tablesONLINE CICS
Users Guide

Release 5.1

Technical Support Hotline (613) 523-5588

To order additional manuals:

Data Kinetics Ltd.
Sales Division
2460 Lancaster Road
Ottawa, ON
Canada K1B 4S5

Telephone	(613) 523-5500
Facsimile	(613) 523-5533
E-Mail	tablebase@dkl.com

Data Kinetics Home Page http://www.dkl.com

Table of Contents

Preface

What this Manual is About

This manual describes tablesONLINE, the online user interface to tableBASE, in a CICS environment.

Who this Manual is For

This manual is intended for:

- Analysts doing online table design and definition
- Programmers doing online table maintenance and testing and requiring interactive access to tableBASE calls
- End users doing online table maintenance

What you Should Know to Use this Manual

You should be familiar with an MVS or VSE environment and basic CICS sign-on procedures.

What this Manual Contains

Chapter 1

This chapter provides an overview of some of the concepts and structures used by tablesONLINE.

Chapter 2

This chapter provides the basic information required to use tablesONLINE. It includes the sign on and sign off procedures as well as information about navigation through Menu screens.

Chapter 3

This chapter describes how to edit or browse data tables using tablesONLINE. This is essential pre-requisite reading as it will familiarize you with the tablesONLINE M2M editor that is used for the remainder of this manual.

Chapter 4

This chapter 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 are the option for creating COBOL copybooks for tablesONLINE tables, the printing of Views, the restructuring of data tables and the specification of M2M (Many to Many) relationships.

Chapter 5

This chapter describes the use of the tablesONLINE utility facilities, including copy, delete, rename, passwords and the user profile.

Chapter 6

This chapter describes how to build table driven applications by editing the application driving tables delivered with tablesONLINE.

Chapter 7

This chapter provides a reference list of all messages that may be returned by tablesONLINE.

Additional tableBASE References

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

- Concepts and Facilities Manual
- tableBASE Batch Utilities Manual
- tableBASE Administrator's Guide
- tableBASE Programmer's Guide
- tableBASE Installation Manual
- tableBASE VTS Server Guide
- tablesONLINE/ISPF User's Manual
- tableBASE Workshop Manual
- Quick Reference Guide

Terminology

The following is a list of important terms that are used in this manual.

View

For purposes of clarity, this term now replaces the earlier FDT or Field Definition Table. All references to FDT have been removed from the balance of this manual and are replaced with the term View. A View must be associated with a data table if that data table is to be edited using tablesONLINE. The View contains all of the information about the layout and structure of the data table contents.

M2M

This is a short form used to described the “Many to Many” relationship between Views and Data that is now supported by tablesONLINE. Prior to Release 5, the product allowed the use of multiple Views with a single data table. It is now possible to define relationships between multiple Views and multiple data tables. This means that one View can be used with multiple tables or multiple Views can be used with a single table. These relationships are defined using the Define M2M option that allows you to create an entry in the M2M table for each Table/View combination.

Row

This term Row replaces all earlier references to Item in tablesONLINE.

Table Object

This term is used to identify the name given to the Table/View combination that is created using the M2M facility. A unique Table Object name is assigned to each Table/View combination defined in the M2M table. Once defined, the combination can be referred to simply by the Table Object name.

Chapter 1 Introduction

This chapter will provide you with an overview of tablesONLINE. First, we will look at what tablesONLINE is, what it can be used for and by whom. We will then briefly discuss the interaction between tablesONLINE and tableBASE. Finally, we look at the library structure that is used by tablesONLINE.

What is tablesONLINE

tablesONLINE 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. 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.

tablesONLINE handles data entry and table editing tasks and provides the access controls and data validation services essential to such tasks. Two of its components, the menu system and the table editor, serve as a framework for building applications.

The Menu System

The menu system manages the control flow of the various programs that make up tablesONLINE itself. To define tables, you simply make selections from the menu and fill in the blanks. The menu system can also:

- Call any non-tableBASE program
- Initiate any transaction and
- Manage complex sequences of these actions.

It provides a convenient framework for managing any system that can be described in terms of sequences of independent actions.

The Table Editor

The table editor provides a framework for editing and displaying data. It includes an extensive set of built-in edit and display features and also offers an exit program capability to enable you to solve any problems relating to the editing and displaying of data.

Types of Users

tablesONLINE provides a wide variety of facilities, and is highly configurable and customizable. Because of these factors it can be used by many different levels of users to perform a wide variety of tasks. Each level of user can use the facilities that they require to perform tasks such as the following.

- **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 can be used to interactively 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 as well as other application programs.

Where 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 recompilation simply by editing the tables that control it. This significantly reduces maintenance programming in many applications.

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 is providing table-driven control over the applications a user, or group of users, can run and over the data they can access.

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 even minor changes. This arrangement improves service to users while reducing the data processing department workload.

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

- Adding help screens or tutorial material appropriate to local usage
- Calling your own validation routines during data entry
- Calling other routines to perform security checks, calculations or whatever operations are appropriate to your application
- Setting up tables which control user access to the system.

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 the 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. It can also be made completely invisible, placing the user directly into an application without showing any menu.

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

tablesONLINE provides facilities for creating data definitions for each data table. These definitions, called Views, are managed by tableBASE and used by tablesONLINE to control the display and editing of data tables. Facilities are provided to create and edit both Views and data tables, to display both Views and data, and to perform utility functions, such as copying or renaming Views and data tables.

Data validation is performed, whenever tables are edited under tablesONLINE, by extensive validation 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 can be used:

- tableBASE batch utilities such as TBEXEC
- CICS applications that invoke tableBASE via the interface routines TBLBASE, TBCALLC, TBASEC
- Non-CICS applications using the tableBASE modules listed later in this chapter

tablesONLINE and tableBASE

tablesONLINE and tableBASE work together to provide the tools necessary to build applications and the tables with which they 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.

There are many ways to access the tables created using tablesONLINE. The following is a list of the methods that may be used in a CICS environment:

- **TBLBASE**
  tableBASE calls from within CICS programs

  Note: TBCALLC and TBASEC are supported for backward compatibility with previous tableBASE releases.

- **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.

Detailed information on the use of TBLBASE and TBDRIVC are provided in the ***tableBASE Programmer's Guide***.

tablesONLINE Libraries

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 software. You may wish to store the tables that you create in these libraries or you may wish to create your own library to house your organization's tables. The following is a list of the delivered libraries together 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 has 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 (TBACTLB) contains the Application Control Table, or ACT, used in system administration, security control and user environment definition. In addition, 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 information 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 (TBAPPLB) 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 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 Table Libraries

These libraries are created by users to contain the data tables unique to your 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 - corporate data, divisional data, departmental data, etc.

- **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 the application-driving tables in the Application Driving Tables Library, and
- Operates on data from both corporate and private data table libraries.

Chapter 2
Getting Started

This chapter provides you with the basics for using tablesONLINE. It begins with a broad overview of logging into and exiting from tablesONLINE and 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 that tablesONLINE allows an organization to reconfigure the PF keys used by tablesONLINE/CICS. All of the descriptions and references to PF keys in this manual refer to the default PF key values used in the product as delivered. If your organization has customized these keys, you may find that the PF keys do not perform the functions described in this manual. Please check with your tableBASE Administrator to determine if this is relevant to your organization.

Starting tablesONLINE

You can access tablesONLINE by providing the appropriate CICS transaction code. "TBOL" is the default setting; however, it may be changed by your administrator. You should consult your administrator to obtain the necessary code. Depending on your requirements, your system will be set up so that on start-up you will access a specific tablesONLINE menu. If you are an application developer, you will likely see a series of menus beginning with the Application Developer's Menu displayed below.

```
tablesONLINE 5.1.0 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.
```

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.

Screen Types

There are four basic screen types used in tablesONLINE. You should be familiar with each of these screens 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.

Screen	**Usage**
Menu	Selecting tablesONLINE actions
Identify Table	Selecting a table
Edit/Browse Table	Selecting rows for browsing or editing Multi-row display One row per line
Edit/Browse Row	Actual data browsing or editing Single-row display One field per line

These screens are fully described later 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, it is simply a matter of stepping back through the screens until you reach your start-up menu. This is done using the PF3 (End) key. Pressing PF3 (End) from your start-up menu will exit the program, returning you to CICS where you can proceed with other tasks or sign off the system using your standard CICS sign-off procedure

Using Help

tablesONLINE provides context-sensitive help. If help is requested with the cursor positioned on a blank command line, then 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 controls 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 in tablesONLINE is table-driven and may be modified by the user. For information on editing help tables see Chapter 6, *Building Applications*.

Levels of Help

Help can be obtained by either typing HELP on the command line or pressing PF1. There are four different levels of help. Typing HELP, or pressing PF1 repeatedly, will move you through these levels in the following order.

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 indicates:

"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 bypassed 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 the new or occasional user.

Using Help from the Command Line

You can specify what type of help is required by inserting one of the following options on the command line and pressing PF1.

Option	Displays
TABLE	Help on the current table
MENU	Help on the current menu
ABC	Help 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 have been 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 is still Help available that describes the contents of the field. To view Help for a protected field, use the Up and Down arrows to move the cursor to the protected field and press PF1 (Help).

Later in this manual you will see that it is possible to define protected fields for your own applications.

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. These keys may be redefined for each application. The default PF key assignments used in the command menus of tablesONLINE, as distributed, are shown on the following screen.

```
tablesONLINE 5.1.0 ---------------------- 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 Items (Default = 12 or cursor position)
PF 8 DOWN      - n Menu Items (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.
```

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 in the screen above are the keys that are active at all times. The following list contains the complete set of default tablesONLINE PF keys.

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
Note: 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 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 certain fields in place so they do not scroll.

tablesONLINE supports up to 24 PF keys. In the distribution version of tablesONLINE, there are 24 PF key entries. Keys 1 to 12 and key 21 are as listed above with keys 13 to 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 may 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 have no effect.

Using Multiple Windows

The Window command allows you to create multiple windows and move from one window to another, in either direction. This command is available from any screen in tablesONLINE. In the distributed product PF9, the Window key, is used for this purpose. The parameters for this command are:

Command	Parameter	Short Version	Affects
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 records a user's environment. The initial setup of options in this table is done by the system administrator. Following that setup there are a number of options that you can set based on your own preferences. This includes options such as 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. This option does not affect error and warning messages, which are always displayed regardless of the setting of this option.

These options can be modified at any time by entering the following command or short form on the command line:

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 row.
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. The following is a sample of the format that the menu screens will take.

```
tablesONLINE 5.1.0 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
T    TRANSFER TO USER     - Transfer to User Application Menus
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.
```

The following menus are delivered with tablesONLINE:

- Administrator's Menu contains entries for all of the administrative functions that may be performed.
- Application Developer's Menu contains entries for all of the functions required to create an application.
- Utilities Menu contains entries for all of the tablesONLINE utilities that may be performed.
- Table Definition Menu contains entries for creating and editing the format of tables.
- Help/Tutorial Menu contains entries for editing all of the tablesONLINE help tables.
- User Applications Menu contains entries for all of the User Applications that have been defined. 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 (UP) and PF8 (DOWN) keys. You can also type the commands UP and DOWN directly on the command line and press <Enter>.

Scrolling up and down by screen is sensitive to cursor position. Scrolling will place the line containing the cursor at the screen boundary. To scroll a specific number of lines, type the desired number of lines on the command line and press PF7 or PF8.

