E	Operation suppressed. You are not licensed to extend use of tableBASE.	
TB-5975 E	Editing of VTS server tables is not supported. Try browse.	You accessed a table residing on VTS that was accessed as a result of the VTS name appearing in the tableBASE library list as determined by your profile on the TBOLACT table. Contact your tableBASE administrator.
TB-5980 E	The table could not be opened because the Data Table 'xxxxxxxx' is not a pointer table. The Data Table must be defined as a pointer table in order to be accessed as an alternate indexed table.	This occurs when the underlying data table is not defined as a pointer table. Change the INDEX attribute of the Data table to be pointer.
TB-5985 A	Error updating table 'xxxxxxxx', reason code 9999.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5986 A	Table TBOLACT must be upgraded to Release 6. See Installation Guide.	See Installation Guide for information on updating tablesONLINE/CICS.
TB-5987 A	Table TBOLPROF must be upgraded to Release 6. See Installation Guide.	See Installation Guide for information on updating tablesONLINE/CICS.

MSG #	Text	Meaning / Instructions
TB-5988 A	Table TBOLCNST does not contain the default row, SIGNON terminated.	An important 'DEFAULT ....' row is missing from the TBOLCNST table. This row must exist to launch TBOL for this CICS region. Contact Data Kinetics tableBASE support.
TB-5989 A	Table TBOLMRO does not contain the default row, SIGNON terminated.	An important 'DEFAULT ....' row is missing from the TBOLMRO table. This row must exist to launch TBOL for this CICS region. Contact Data Kinetics tableBASE support.
TB-5990 A	An error occurred when attempting to terminate your session.	Contact your tableBASE administrator.
TB-5991 A	Unable to open table 'xxxxxxxx', reason code 9999. Job Aborted.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5992 A	Transaction work area 99999999 too small. It should be 99999999. Abort	Contact your CICS system administrator.
TB-5993 A	Your application is not defined to the system, SIGNON terminated.	Contact your tableBASE administrator.
TB-5994 A	Descript. '****NO DESCRIPTION ' not on tbl 'xxxxxxxx'	An important '**** NO DESCRIPTION' row is missing from the XXXXDESC table. This row must exist to launch TBOL. Contact Data Kinetics tableBASE support.
TB-5995 A	Error creating table 'xxxxxxxx', reason code 9999. Job aborted.	Refer to the tableBASE error return codes for more information. If problem not explained, contact Data Kinetics tableBASE support.
TB-5996 A	tablesONLINE incomplete environment.	Check for further tablesONLINE or tableBASE messages.
TB-5997 A	Message '5900 W Help message unavailable, . . . ' not on tbl 'xxxxxxxx'	An important '5900 W Help ....' row is missing from the XXXXMSGS table. This row must exist to launch TBOL. Contact Data Kinetics tableBASE support.
TB-5998 W	Unable to save user profile. tBASE COND:	This can happen if the tableBASE administrator has the TBOLPROF table open for edit
TB-5999 A	CICS error: command code 'xx' response code 'xxxxxx'. Job aborted.	Contact your CICS system administrator.
TB-6000 A	The VTS region 'xxxx' needs to be started before launching TBOL.	Contact your tableBASE administrator.
TB-6001 E	The VTS region 'xxxx' needs to be started before 'tttttttt' can be opened. Another approach is to edit the TBSYSVTS table that specifies in which VTS region to open the table.	Contact your tableBASE administrator.
TB-9000 E	Exit program 'xxxxxxxx' called with invalid indicators: xxxxxxxxxxxx.	Contact your tableBASE administrator.
TB-9001 W	Table xxxxxxxx called exit with indicators: xxxxxxxxxxxx in test mode.	Contact your tableBASE administrator.
TB-9999 A	CICS error: command code 'xx' response code 'xxxxxx'. Job aborted.	Contact your CICS system administrator.
TBA1000 E	There is no audit control record for this table. Contact administrator	Contact your tableBASE administrator.