Chapter 3
Editing and Browsing Tables

In the previous chapter we looked at navigating through the menu system. In this chapter we will discuss how to navigate through the tablesONLINE editor. Familiarity with using the tablesONLINE editor for Editing and Browsing will prepare 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. The chapter concludes with a discussion of your options upon completing your editing of a particular table.

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 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 will discuss 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

You select a table to browse or edit a table in the same way. In both cases you enter the selection process from the Application Developer's Menu (options A – Edit Table and B – Browse Table). You will be presented with a screen, known as the Identify Table/Row screen. It will look like this:

```
tablesONLINE 5.1.0 Administrator ------ Edit --------------- Identify Table/Row
COMMAND ===>

Please indicate the table required by entering the parameters below.
 Data Table Library          ==> DATA.TABLE.LIBRARY
 View Library (If different) ==> DICTIONARY.LIBRARY
 Table Object==> EXAMPLE
                                                    Generation ==> 38

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

LAST NAME            ===>
FIRST NAME           ===>









Dupl Keys Allowed?   ===> Y
```

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 these fields are left as they first appear on your screen and the Table Object name is modified, tableBASE will search the specified library. If the table object is not found, then the list of authorized libraries will be searched for the specified 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.

If you would like to see a list of all the table objects that you are authorized to access, use the List Facility described below.

Using the List Facility

The List Facility provides a fast, efficient way to display and select the libraries or tables available to the user. Once the list is displayed, it is possible to select the required library or table and have it automatically inserted in the field.

This facility is accessed by inserting the cursor in any table or library field and pressing PF1 (Help).

Library Lists

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 (Help). 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, described later in this chapter.

Table Lists

If you wish to see a list of tables in the specified library, clear the Table Object field and press PF1 (Help). This will display a list of all authorized Table Objects in the specified library.

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 (Help) with the cursor in the Table Object field. This will display a list of all authorized 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. (e.g., T*B* will list all Table Objects whose name contains a T in the first position and a B in the third.)

Identifying a Library

Throughout tablesONLINE you will use the Identify Table/Row screen 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. In addition to the List Facility, there are four ways to specify a library name:

- You can insert a DDNAME
- You can insert a Dataset Name
- You can specify a library alias
- You can specify a number from 1 to 6.

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 application driving table for libraries.

Or, you can identify a table in a library name field by inserting a number from 1 to 6 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 six 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, although displayed for reference, may 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 passed over.

Table and View Relationships

As with previous versions of tablesONLINE, 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 M2M (Many to Many). This facility, described in the next chapter, allows for the definition of relationships between multiple Views and multiple data tables.

These relationships 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. The combination may then be referred to in the Identify Table/Row screen using only the Table Object name. This feature simplifies the referencing of tables by removing the need to know both the name of the data table and the appropriate View. Users need to know only the Table Object name by which the combination is identified.

The default mode is for tablesONLINE to run with M2M turned on. This affects the Identify Table/Row screen for both browse and edit. If M2M is turned off, you can specify 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 only done 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 the Define M2M section of Chapter 4, *Define Tables*, for details.

The ability to control whether M2M is 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, this can be done in the following manner:

- Should you wish to turn M2M off, type M2M OFF in the command line and press <Enter>.
- When you wish 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. The balance of this chapter discusses these functions without any further mention of M2M.

Passwords

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

If a table has a read password, you will not be able to open the table in either Browse or Edit until the required password has been provided. If a table has only a write password, you will be able to open the table in Browse, but not in Edit, until the required password has been provided.

Specifying Table Generations

Each of the Identify Table screens in tablesONLINE 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 to a library with an absolute generation number from 1 to 255, starting again at 1 when it reaches 255. The table definition specifies the number of generations to be kept (maximum 9). 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. This is the default.

If generations 2, 4 and 5 are available, then generation 5 is the current generation.

You can specify either a relative or an absolute generation number:

Using Relative Numbers	Indicates	
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.

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 (e.g., 0), reading in the previous generation (e.g., -1) and saving it gives you a valid current generation. The previously current generation (0) becomes the previous generation (-1); thus, if only three generations are being kept, this also has the effect of leaving a corrupt generation (-1). A safer course of action is to delete the corrupt generation with the tablesONLINE delete utility.

Duplicate Keys Control

Your tableBASE administrator can allow you to enter rows containing duplicate keys. He does this by allowing duplicates when defining

1) the VIEW for this table and
2) the menu selection used to access this screen.

If both conditions prevail, the 'duplicate keys allowed field' will be present on the edit screen 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.

Specifying a Row Key

If the information contained in the Identify Table/Row screen when you first enter it is correct, the bottom of the screen will display fields that represent the key fields for the specified table. If you have modified the information in this screen, as described above, to select a different table to browse, you must press <Enter> to refresh the screen to display the key fields for 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.

If a row key is given and an exact match for it is found on the table, then the Edit/Browse Table screen is bypassed and the Edit/Browse Row screen is brought up immediately. Otherwise, the Edit/Browse Table screen with its multi-row display comes up so that you can select a row there. For paged tables, if an exact match is not found, the Identify Table/Row screen will remain on the display screen with a message indicating that the row could not be found.

Edit/Browse Table

An Edit/Browse Table screen displays the rows of a table in a horizontal manner with field names displayed at the top of each column.

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.

```
tablesONLINE 5.1.0 Administrator ------ Edit Table ----------------------------
COMMAND ===>

Table Name : EXAMPLE                                    Row Location : 1

             LAST             FIRST      DIVISION DEPARTMENT CHARITABLE     DAT
             NAME              NAME                            DONATION       OF
     ____________________ _____________ ________ __________ __________ CONTRIB
     AAGIN                JOHN          A/R      ACCTG       156.00     19870
     ALLEN                GORDON        OPR      OPNS        180.00     19870
     ANCHRUTHER           DORA          SECT     ADMIN       153.00     19870
     ASSIGNY              MICHEL        SECT     ADMIN        83.00     19870
     AXOLOTLOVOVITCHSKI   STEFAN        FLACK    ADVTG       345.00     19870
     BAKER                JOHN          NEWS     ADVTG       130.00     19870
     BELLEFEVILLE         JOHN          ART      ADVTG       125.00     19870
     BLOGGINS             JOHN          ANAL     MIS         230.00     19870
     BROWN                FREDDA        PERSNL   ADMIN       160.00     19870
     BROWN                FREDERICK     PERSNL   ADMIN       160.00     19870
     BROWN                GEORGE        P/R      ACCTG       143.00     19870
     BROWN                IAN           PGMG     MIS         210.00     19870
     BROWN                JOHN          SCH      OPNS         97.00     19870
     BROWNBAG             SAM           A/R      ACCTG       143.00     19870
     CALLAN               MICHAEL       PGMG     MIS         180.00     19870
++++ DALE                 TERENCE       OPR      OPNS         95.00     19870
```

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

There are several different methods that may be used in edit and browse to move through the rows in your table. 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 that the following descriptions assume that the PF keys, as delivered with tablesONLINE, have not been changed by your system administrator.

Scrolling

If the table you are editing is too large to fit on the display screen of your terminal, you will see a row 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 using:

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 right and left using the following PF keys:

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 leftmost 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 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 fields of your table remain on the screen. To do this, use the following:

PF21	FREEZEKEYS	Enter the number of fields to freeze or enter zero to unfreeze any frozen fields.

In the Edit/Browse Table screen, the frozen fields will remain at the left of the screen. The balance of the fields will scroll left and right leaving the frozen fields displayed at all times.

In the Edit 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

You can move to specific rows in the table using the following commands. These commands may be executed using the PF keys or the short form may be inserted in the command line and <Enter> pressed.

PF4	GETCNT	Enter the number of the row you wish to display in the command line. 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 If the command line is left blank, this command will display the next row in the table. For variants of this command see the **Additional Navigation Commands** below.
PF5	FINDFLD	Enter the full or partial field name of the field you wish to display.
PF6	GETKEY	Enter the full or partial key of the row you wish to display.

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.

GETCNT

GETCNT will retrieve and display the row that is identified by the count specified. 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 the following are some examples of the rows that will be returned based on the number inserted in the command line when PF4 is pressed.

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

In the Edit/Browse Table screen, FINDFLD will move the specified field into the first position to the right of any frozen fields. 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

Using GETKEY in an Edit/Browse Table screen will move the row matching the specified key into the second position on the screen. In an Edit/Browse Row screen, using GETKEY will display the row with the matching key in the Edit/Browse Row screen.

When using GETKEY, several keys may be listed 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. 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 you 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 very effective. It will find the first entry in the table whose key begins with 'S'.

If a partial key is specified, tableBASE treats it as a full key for searching purposes. Because of this, the partial key will not actually be found. Based on this practice, where tableBASE will position you in the table depends on the organization of the table involved. These rules also apply in cases where a full key is specified but not found. The following list describes where you will be positioned in the table following the specification of a partial key based on table organization.

Table Organization	**Partial Key Given But Not Found or Full Key Given But Not Found**
Sequential, Descending or Hash Pointer	Position in the table where the row would be, if it existed on the table.
Random or User Ordered	End of the table.
Hash True or Paged	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 on to another row. If there are no errors in the data for the current row, the next row will be displayed.

Additional Navigation Commands

The following commands may also be used to search for rows in a table. These commands are used by inserting the short form in the command line and pressing <Enter>.

Short Form	**Command**	**Description**
GN	GETNEXT	Displays the row that follows the currently displayed row.
GP	GETPREV	Displays the row that precedes the currently displayed row.
GF	GETFIRST	Displays the first Row in the table.
GL	GETLAST	Displays the last Row in the table.

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.

Once you have opened a table in the Edit/Browse Table screen you are now ready to begin selecting individual table rows to edit. This is done using line commands. These commands 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. The following codes may be used:

Code	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)
blank	Repeat Previous Line Command (S, U, or N) for row at the current cursor position.

S Select Row

This 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 View protect keys flag which, if set to N, would normally allow the editing of key fields.

U Update Row

Like the select row command, this line command allows you to select a row for editing. However, in this case, you may be permitted to edit key fields. Whether or not key fields can be edited will depend on the setting of the protect keys flag in the View. If the flag has been set to Y, the key fields cannot be changed.

If you bypassed 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** is 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, enter NEW on the command line and press <Enter>. If you then decide you wish to leave the original row unchanged, press <Cancel>.

N Create New Row

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

Both the **N** and **U** commands read in an existing row and allow it to be modified. The **U** command is used to actually modify an existing row, whereas, the **N** command leaves the existing row unchanged and creates a new row 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 merely pressing <Enter> will 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.

Note: If the protect keys flag is set to Y in the View, this operation will fail since this flag prevents the creation of new rows.

If the duplicate keys allowed flag in the View is set to N, or if the duplicate keys allowed flag 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.

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 (e.g., NEW 10 will create a new row and place it in the 10th position in the table.)

If **N** is 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 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.

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 to 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 Editor screens by entering INSERT on the command line. This will create an empty row that can then be completed.

Note: If the protect keys flag is set to Y in the View, this operation will fail since this flag prevents the creation of new rows.

If the duplicate keys allowed flag in the View is set to N, or if the duplicate keys allowed flag 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.

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 (e.g., NEW 10 will create a new row and place it in the 10th position in the table.)

D Delete Row

This line command will allow you to delete a row in the table. This operation will fail if the protect keys flag is set to Y in the View, since this flag does not allow the deletion of rows from a table.

When a row is deleted, you may be asked to confirm that you wish to proceed with the deletion. Whether or not this confirmation is required will depend on the setting of a flag in your User Profile. (See the User Profile section of Chapter 5, *Utilities*, for details.)

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 code.

The Move Command

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

The line commands required to do this are:

M Move this row

A Move to the position after this row

B Move to the position before this row

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

Move commands are restricted to Random and User Ordered tables as well as the updating of Views. They cannot be used with other table organizations since the organization of these other tables is pre-determined.

Operating on a Range of Rows

For the delete and move commands, it is possible to select a range of rows to be operated on, either by marking both ends of the range with a command (DD or MM) or by marking the beginning of the range and indicating the number of rows to be operated on. Commands of this type are:

Command	**Operation**	**Block**
DD	Delete	Rows enclosed by two markers
Dn	Delete	***n*** (1-999) rows beginning with the marked row
MM	Move	Rows enclosed by two markers
Mn	Move	***n*** (1-999) rows beginning with the marked row

To use the commands that mark both ends of a range, type DD or MM, respectively, to the left of the first row to be deleted or moved, and type DD or MM to the left of the last row to be deleted or moved.

To use the commands that specify the number of rows, type Dn or Mn, where n is the number of rows to be deleted or moved.

For a block move, you must also specify the position to which the rows are being moved, using either an A or B command. Once these commands have been entered, press <Enter> to complete the operation.

Multi-line delete commands may be used with any Resident (i.e., non-Paged) table that is not a hash table. As with the single line move commands, the multi-line move commands may be used only with Random and User Ordered tables.

Canceling Line Commands

To cancel any of the line commands, move commands or multi-row commands, either use the delete key to remove the line command, type over it with spaces or use:

PF12	CANCEL	to clear all entered line commands

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

Edit/Browse Row

An Edit/Browse Row screen is used to edit the actual data in a row. This screen will be 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 will be found under the heading: Navigating to other Rows - later in this chapter.

```
tablesONLINE 5.1.0 Administrator ------ Edit Row -----------------------------
COMMAND ===>

Table Name : EXAMPLE        Update Mode               Row Location : 6


     LAST NAME            : BAKER
     FIRST NAME           : JOHN
     DIVISION             : NEWS
     DEPARTMENT           : ADVTG
     SEX                  : M
     CHARITABLE DONATION  : 130.00
     DATE OF CONTRIBUTION : 1987-06-26
```

In the example displayed above, 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 scrolling keys as follows:

PF7	To scroll up
PF8	To scroll down

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	FINDFLD

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 flag 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. Your application may also require that the data that is input conform to additional types of editing criteria.

The first type of data validation performed is 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.

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 is entered. The data, if valid, is redisplayed 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.

The second type of data validation involves 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 will be 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 the values that may be selected.

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. Additional information on using field exit programs is provided later in this section.

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. 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

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.

Field Exit Programs

Field exit programs, if defined in the View, may be called each time data in the associated field is changed or displayed. 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 exits in tablesONLINE/CICS, see the *tablesONLINE/CICS Exit Programming* chapter of the ***Programmer's Guide***.

Navigating to Other Rows

There are several different methods that may be used from the Edit/Browse screen to move directly to different rows in your table.

Leaving a Row

There are several ways to leave the Edit/Browse Row screen:

Key	Command	Effect**
PF3	END	Update/create row and return to select* 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. The following shortcuts can also be used: GF, GL, GN and GP. 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.
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 the data and returns to the select* screen.

* The select screen is the screen where you selected the row. This may be the Edit Table screen or, if that screen was bypassed on entry to the editor, the Identify Table screen.

** 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.

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.

Changing or Creating a Row with Wrong Command Setting

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 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.

Similarly, 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, enter NEW on the command line and press <Enter>. If you then decide you wish to leave the original row unchanged, press <Cancel>.

The Execute Command

Entering EXCUTE on the command line, or simply 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.

The Delete Command

Entering DELETE on the command line, and pressing one of the following keys will also exit a row. Each of these keys has a slightly different effect on exit.

Key	Command	Effect
PF2	EXECUTE (DELETE)	Delete row, and if a hash table or at the end of the table, return to select screen* otherwise, present next row
PF3	END (DELETE)	Delete row and return to select screen*
PF12	CANCEL (DELETE)	Quit without deleting and return to select screen*
ENTER	DELETE	Delete row, if a hash table or at the end of the table, return to select

screen* otherwise, present next row

* The select screen is the screen where you selected the row. This may be the Edit Table screen or, if that screen was bypassed on entry to the editor, the Identify Table screen.

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, this will return you to either the Edit/Browse Table screen or the Identify Table screen. Pressing PF3 or END repeatedly, will return you to the tablesONLINE menu where you selected the editor.

Depending on the setting of the CONFIRMSAV flag in your user preferences, the screen below may be displayed so that you can confirm that you wish to save the changes you have made to the table. If the flag 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.

```
tablesONLINE 5.1.0 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'.
```

Chapter 4
Defining Tables

Table-defining information is maintained in two forms. Basic 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 have detailed 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.

If you are new to tablesONLINE and have skipped reading Chapter 3, Editing and Browsing Tables, it is strongly recommended that you read Chapter 3 so that you will benefit from knowing how to use the tablesONLINE editor, as that information is not repeated here.

Introduction

The basic table defining information, including table organization, search method and row size, is required by both tableBASE and tablesONLINE. In addition to this information, tablesONLINE requires more detailed information in order to allow you to browse and edit table rows online.

This detailed information includes information about:

- Where each field of data is positioned within a row
- Whether to interpret data as alphanumeric characters, a binary integer or some more complex data types such as a date
- Where to position data on the screen
- How to label data.

tablesONLINE sees tables in pairs. Each pair consists of a data table and a View containing the data definitions for the table. One data table may be paired with several Views, each of which defines a different layout of the underlying data. To the user, each View appears as a separate table.

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. However, it is possible 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. There are some naming conventions that must be followed if the table is to be used in tablesONLINE:

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

Although positions 2-8 may contain blanks, a table name may not contain embedded blanks.

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 up to 8 characters in length 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 Indicators later in this chapter.

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 Table Definition Menu that contains all of the options necessary to create and edit tablesONLINE tables.

In the Table Definition Menu, there are four options used to create a table:

- Define View, which describes the row layout
- Define View Supplementary, which describes table access controls
- Define Data Table, which describes the physical table attributes, and
- Edit Display Order, which allows you to change the order in which table fields are displayed to the user.

A table that is to be used in tablesONLINE is actually made up of two separate tables. The first is the data table and the second is the table's View. 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, if you wish 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 you with 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. The following items illustrate the type of information that can be specified:

- Field size and format for display (non-displaying fields are also possible)
- Field size and format for storage on table (fields that do not exist on the data table may be displayed - i.e., comment fields)
- Field-level editing and validation information
- Physical field order -- identical to definition order

This information is specified using the Define View option in the Table Definition Menu. The information provided is stored in a View. The View contains one row for each column 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 Table Definition Menu.

Define Data Table

This option allows you to define the overall 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 Table Definition Menu. The information provided is stored as a data table in the tableBASE library.

Field Display Order

This option allows you 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. This 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 following Table Definition Menu.

```
tablesONLINE 5.1.0 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 View Copybook
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.
```

Options 1, 2, 3 and 4 of this menu allow you to specify the types of information described in the previous section. These options may be selected in any order when creating your table.

Option D - Define All, has been provided to simplify this process. It includes options 1 - Define View, 2 - Define View Supplementary 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, will require 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, will allow 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 keys fields in the table. This requires the definition of an alternate index for the table.

Option 6 - Create Alternate, allows you to create alternate indexes 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, will allow you to generate a COBOL copybook for a table View. This copybook can then be used in an application program.

Option 9 - Define M2M, will allow you to define data table/View relationships that are identified by a unique name.

You use Option P – Print View, to submit a batch job which will print a formatted report of the view specified.

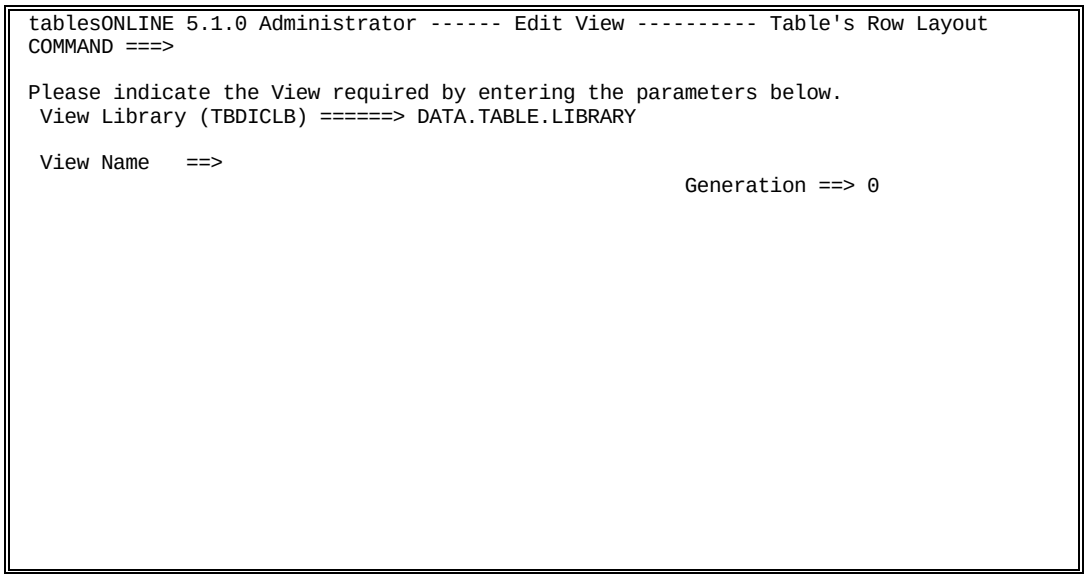
Each of these options is described in detail in the following sections.

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. You may wish to define a new View using the Define View option or you may wish to copy an existing View.

Creating 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.



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

You can now begin defining the fields that will make up your table. The next section, Defining Table Fields, describes how to define your table fields.

Copying an Existing View

Instead of creating a new View, you may wish to copy and modify an existing one. This is done using the Copy utility, which is fully explained in Chapter 5.

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 Supplementary, and 4 - Edit Display Order can all be used to modify an existing View as well as to define a View.

If you wish to edit an existing View, select 1 - Define View from the Table Definition Menu and press <Enter>. This will display the Identify screen where 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 **U** (Update) line command and press <Enter>. This will display 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) will create 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.

Defining Table Fields

Once a row has been selected with **S**, **U** or **N** on the Edit Table screen, it is displayed in a vertical format in the Edit Row screen.

```
tablesONLINE 5.1.0 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

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

     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
```

Some of the fields in this screen must contain information, while others may be left blank. The fields that must contain information are the Key Field Indicator, Display Length and Display Format. If these fields are left blank, default values will be inserted in these fields.

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 <Execute> to save that field. Repeat this process until you have defined all fields for your table and then exit the Edit Row screen using <End>.

Field Name-

This field contains the name of the field being defined. It may be left blank, in which case 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

This field indicates whether the defined field will form part of the key for tableBASE operations on the table. You indicate a key field by placing a 'Y' in the key field indicator; an 'N', which is the default value, is used to indicate that this is not a key field.

It is possible to define multiple key fields within a table. tablesONLINE allows you to designate up to 50 key fields within a table. All key fields must be contiguous and, when combined, may not exceed 256 bytes. The set of contiguous fields that make up the key can be located anywhere in the row. They do not necessarily have to begin in the first byte of the row.

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. 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. This row identifier is referred to as the Dynamic View Suffix.