Table 2:

For a list of tableBASE ERROR messages see the tableBASE Programming Guide.

Appendix A

TBAUDIT

If you have licensed the tablesONLINE/CICS interface for tableBASE, a sample special exit program TBAUDIT is included with the product. TBAUDIT is customer submitted program that is now supported by Data Kinetics Ltd.

TBAUDIT captures all changes (before and after images) made to any and all Data Tables or the associated Alternate Indexes over time. It also captures the date, time and the User Id of the user making the change. In addition, if the user decides to cancel the updating process the corresponding entries in the AUDIT table(s) are also cancelled. Please note that Version 6 of tablesONLINE/CICS supports simultaneous updates from multiple users, the TBAUDIT exit however has not been upgraded to support this new feature.

To install this exit you will need the following components from TBDIST.SRC:

1. the source program TBAUDIT in TBDIST.SRC.

2. the EXITWS and EXITPARM copybooks in TBDIST.SRC.

Please compile the program as delivered and define the appropriate Program Properties in the PPT table of CICS before attempting to incorporate the additions discussed below.

Once the EXIT is present in the CICS region, all of the xxxxMSGS tables need to be updated for the proper functioning of the exit, an audit control table, TBAUDCTL, has been defined and an audit table will need to be set up for every Data Table requiring auditing. We have included table EXAMAUD as the audit table and the EXAMPLE table as the table to be audited to serve as a sample for demonstration purposes. The EXAMPLE, EXAMAUDT and TBAUDCTL tables as well as their corresponding Views have been included in the sample MAINLIB shipped with tableBASE.

By activating the TBAUDIT exit in a View, the View for any table controls the auditing process. This is accomplished via Option 2 on the Define Tables menu of tablesONLINE. Please see the program source of TBAUDIT for setting the required values.

1. the xxxxMSGS tables each require a row to be added to represent a message with a key TBA1000 and an error severity of E. This message appears when there is no valid

entry in the TBAUDCTL table. The user will not be able to process (edit) this table if the audit function cannot be performed.

2. the audit control table TBAUDCTL is used to specify both the audit table name for any given Data Table and the passwords for the audit table. This table serves a further function as a process control table for all the Data Tables undergoing audit while being edited. The exit loads this table into the tableSPACE region, renames it to TBSYSAUD and inserts the TBLBASE command area for each table being audited. The rename is very important so that several regions can share this table. The TBAUDCTL table must be password protected as it contains the passwords of all the audit tables. In the delivered exit, the password for this table is READIT. The layout of the TBAUDCTL table can be seen by browsing the View table.

Once you have tested the exit as delivered, you may wish to consider the following two enhancements:

1. The creation of a more secure audit control mechanism.

   In the exit the audit control table is renamed as TBSYSAUD and is then used to contain the TBLBASE command area each audit table. However, because the table contains the passwords for the audit table, the passwords can be seen by a sophisticated user who can scan the TSR using CICS utilities or TBDRIVC.

   Our recommendation is to define the TBAUDCTL table as having only two fields: the Data Table name and the AUDIT table name. (i.e., do not keep the passwords in TBAUDCTL.)

   After the change name command (CN), the table is expanded to contain the command area and continues to be used as before. TBAUDIT should be changed to generate the MASTER password which is then used to open the AUDIT table.

2. The use of a Journal to keep the audit records.

   TBAUDIT can be changed so that the audit information is written to a CICS journal. If this approach is adopted, it will still be necessary to use an audit table. Under this situation, the audit table would be a temporary table that could be assigned the Session Id as part of its name. This insures uniqueness and works well if a user has multiple windows open. Because the user has the option of canceling table updates, the contents of the temporary AUDIT table would only get copied to the CICS journal after a successful store (ST) operation of the Data Table.

Note: Using a CICS journal may impact performance, depending on how many stores (ST) of audited tables are done and how fast CICS executes the journal write.