For example, if the View name is TB068A00 and has TB068A as a prefix, the identifier is two bytes in length and contains the values 00 through 11, indicating 12 possible row layouts. The row identifier dynamically becomes the suffix of the View. Prior to processing a row with a suffix of 03, tablesONLINE 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.

The key field indicator is also used to indicate that a field contains the Dynamic View Suffix. If the Dynamic View Suffix forms part of the key, the key field indicator is set to B. If it is contained in a non-key field in the table, the key field indicator is set to S. A field defined as a Dynamic View Suffix may be up to eight characters in length.

The possible values for the key field indicator are the following:

Possible Values	Part of Key	Dynamic View Suffix	Indicates
Blank	No	No	Not a key
N	No	No	Not a key
Y	Yes	No	Key Field
S	No	Yes	Dynamic View Suffix
B	Yes	Yes	Key & Suffix

Only one field may be designated as a Dynamic View Suffix within a table. As a result, values S and B may only be used once within a table and they may not both be used within the same table.

Display Format and Field Format

These two options 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 and field formats. If only the display format is entered, the system will default the field format to the first value allowed for that display format. Conversely, if only the field format is entered, the system will default the display format appropriately.

The numeric display formats listed below can be combined with any of the indicated numeric field formats. Any date format can be combined with any of the other date field formats.

Display Format	Displayed As	Field Format	Stored As
X	Alpha-numeric	**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
E	ddmmmyy/ddmmmyyyy	**E**	FORMAT

If field format is omitted, the following table indicates the default value to which it will be set, based on the value inserted for display format.

Display Format Entered	Default Field Format	Default Rule
X	**U**	Translate lowercase alphabetics to uppercase
Y	**X**	Pack hex digits into characters, unedited
N,0,1,2,...9	**N**	Store numbers as zoned decimal
A	**A**	
B	**B**	Store dates in format used on input;
C	**C**	if LENGTHs are not entered, they
D	**D**	will default to the longer lengths
E	**E**	(e.g., year represented as CCYY)

If neither display nor field format is entered, they will default to X and U, respectively.

Display Length and Field Length

These two options are interrelated and dependent on the field and display formats. 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. The relationship between the display and field lengths depends on what the associated values for display and field formats are. The following table shows the interrelationships, as well as the minimum and maximum lengths allowed.

Display Format	Display Length		Field Format	Field Length
X	1 - 50		**X**	Same as display
			U	
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		**F**	4 (display length
:	& decimal point)			limited to 9 digits)
9	10 - 18 " "		**H**	2 (display length
				limited to 5 digits)
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

Because the display format is set within the table View, it is possible to create different layouts of your data table for different users by creating multiple Views,

each containing different display formats. 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:

Value	Display	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, e.g. password entry, suppressed on Edit/Browse table screen.
S	No	No	Suppress on Edit/Browse table screen.
	Yes	Yes	Display on Edit/Browse row 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. Must be updated

			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 updated if the row is newly created. Suppressed on the Edit/Browse table screen and displayed on the 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. 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, v are not displayed on the edit table screen used to select rows for editing but do appear on the edit 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. Additional information on the use of these types of text is provided below. Valid values for display features are:

Value	Display Fieldname	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	Display ruler, comments, table instructions or other invariant data
Y	Yes	Yes	Description	Table
N	No	No	Description	Table

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 look-up 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 Chapter 6, *Building Applications*, for more information about editing description tables.

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

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 and Field Format" above.
+ or **-**	+ 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 or CR.
( )	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.

Numeric data is filled from right to left. 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.

For display formats 0 to 9 or N, the numerics are filled in from right to left and the decimal point is placed in the exact location specified in the display mask.

If a numeric field (Display Format N) contains the data '3456789', the following table shows what would be displayed with several different display masks.

Display Mask	Produces
!,!!!,!!!.!!+	34,567.89+
Model: -!!!!!-!!.!!	Model: 345-67.89
Dfl -!.!!!.!!!,!!	Dfl 34.567,89
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.

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 alpha-numeric field contains the data, ABCD1234, the following table shows what would be displayed with several different display masks.

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.

Edit Pattern

The edit pattern field is used to specify an edit pattern for data entry. This edit pattern is then used to validate the data that is input into that field. 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. The following is a list of the acceptable edit pattern values.

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)
Literals	A set of characters surrounded by exclamation marks (!)

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

If a display mask is used in conjunction with an edit pattern, the following applies. If data is corrected after being displayed through a display mask, the pattern validation is performed as if the display mask is part of the edit. There is no need, however, to enter the display mask when entering data. The entered data, if acceptable, will be redisplayed with the display mask.

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'.

Action Codes

This is a multi-purpose field. It may be used to specify existence, exclusion or range validation, to specify the importation of data from external tables, or to specify automatic date updating. 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 the following table, together with the requirements for the other three fields:

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 (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
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.	No	No	No

Action Code	Action Performed	Source Table	Source Field	Trigger Field
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.	No	No	No
D	Duplicate Value. Duplicate the data into this field whenever the value changes in the source field.	No	Yes	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 (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.

The following action codes, when assigned to a table/library field, will display the described list when HELP is selected for that field.

Action Code	Action Performed	Source Table	Source Field	Trigger Field
L	List Libraries. Produces a list of authorized libraries from the TBOLLIBR table. If the field is defined to be 44 bytes in length, the Dataset Names will be listed and if the field is defined to be 8 bytes in length, the DDNames will be displayed.	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 this table or on another specified table (referred to 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 (Help).

If additional data is to be retrieved from the source table, refer to the Data Importation action code described below.

Exclusion Check (N)

Verify that the value entered in this field does not exist on this or on another specified table (referred to 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 example below). The first of the pair, the lower limit, is the key and the second, displayed to the right, is the upper limit.

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

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

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

If additional data is to be retrieved from the source table, refer to the Data Importation (I) action code described below.

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.
- In addition, a trigger field name must 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.

Data Duplication (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	Population

- User1 wishes to view the table with the key “City/Population”
- User2 wishes to view the table with the key “Population/City”.

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

State	City	Population	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/Population

Key 2 Population/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.

Create 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 Create 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 which is imported from 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 into the source field name, 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

The last three fields control the calling of a user exit program from the tablesONLINE editor. See the chapter on *tablesONLINE/CICS Exit Programming* in the ***tableBASE Programmer's Guide*** for information on the design and use of exit programs. Here we will only discuss the invocation rules.

This field allows you to specify the name of the exit program to be called by tablesONLINE when it operates on this field.

Field Exit Version

This option allows you to indicate whether the exit program specified has been created to take advantage of the new features in tableBASE release 5 or if it was created based on the requirements of the earlier 4.x releases. Enter 5 in this field to indicate release 5 or 4 to indicate releases 4.0 to 4.2.

Field Exit Input/Output Indicators

The indicator fields tell tablesONLINE when the exit program is to be called. The valid values for these indicator fields are:

Value	Exit program called
(blank)	Not at all
B	**B**efore the action
A	**A**fter the action
Y	**Y**es, both before and after the action

For example, when the editor is fetching 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 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 View trailer discussed in the following section.

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 (e.g., 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 (see Chapter 5, *Utilities*, for additional information) 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 handle 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.

Note: It is the user's responsibility to ensure that nothing is used that the local CICS installation, or the local terminal, cannot handle.

Supplementary View Information

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 Supplementary, from the Table Definition Menu. The following screen will appear:

```
 tablesONLINE 5.1.0 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        : Y2K UPGRADED
      USER ID              : DKLS02
      DATA TABLE NAME      :
      HELP TABLE NAME      : EXAMHELP

      DUPLICATE KEYS IND   : y
      PROTECT KEYS IND     :
      HASH RE-ORG CODE     :
      FREEZE KEYS COUNT    : 2
 ++++ ROW EXCLUDE IND      :
```

Note that this is the first of three screens that are used to enter supplementary information for a View. You navigate through the screens by using the PF7 and PF8 keys. Descriptions of all the fields you will encounter follow.

Library Name

Enter the name of the library where the View for which you wish to create supplemental information is stored. You may alternatively enter any alias that is defined in TBOLLIBR, a number from 1 to 6 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, blank out the name in this field and all authorized libraries will be searched.

There is a safety feature to prevent illegal 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 <End> or <Execute> 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, place the cursor in the View Name field and press PF1. If you clear the Library Name field and press PF1, a list of all libraries containing a View with the specified name will be displayed. If you clear both the Library field and 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 contain an entry indicating whether the selected View was created for release 4 or release 5 of tablesONLINE. An entry will automatically appear in this field when this screen is displayed.

Write 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 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 placed in this field is automatically generated by tablesONLINE based on the signon id of the individual creating 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 automatically. The List Facility can also be used here to find and select a data table name.

Help Table Name

Enter the name of the help 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.

Duplicate Keys Indicator

This option indicates whether duplicate keys can be created within a table. If duplicate keys 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.

Value	Display on Table ID Screen?	Initial Value	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 Re-Organization 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 that is set here is used as a default. The user may change it once the application is running.

Item Exclude Indicator

The item exclude indicator is used in conjunction with user exit programs that prevent certain rows from being displayed.

If you have an exit program that accesses a table where some rows have been excluded from the display 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 undisplayed rows, the item exclude indicator, when set to Y, excludes all non-displayed rows from the operation being performed.

Dynamic Suffix Location

The dynamic suffix location is automatically generated and inserted in 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** or a **B** key field indicator. The value contained in this field will identify the location of that field within the row.

Dynamic Suffix Length

The dynamic suffix length is automatically generated and inserted in 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 or a B key field indicator. The value contained in this field will identify the length of that field.

View Row Size, Key Location and Key Size

The row size, key location and 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 not, an error message is generated and the open fails.

Item Exit Name

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

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 or cross-validated, depending on the indicator that is selected. The following options allow you to specify under which circumstances the exit program is called.

Item/Table Exit Version

This option allows you to indicate whether the exit program specified has been created to take advantage of the new features in tableBASE release 5 or if it was created based on the requirements of the earlier 4.x releases. Enter a 5 in this field to indicate release 5 or a 4 to indicate releases 4.0 to 4.2.

Item Exit Input/Output/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, 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 as follows:

Value	Exit Program Called

(blank)	Not at all
B	**B**efore the action
A	**A**fter the action
Y	**Y**es, both before and after

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 hits <Enter> or takes any action that requires that the current row be output.

Table Exit Program

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 or closed. The following options allow you to specify under which circumstances the exit program is called.

Table Exit Open/Store/Close Indicator

Table exit indicators 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.

The set of permitted values for these item exit indicators are as follows:

Value	Exit Program Called
(blank)	Not at all
B	**B**efore the action
A	**A**fter the action
Y	**Y**es, both before and after

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 displayed below.

```
tablesONLINE 5.1.0 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 ITEMS          : 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
```

Note: 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 only updates the screen. If the library name and/or View name have been changed, then the definition of the newly specified table is displayed.

The <Execute> key is required to actually change the table definition. It is recommended that you press <Enter> first and verify that the screen has no errors before pressing <Execute>.

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 presented. 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 the authorized libraries, press PF1 (Help). This will display a list from which a library may be selected using the **S** line command.

Table Name

Enter the name that you wish to assign to the new data table. This name must not already exist on 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, press PF1 (Help). This will display a list from which a table may be selected using 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 chapter 3 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 organization option indicates how data entries are to be ordered within the table. You can choose from the following options:

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

For a list of the permitted organization/search method combinations, see the table in the following section on Search Methods.

R Random

With a random organization, new rows are added at the end of the table. When there is a deletion, the last row in the table is relocated to the vacated position. This means that overall data movement is kept to a minimum. Since there is no systematic order to the table, searching must be done using the Serial method.

U User Ordered

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.

S Sequential

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

D Descending

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

H Hash

In a hash table, new rows are indexed by an integer that is derived 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 Methods

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	Tree Binary
H	Hash

S Serial

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, this 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 their extra computation requirements.

Q Queued

The queued search method is a variation of the Serial search. Each subsequent search begins where the previous one left off. This method only works 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.

B Binary

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.

C Tree Binary

This search method, also called Address Tree Binary, is a variation of the binary search. The index for halving the table rows used with the binary search is stored rather than recalculated each time a search is done. This method becomes inefficient if the table size changes often and the index tree must be rebuilt frequently.

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

H Hash

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.

Provided that the table does not become too full, or so large that it must be paged, this method of searching can be very fast.

Table Organization/Search Method Combinations

The following table shows which search methods may be used with each type of table organization. The default values are indicated and illegal combinations are indicated by dashes (---).

		Search Method				
		S Serial	**B Binary**	**C Tree Binary**	**Q Queued**	**H Hash**
Organization						
R	Random	Default	------	------	------	------
U	User Ordered	Default	------	------	------	------
S	Sequential	------	Default	OK	OK	------
D	Descending Sequential	------	Default	OK	OK	------
H	Hashed	------	------	------	------	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. For these organizations the user cannot change the default search method.

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 T.

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 stored in a second table. Organization and search method actually apply to this second table. The appearance of the table to the user is identical whether the table is a true table or a pointer table.

Pointer tables require additional space for a second table to hold the pointers and they require additional computation to maintain and use the index. However, if

the table is very large and moving rows requires the movement of large amounts of data this may be a practical solution. Some situations in which pointer tables may be useful are described below.

- In cases where multiple Views are to be created for a data table, the table must be defined as a pointer table. This is necessary since each View for the table is a pointer to the data table
- Using a sequential pointer table may be more efficient when sequence is required for table rows but updates on a true table would require the movement of too much data. Maintaining the sequence in a pointer table only requires the movement of pointers and is therefore more efficient if the actual table rows are very large.
- A hashed pointer table may save considerable storage over a hashed true table. The actual data is stored as a random table that is densely packed. The empty row spaces required for a hashed table are in the index table and are not as costly since they are smaller rows. Another more efficient method for accessing large hash tables is to use a paged table, as described in the following section.

Storage Mode Code

This option controls how a table is to reside in memory. The table may be either R, Resident, in which case it fully resides in memory or P, Paged, in which case only the portion of the table that is required will be brought into memory. The default setting here is Resident, which permits optimum efficiency for most tables. Tables should generally be Paged only when they are very large and there is insufficient memory to justify the table being Resident.

With Paged tables, most accesses will result in an I/O operation to bring the appropriate block or page into memory. As such, Paged tables will only be practical for very large tables.

Paged tables must have the following settings in other fields of the table definition screen:

Organization	Hash
Search	Hash
Row Size	Not more than 1000

The storage mode code will also affect the way that multiple generations are handled. With a Resident table, a complete copy of each generation is kept on the tableBASE library. With a Paged table, only the updated pages are stored to create a new generation. Unchanged pages are stored only once no matter how many generations there are.

Note: The tableBASE mechanisms for automatically increasing the size of tables as necessary, (see expansion factor described below) do not work with Paged tables, nor does the tablesONLINE Table Copy Utility. To expand or copy Paged tables use the TBEXEC utility described in the ***Batch Utilities Manual***.

Row and Key Parameters

The following three parameters are used to allocate storage for rows and to find keys within the rows. If you have defined your table using the D, Define All, option in the Table Definition Menu, these numbers will be generated for you by tablesONLINE based on the information provided in previous screens.

Field Name	Minimum Value	Maximum Value	Default Value
Row Length	1	32,767 (1000 for paged)	1
Key Length	1	Smaller of 256 or Row Size	1
Key Location	1	Row Size - Key Size + 1	1

Warning: 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.

Note: If the row size in your table is large, you may wish to consider using a pointer table. See the section on True vs. Pointer Tables above.

Number of Rows

Field Name	Minimum Value	Maximum Value	Default Value
Number of Rows	1	see below	1

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.

tablesONLINE/CICS always runs with the tableSPACE memory management facility. This manages a portion of the CICS region that has been dedicated to tableBASE tables. This table space is shared by all tableBASE users. Tables are automatically swapped to a tableBASE library, as necessary, to support this sharing.

Resident tables cannot become larger than the available table space. For Paged tables only the page index and the active page must fit in the table space. The effective limit is library size.

Generations to Keep

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

tableBASE allows up to nine generations of a table to be kept on the tableBASE library. For Resident tables, disk space usage increases in proportion to the number of generations kept. Extra generations are less costly for Paged tables since only the pages that are changed are stored to disk in subsequent generations.

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

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

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 if a different View is required.

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 for information purposes.

Date	This field contains the date the table generation was stored.
Time	This field contains the time 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 tablesONLINE 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	If the table is an alternate index, this field contains the name of the data table.
Table Open Status	This field indicates whether the table has been opened for read, write or not at all.
Alternate Index Indicator	This field indicates whether or not an alternate index has been invoked.

Display Order Editing

Once you have completed the field definition 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 of the Table Definition Menu, Edit Display Order. This will display the Identify Table screen. To move to the display order editing screen, you must insert the name of the library that contains the View and the name of the View. When the Identify Table screen is displayed, you will note that default information has been provided in all of these fields. This information is stored and provided by tablesONLINE. It describes the last table you worked on in tablesONLINE. If the information provided is not correct, make any required changes. Press <Enter> to open the following Edit Table screen.

```
tablesONLINE 5.1.0 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     U
     FIRST NAME             Y    14     X                   21    14     U
a    DIVISION               N     8     X                   43     8     U
     DEPARTMENT             N     8     X                   35     8     U
m    SEX                    N     1     X      S            51     1     U
     CHARITABLE DONATION    N     6     2                   52     5     N
     DATE OF CONTRIBUTION   N     8     A      V            57     8     A
```

The Edit Display Order screen displays all of the field names for the table together with the row layout. Each field definition occupies one line on the screen. Use the editor line and block move 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.

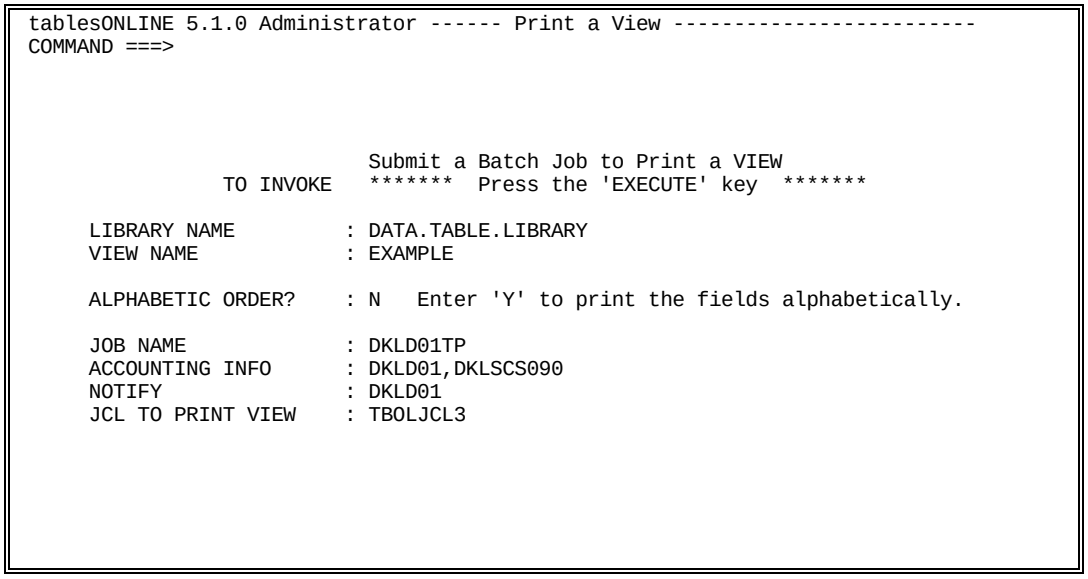
It is also possible, from this screen, to perform some additional editing functions. New fields can be added to the table, using this option, only if they are comment fields. In addition, edits to existing rows may be made, provided they do not alter the layout of the row (i.e., changing the display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

Browse View

This option 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

Print View

By selecting Option P in the Table Definition Menu you will be presented with the following screen. As is indicated, the screen enables you to obtain a hard copy report of a particular view.



Library Name	Insert the name of the library where the view is kept. You may also enter a number from 1 to 6 to select a relative library number, a DDName or a full dataset name. Pressing PF1 (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 copybook. Pressing PF1 (Help) 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.
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.

.

Create Alternate

An alternate index to a data table permits access to the contents of a table using a different table key, organization and/or search method from those originally defined in the physical table attributes. Using this feature, you can create as many different layouts of the data table as you wish without having to generate multiple copies of the data.

Note: If you wish to create an alternate index for a table, the data table must have been defined as a resident pointer table.

Before you can use an alternate index, you must create a View using Table Definition Menu options 1, 2, 4, as previously described or by copying and modifying an existing View.

You can create an alternate index for a data table by selecting option 6, Create Alternate, from the Table Definition Menu. Selecting this option will display the following screen.

```
tablesONLINE 5.1.0 Administrator ------ Create an Alternate Index -------------
COMMAND ===>


                          Create or Edit an Alternative key definition.
               TO INVOKE  *******  Press the 'EXECUTE' key  *******

     LIBRARY NAME         : DATA.TABLE.LIBRARY
     ALTERNATE INDEX NAME : FIRSTNAM

     TABLE NAME           : EXAMPLE

     INDEX ORGANIZATION   : S
     SEARCH METHOD        : B
     KEY LOCATION         : 21
     KEY LENGTH           : 14
```

In this screen you must enter the name of the library where you wish to store the alternate index for the table, the name to be assigned to the alternate index and the name of the table to which the alternate index applies.

Once you have completed these fields, you can specify a table organization, search method, key location and key length that differ from those of the original table. Any or all of these fields may be changed depending on the needs of the user.

Note: Pressing the <End> or <Enter> key on this screen does not create an alternate index. You must press <Execute> to complete the operation.

Attempting to define an alternate index with a name that already exists results in an error message.

Modifying an Existing Alternate Index

To update an existing alternate index in tablesONLINE, open the existing alternate index 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 press <Execute>.

Restructure Table

Once you have modified a view by inserting new fields, expanding or reducing the length of existing fields or deleting existing fields, this option will restructure the data accordingly. Note that, if you have expanded a date field to accommodate the century (i.e., changed a field's format from YY to YYYY), the following rule applies:

If the value of YY is 80 or greater, the field will be converted to 19YY.

If the value of YY is less than 80, 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 displayed below. This is only necessary if the View relates to a data table that already exists.

```
tablesONLINE 5.1.0 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.
```

When this screen is displayed, the library name and table name will already be inserted for you 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, press <Execute> 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. This will open the screen displayed above. 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 <Execute>. 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 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 Menu to display the following screen and complete the fields described below.

```
tablesONLINE 5.1.0 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
```

Library Name	Insert the name of the library where the view is kept. You may also enter a number from 1 to 6 to select a relative library number, a DDName or a full dataset name. Pressing PF1 (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 copybook. Pressing PF1 (Help) in this field will display 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. (e.g., '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 new copybook will be placed.
Member Name	This field contains the name of the member in the COPYLIB that will contain the newly created copybook.
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 (User-ID) that will appear on the job statement of the batch job to create a COBOL copybook for a View.
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.

Define M2M

Using this option, it is possible to define multiple relationships between tables and Views within tablesONLINE. The facility that allows you to do this, and to subsequently make use of these defined relationships, is called M2M (Many to Many). 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 that table object name that will indicate 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 Menu. This will display the following screen.

```
tablesONLINE 5.1.0 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         ===>
```

This screen will already contain the Data Table Library, View Library and Table Object name 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 will display the Row Edit screen for the selected table object. If no Table Object is specified in this field, the following edit table screen will be displayed.

```
tablesONLINE 5.1.0 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
```

From this screen 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 <Execute> 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.

Note: 1. An entry is not required in the M2M table if the relationship between the View and the Data table is 1 to 1, and the View and Data share the same name. This is the most common use of tablesONLINE.

2. Nor do we need entries in the M2M table to describe many Views for 1 Data table. 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 (default), the name entered in the Table Object field is resolved as follows:

Step 1 The M2M table is searched. If an entry on this table is found that matches the entered table object name, then the actual table and view names are obtained from this entry. Otherwise,

Step 2 The available tableBASE libraries are searched for a View with the same name as the entered table object name.

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

Step 4 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. Otherwise,

Step 5 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. However,

Step 1 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.

Step 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.

Chapter 5
Utilities

tablesONLINE 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 following screen:

```
tablesONLINE 5.1.0 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.
```

Each of these options is described in detail on the following pages. Before proceeding, however, there are some points you should be aware of.

- 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. Whether tables have passwords is a decision made by the system administrator at your site.
- Several of the utility screens have a library name field. You are authorized to operate only on libraries that are included in your library list. This list is set up in 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 (Help). If you do not know the name of the library where a specific table or View resides, inserting the table or View name in the appropriate field, clearing the library field and pressing <Enter> will 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:
    - You can insert a DDNAME
    - You can insert a Dataset Name
    - You can specify a library alias
    - You can specify a number from 1 to 6.

  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 for libraries.

  Or, you can identify a table in a library name field by inserting a number from 1 to 6 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 six 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 following screen.

```
tablesONLINE 5.1.0 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-DISSAPEAR-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
```

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 it in the password field. Specify the table generation to print; by default, this is the current generation.

If you wish to print a particular view of the table, 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 Number 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 fieldname(s) to be

selected from the View for printing. The default is to print all fields. Use double quotes to enclose fieldnames that contain embedded blanks.

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

Fieldnames may be selected "generically" by placing an asterisk '*****' after the significant characters of the fieldname. All fields beginning with the significant characters will be printed. An asterisk '*****' by itself will select all fields

Print all fields beginning with D.

Example: FIELDS=D*

Print all fields and two again.

Example: FIELDS=*,KEY1,KEY2

The first occurrence of a generic fieldname 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 Number 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 table or view by specifying starting and ending key fields. If you specify a start key and no end key, all rows beginning with the starting key will be 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:.

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 (User-ID) 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. Please note that this copy utility may not be used to copy paged tables. If you wish to copy a paged table, you must use the batch facility TBEXEC.

To copy a table, select 1, Copy Table, from the Utilities Menu. This will display the following screen.

```
tablesONLINE 5.1.0 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       : TESTMENU  Enter new name if copy to same library.


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

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 it 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 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 on 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 the table is password protected.

Note: If you have inserted a table name and relied on tablesONLINE to insert the proper library name in the library field, you should press <Enter> before pressing <Execute>. This will allow 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 <Execute> or PF2.

Copy View

The Copy View utility allows you to copy an existing View within the same library or to another library.

To copy a View, select 2, Copy View, from the Utilities Menu. This will display the following screen.

```
tablesONLINE 5.1.0 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)    :
```

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 it in the password field. To copy the View within a library, leave the new library name 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 the View is password protected.

Note: If you have inserted a View name and relied on tablesONLINE to insert the proper library name in the library field, you should press <Enter> before pressing <Execute>. This will allow 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 <Execute> or PF2.

You can use this utility to copy the View for a paged table. However, as indicated above the associated data table cannot be copied with tablesONLINE. It must be copied using the batch utility TBEXEC.

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 3, Delete Table, from the Utilities Menu. This will display the following screen.

```
tablesONLINE 5.1.0 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
```

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 it 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 <Execute> or PF2 to delete the specified table generation.

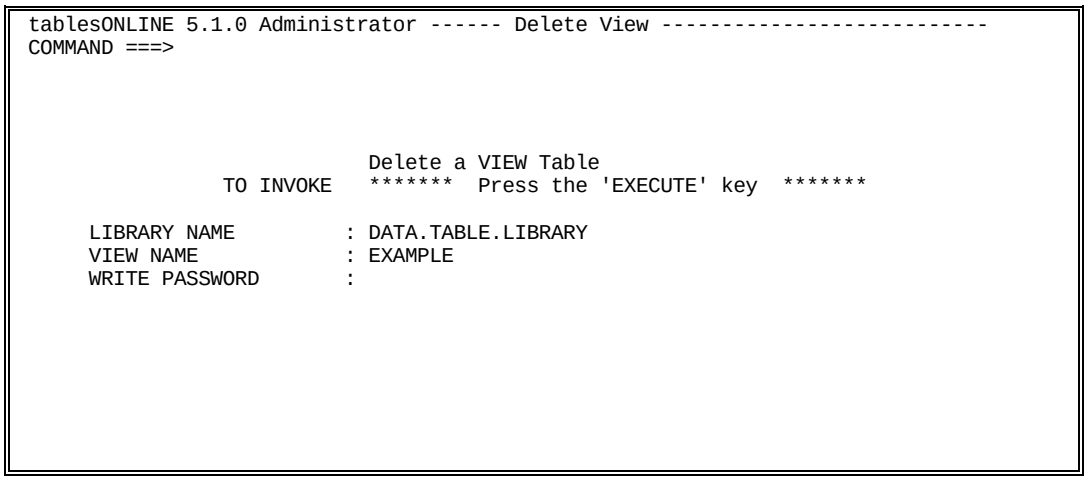
If you wish to delete more than one generation of a table, successive presses of <Execute> will delete generations, one for each use of <Execute>, until no further generations exist.

Note: If you have inserted a table name and relied on tablesONLINE to insert the proper library name in the library field, you should press <Enter> before pressing <Execute>. This will allow 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 4, Delete View, from the Utilities Menu. This will display the following screen.



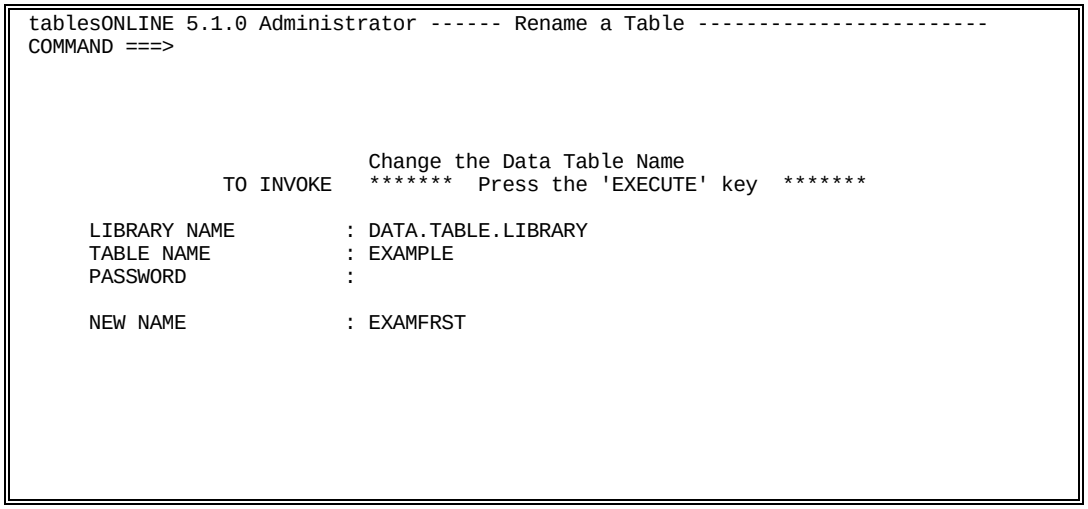
Insert the name of the View you wish to delete and the name of the library in which it resides. If the View is protected by a password, you must insert it in the password field.

Once you have completed all of the necessary information in this screen, press <Execute> or 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 field, you should press <Enter> before pressing <Execute>. This will allow 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 5, Rename Table, from the Utilities Menu. This will display the following screen.

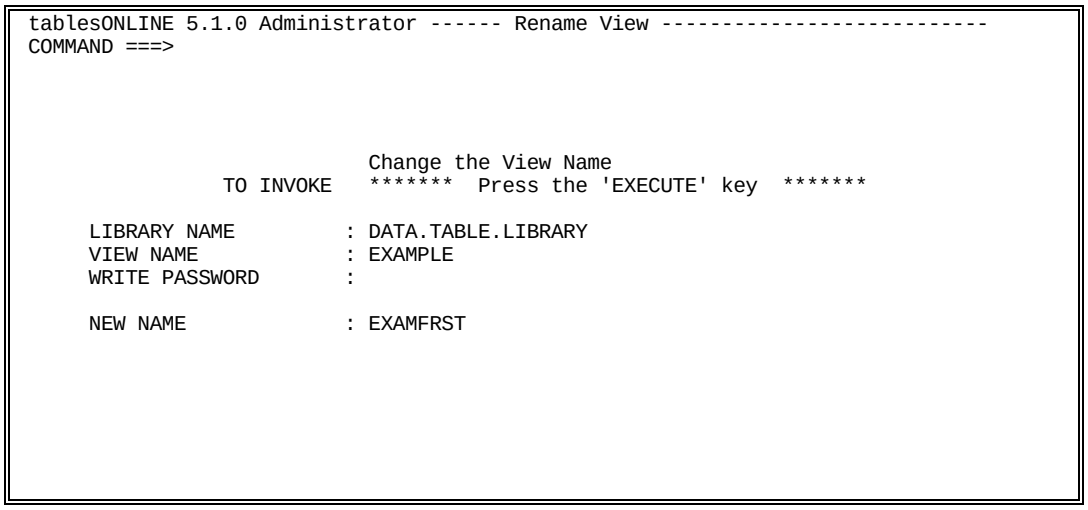


Insert the name of the table you wish to rename and the name of the library in which it resides. If the table is protected by a password, you must insert it in the password field. Insert the new name that you wish to assign to the table.

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

Rename View

The Rename View utility allows you to rename a View. To rename a View, select 5, Rename View, from the Utilities Menu. This will display the following 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 it 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 <Execute> or PF2 to rename the specified View.

Change Passwords

The Change Password utility allows you to change the read and/or the write password for a data table. To change the password for a table, select 7, Change Password, from the Utilities Menu. This will display the following screen.

```
tablesONLINE 5.1.0 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   :
```

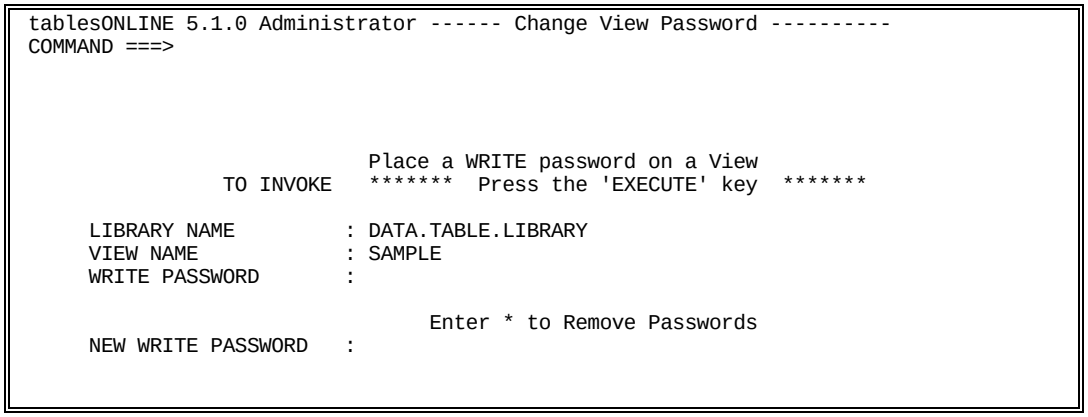
Insert the name of the table you wish to change the password for and the name of the library in which it resides. If the table is already protected by a write password, you must insert it in the write password field. Insert the new read password or the new write password, or both, in the appropriate 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 <Execute> or PF2 to change the password.

A new generation of the table will be 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 8, Change Password, from the Utilities Menu. This will display the following 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 it 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 <Execute> or PF2 to assign the write password to the View.

A new generation of the View will be created which can be accessed using the new password.

Edit Profile

The User Profile table contains an entry for each user, specifying 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 9, Edit Profile, from the Utilities Menu. This will display the following screen.

```
tablesONLINE 5.1.0 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
```

Note that, as indicated by the ++++ at the bottom of the screen, the profile table spans more than one screen. You navigate through the screens by using the PF7 and PF8 keys to go backwards and forwards. Once you have completed all of the necessary information on this screens, press <Execute> or PF2 to update the user profile.

Each of the fields contained in these screens is described below. Many of the fields in this table are set automatically by tablesONLINE based on the actions of the user. As such, there is no point served by changing these fields. In the following descriptions, the fields that are automatically set 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 flag is Y and AAA if this flag 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 as well as an identifying code for the message, will appear.

Job Name

(*MVS Only*) 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

(*MVS Only*) 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

(*MVS Only*) This field contains the notify userid that will appear on the job statement of the batch job to create a COBOL copybook for a View.

Copylib Dataset Name

(*MVS Only*) This field contains the COPYLIB dataset name that contains the FD copybooks available to application developers for the compilation of application programs.

Member Name

(*MVS Only*) This field contains the name of the member in the COPYLIB that will contain the copybook.

Field Name Prefix

(*MVS Only*) This field contains a prefix that will be placed in front of all field names in the COBOL FD. (e.g., 'WS-TBXX-')

Include Command Area

(*MVS Only*) 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

(*MVS Only*) 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 may be modified by your installation.

JCL To Print Table

(*MVS Only*) 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 TBOLJCL2. It may be modified by your installation.

JCL To Print View

(*MVS Only*) 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 TBOLJCL3. It may be modified by your installation.

Extended Attributes

The balance of the fields on this screen are 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 them
- You are uncertain about the above conditions

If you do wish to make modifications to these attributes, experimentation is the best way to determine what function each of them performs and what effects are possible.

Warning: 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 the installed 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.

Chapter 6
Building Applications

tablesONLINE Architecture

tablesONLINE is both a CICS transaction 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 itself so common data is not duplicated.

Using this type of approach has 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 have a positive impact on reliability and security. Finally, system changes -- whether maintenance, customization, performance tuning, or installation of a new release -- are simplified.

Introduction

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. The table names listed below 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, etc.

Table Name	Description
XXXXMENU	Contains customized menu entries for the application.
XXXXPFKS	Contains customized PF Key assignments for the application.
XXXXCMDS	Contains the command translation information for the application.
XXXXDESC	Contains screen descriptions for the application.
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 these tables. These Views ensure that the system knows how each table is set up. There is a great deal of data validation and error-checking done 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 distribution version of tablesONLINE, two of these functions, copying the application driving tables and creating entries in the Application Control Table, are functions that are reserved for the tableBASE administrator. These functions are documented in the ***Administrator's 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 Programmer's 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 manual we will be using the delivered tablesONLINE driving tables as templates for creating a new application. It is possible that when you create a new application you may be doing so using tables that have already been modified. As such, these tables may have different entries from those we describe in this manual.

It is not possible for us to anticipate the changes that may be made to the tables within your organization and it will therefore be necessary for you to understand the relationships between the menu, help and description table entries and where they appear in your application.

Selecting a Table

Once they have been copied, the application driving tables are selected for edit from the Application Developer's Menu displayed below.

```
tablesONLINE 5.1.0 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.
```

The first five options in this menu have already been discussed. 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 the following:

```
tablesONLINE 5.1.0 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        ===>
```

You will notice that Data Table Library name and the View Library name are displayed in the screen in white. The application driving tables will always be found in the Applications. Driving.Tables library and the View required to edit those tables will be found in the System.Tables library. As a result, these are protected fields that may not be edited.

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, in the category

selected (e.g., menu), place the cursor in the Table Object field and press PF1 (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. 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> again 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 the line commands.

Menu Tables

The first application driving table that we will look at is the Menu table. 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 1 from the Application Developer's Menu, leaving the key field on the table identification screen blank. This will display a screen similar to the following:

```
tablesONLINE 5.1.0 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
s    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    P     PRINT VIEW           Submit a Batch Job to Print a View
++++ DEFINE    1     DEFINE VIEW          Define the Fields in a Table's View
```

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.

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. This table contains one entry for each menu option in the tablesONLINE menus. Once a row is selected, the Edit Row screen appears. All of the fields are displayed below, however, on your terminal they will be spread over two screens. Use the scroll keys to move to the fields that overflow.

```
tablesONLINE 5.1.0 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.  :
```

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, the following fields are used. A brief description of each field is provided.

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 8 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 4 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.
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 a selection has been made and completed. If this field is blank, the menu is redisplayed. 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.

This field may be used to automatically invoke additional menu selections. If you wish to automatically invoke a menu selection, set the display indicator to not display the row to the menu and then insert the select symbol for that row in this field. This will automatically invoke the specified menu row.

If the value 'CICS' is inserted in this field, the user is sent back to the 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. These extended attribute fields 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 how these attributes are used within your environment, it is recommended that you leave these fields as they are delivered.

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 transfers 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 that is 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 following values:

Field Value	Transfer To
T	Table Editor
M	Menu
S	Utility
Blank	Program/Transaction ID

T Table Editor

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, displaying the following fields for the menu row:

```
DYNAMIC VIEW SUFFIX  : T
PROGRAM OP MODE      : B
FUNCTION             : BROWSE
DB TYPE              : TABLE
TABLE NAME           :
TABLE LIBRARY        :
SPECIAL VIEW NAME    :
DUPLICATE KEYS IND.  :
```

Program Opening Mode

This field is used to control which tables 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:

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 fields FUNCTION and DB TYPE are used by the editor as components of keys for look-ups 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 will open 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/View Library

This field is used to specify the View 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.

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 to only 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 ensure that only the necessary fields are displayed to the user.

Duplicate Keys Indicator and Duplicate Keys Allowed?

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, or
- 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? field does not appear on the table identification screen.

Otherwise:

- The Duplicate Keys Allowed? field appears on the table identification screen,
- The menu Duplicate Keys Ind controls the initial setting of that field as follows:

Indicator	Setting
space	N
Y	Y

and the user is free to alter that setting.

M Menu

To create a menu item that transfers control to another menu, insert M in the Dynamic View Suffix field and press <Enter>. This will refresh the screen, displaying the following fields for the menu item:

```
DYNAMIC VIEW SUFFIX  : M|
TRANSFER MENU        : UTILITY |
TRANSFER MENU SYMBOL :     |
```

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. This is 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 <Cancel>.

S Utility

A menu item with a Dynamic View Suffix set to S transfers control to a utility function. The following fields are displayed in this case:

```
DYNAMIC VIEW SUFFIX  : S|
PROGRAM OP MODE      : U
FUNCTION             : DEF TBL
```

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.

The following is a list of the supported 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

This option has no user-set parameters. There are two protected fields displayed for this type of transfer. A program opening mode of **U** indicates this is a utility and the function field indicates which one.

Setting up a menu item that calls a tablesONLINE utility is accomplished by copying an existing item in a menu table, such as TBOLMENU, that calls the required utility function. Use the **N** line command to copy the existing menu item. Once copied, modify the unprotected fields such as Screen Name, Select Symbol, Display Indicator, Short Description and Long Description for your application.

Blank Program/Transaction ID

To create a menu item that transfers control to a program or a CICS transaction, leave the Dynamic View Suffix blank and press <Enter>. This will refresh the screen displaying the following fields for the menu item:

```
DYNAMIC VIEW SUFFIX  :
TRANSFER PROGRAM     : TBDRIVC
RETAIN ENVIRONMENT?  :
TRANSFER TRANSACTION :
TRANSFER PARAMETER   :
TRANSFER PARAMETER   :
```

Transfer Program

Enter the name of the program that is to be invoked by the menu item. 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. 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(2).
    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. 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.

A special menu entry can be included on the menu table to transfer control to a program or transaction only when attempting to terminate tablesONLINE. tablesONLINE is terminated when the command 'CICS' or its alias 'X' is executed, or the <Clear> key is pressed. The terminating transaction or program is activated when tablesONLINE finds a menu entry with a blank screen name and the literal TRSF in the select symbol field. This can be used to transfer to another menu system or to pass control to another tablesONLINE session.

Using TBDRIVC as the terminating program may be useful for diagnosing user exit programming. TBDRIVC uses the standard 20 character DFHCOMMAREA protocol and, when finished, will return control to the point at which it was initiated. See the ***tableBASE Programmer's Manual*** for a full description of TBDRIVC.

To Create a New Menu Item

To create a new menu item, select an existing item to use as a template with the **N** line command. You may now make any required changes to the fields defining the menu item and save the new item.

To Create a New Menu

To create a new menu, simply 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 Administrator's 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 2 from the Application Developer's Menu and press <Enter>. This will refresh the screen displaying a field where you may insert a specific PFK identifier. If this field is left blank, the screen below will be displayed. If you wish to edit the information for a particular PF key, you can insert the PFK identifier for that key and the item will be opened in the Edit Row screen.

```
 tablesONLINE 5.1.0 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
```

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. Thirteen of these keys, PF1 to 12 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 re-assign the keys the distribution version uses. However, keep in mind that, if you alter the pre-assigned PF keys for tablesONLINE, they may 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 3, Edit Cmds Table, from the Application Developer's Menu and press <Enter>. This will refresh the screen displaying a field where you can insert a command alias and command parameter. If this field is left blank, the screen below will be displayed. If you wish to edit the information for a particular command alias, you can insert the command alias and command parameter for the command and that row will be opened in the Edit Row screen.

```
tablesONLINE 5.1.0 Administrator ------ Edit the Application's Alias Commands -
COMMAND ===>

                                                    Row Location : 1

          COMMAND                  COMMAND                  ACTUAL
           ALIAS                    PARM                    COMMAND
     _____________________ _____________ALIAS_____________ _____________________ _
     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
     CS                   YES                              CONFIRMSAV            Y
     DEL                                                   DELETE
     DELE                                                  DELETE
     DELET                                                 DELETE
++++ DO                                                    DOWN
```

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 edit row 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 third line, that entry is translated to CONFIRMDEL N and passed to tablesONLINE.

When a command alias in inserted on the command line, tablesONLINE will search 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 DO blank DOWN blank entry. The command is translated to DOWN 1000, with 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 your users from issuing certain sensitive commands.

For example, to disable the command CICS add a row to the command table with the command alias set to CICS, the actual command field set to ===>CICS and the other fields set to blanks. Then, if users try to use the CICS command, the message

```
TB5912E Invalid Command or option: '===> CICS           ', please reenter
```

will be displayed on their screen.

Note: Disabling the CICS command will also prevent users from exiting from tablesONLINE to native CICS by using the <Clear> key.

Another way of disabling the <Clear> key is to add a row with the command alias set to CICS, the actual command field set to END and the other fields set to blanks. Then, if users try to use leave tablesONLINE by using the <Clear> key, they will simply be returned to the previous screen. If this is desirable, it is also prudent to remove the row with the command alias X, to avoid users accomplishing the same function using this command alias.

Help Tables

It is possible to customize the help available for any tablesONLINE application. This is done by editing the Help table. To edit this table, select 4, Edit Help Tables, from the Application Developer's Menu and press <Enter>. This will refresh 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 will display the screen below.

```
tablesONLINE 5.1.0 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.
```

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 inserting TABLE on the command line and pressing PF1 (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.

The remaining three fields 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.

Help for Menus

Help is available for both menus and tables. When a menu is displayed, and help is requested, the help table from the ACT is searched. The following two searches are performed in sequence.

Key Fields	1	2
	<Menu-ID>	<Short Description>
		<Short Description>

When the cursor lies on a Menu item and Help 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:

'MENU,<short description>'.

If general help is required for the entire menu, only the keyword MENU should be entered on the command line.

Creating an Indirect Help Entry

To create an indirect help entry, first create a new item using the line commands, insert an 'I' in the Indirect Lookup Indicator field, clear the remaining non-key fields and press <Enter>. This will display the following screen that contains several fields (indirect table, indirect key - table and indirect key – field) that are used to define an indirect help entry.

```
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
```

In the Indirect Table field you must specify the name of the help table from which the indirect entry is to be retrieved. The Indirect Key - View field allows you to specify the View 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.

Search 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 searches that are performed for contextual help for data are as follows, in sequence:

Table Name	**Search Done**	**Key Fields**	
		1	**2**
From View	If present	<table name>	<field of item> <field of item>
From ACT	Always	<table name>	<field of item> <field of item>

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 are 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 some help table name must be specified in the ACT 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. This is done by editing the Messages table. To edit this table, select 5, Edit Msgs Table, from the Application Developer's Menu and press <Enter>. This will refresh the screen displaying the key fields where you can insert a message group, message number and message sequence number. If you wish to edit the information for a particular message 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 can select a message item to edit using one of the line commands. This will display the screen below.

```
tablesONLINE 5.1.0 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
```

A tablesONLINE user exit program may return a message code such as TB-5934 to indicate some type of 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.

There are several message groups that exist in the delivered tablesONLINE messages table (TBOLMSGS). These groups include TB- and TBS. These are tableBASE and tablesONLINE message groups and should not be deleted by the user. When creating customized messages for an application, you should simply add to the existing message table, creating a unique message group for your application and creating new messages within that group.

As with a help table, it is possible to create a multi-line message. This is done using the message sequence number. Messages are retrieved using the message group and message number as the key and are then displayed in the order indicated by the sequence number.

There are four types of messages that may be created. These types are:

I	Information	**E**	Error
W	Warning	**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 carry on 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 set by the tableBASE Administrator in the User Profile table.

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 are handed additional strings when invoked. The variable offset fields tell the message-handler where to put these additional strings. In the example, a first string will overlay the X's starting at the character position specified by the 1st variable offset, while a second string will overlay the X's starting at the character position specified by the 2nd variable offset. A value of zero for either of these offsets will suppress the substitution of variable data into the message.

Description Tables

tablesONLINE uses the description tables to construct the different screens that make up its components. It is possible to customize these screens or create new screens for a tablesONLINE application. This is done by editing the Description table. To edit this table, select 6, Edit Desc Table, from the Application Developer's Menu and press <Enter>. This will refresh the screen displaying the key fields where you may insert a Program/DBtype, Screen-ID/ViewName, Function and Field Name. If you wish to edit a particular description 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 row from the description table to edit using the line commands. This will display the screen below.

```
tablesONLINE 5.1.0 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
                          :
```

The first four fields in this screen form the key for this table and provide the information that indicates the type of screen to be constructed. The other five fields provide the text that will make up some of the screen display and the display attributes for that text. A detailed description of each of these fields together with information about the values that are valid for each field is provided below. Not all of the information for screen construction is provided by the description table. In the case of a menu screen, information also comes from the menu table, for a help screen from the help table, etc.

Program/DBtype

The value in this first key field determines how the next three are used. The following is a list of the possible values for this field. Each of these values is described in detail later in this section of the Manual.

Value	Used For
ERRORCHK	Displaying messages after the called program returns error status.
HELP	Constructing Help Screens
MENU	Constructing Menu Screens
MSGR	Constructing Message Screens
NO-DESCR	Displaying messages after failed description table lookup.
PFKS	Constructing PF Key Screens
TABLE	Constructing Editor Screens
TBONLINE	Constructing screens that apply to the entire tablesONLINE application.

The following table shows how the three other key fields are used, based on the value in the first key field. Dashes indicate that the field should be left blank for rows with this value in the first key field. Descriptions in angle brackets indicate the type of entry that must be inserted in the field. For example, where <menu> occurs below, the name of an existing tablesONLINE menu must be inserted in that field of the description item.

Program/ DBtype	Screen-ID	Function	Field Name	Text retrieved
ERRORCHK	-----	-----	<error code>	Displayed as part of the error message
HELP	<screen>	<function>	<fieldname>	Defines screen field
MENU	<menu>	-----	<fieldname>	Defines screen field
MSGR	<screen>	<function>	<fieldname>	Defines screen field
NO-DESCR	-----	-----	-----	Displayed after failed lookups
PFKS	<screen>	<function>	<fieldname>	Defines screen field
PFKS	<screen>	<function>	<command>	Displays command description
TABLE	<name>	<function>	<fieldname>	Defines screen field
TBONLINE	<screen>	<function>	<fieldname>	Defines screen field

Search Strategy

The search strategy for these items supports a hierarchical default mechanism. In each case, the series of searches is:

Search Order

1	<Top Key>	<Screen>	<Function>	<Field Name>
2	<Top Key>	<Screen>		<Field Name>
3	<Top Key>			<Field Name>

where the value retrieved by the first successful search is used.

Screen-Id/ViewName, Function, Field Name

The value of the Screen-ID/ViewName, Function and Field Name varies depending on the setting of the first key field. The next section of this guide describes the values used for the remaining key fields and then describes how they are used in the case of each program/DBtype value.

Screen-Id/ViewName Values
The values placed in this field must represent either a screen-ID, menu name, ViewName or the literal 'MENU', depending on which Program/DBtype was specified in the first key field.

Function Values
The values placed in this field always identify a function. The valid values include any existing tablesONLINE functions such as browse and edit or any function defined in the Function field of the application's Menu table. You can define new functions in the Function field of the menu table and then insert description information for that new function in the description table.

Field Name Values
This field allows you to specify where on the screen you wish the description text to appear. Each of the valid values for this field define a particular location on the screen. The valid values are the following:

Value	Where retrieved text will be placed
APPL-ID	Left half of the top row of every screen in the application.
FUNC-ID	Right half of the top row of the screen.
DESC0	Fourth row from the top of the screen.
DESC1	Fifth last row from the bottom of the screen.
DESC2	Fourth last row from the bottom of the screen.
DESC3	Third last row from the bottom of the screen.
DESC4	Second last row from the bottom of the screen.
DESC5	Last row at the bottom of the screen.

Not all of these values are used in each screen type. Each screen type has a slightly different layout and therefore requires only certain fields.

Command Values
This field only applies if the Program/DBtype is PFKS. In these cases, the fourth field can contain any valid tablesONLINE command.

Program/DBtype Values

ERRORCHK

When an error occurs in data conversion at the field level, text retrieved from this type of description table item can be 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, is interpreted as an error code. The error code is the numeric value returned by the data conversion routine. This error code is used as a key to retrieve the appropriate description text when the error occurs and the error code is returned.

HELP

Items in the description table that include HELP as their Program/DBtype are used to describe the display of the help screens. It is possible to define both help and tutorial information here. The standard tablesONLINE help screen appears with the following format:

```
tablesONLINE 5.1.0 Administrator ------ Utility Menu ----------------- HELP
COMMAND ===>
TB-5001I Enter Down command to View additional rows.

******************** TUTORIAL ******************

The following commands are available for this screen:
```

The balance of the text that makes up a help screen comes from the Help Table defined for the application.

If you wish to create an entry that is displayed on all help screens, you must leave the Screen-ID/ViewName and Function key fields blank. The Field Name field may then contain either DESC0-INFO or DESC0-TUTR depending on whether the description text is to appear on all help screens or on all tutorial screens. Both of these values will place the line of description text in the fourth line from the top of the screen as would the DESC0 value.

To define description text for one particular help screen, you must specify the screen-ID. The Screen-ID field may contain the name of any screen that currently exists in tablesONLINE. If you wish to further refine the description by function, complete the Function field. The Function field may contain the name of any function that is currently defined to tablesONLINE. If you wish to create a new function, you may do so by inserting the function name in the

Function field of the menu table and then creating description text entries for that new function.

If the Screen-ID points to a menu name then the Function field simply contains the literal 'MENU'. The description on the help screens in tablesONLINE is determined by the values that may be used in the Field Name field. In the case of a help screen, the valid values are DESC0-INFO, DESC0-TUTR or FUNC-ID.

MENU

Items in the description table that include MENU as their Program/DBtype are used to describe the display of the menu screens. The standard tablesONLINE menu screen appears with the following format.

```
tablesONLINE 5.1.0 Administrator ------ Utility Menu --------------------------
COMMAND ===>


To select, enter number/symbol on command line:



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.
```

The remaining text in menu screen, the options that make up the menu, comes from the Menu Table defined for the application.

If you wish to create an entry that is displayed on all menu screens, you must leave the Screen-ID/ViewName and Function key fields blank. The Field Name field may then contain any or all of the following values: DESC0, DESC1, DESC2, DESC3, DESC4, DESC5.

To define menu text for one particular menu, you must complete the Screen-ID field. The Screen-ID in this case must be a tablesONLINE menu name. This menu name may be any of the delivered tablesONLINE menus or it may be a new menu that you have created in the Menu table for your application. The Function field is not necessary for this type of description text and is left blank.

The description on the menu screens in tablesONLINE is determined by the values that may be used in the Field Name field. In the case of a menu screen, the valid values are the DESC0 through DESC5 values or FUNC-ID.

MSGR

Items in the description table with MSGR as their Program/DBtype are used to describe the display of the message screens. Entries may be defined so that they appear on all message screens, on all screens of one type or on all message screens for a particular function. The standard tablesONLINE message screen appears with the following format.

```
tablesONLINE 5.1.0 Administrator ------ Edit/Browse ----------------- MESSAGES
```

```
COMMAND ===>

Messages generated processing the table -- Asterisks *** indicate data errors:
```

The balance of the text that makes up a message screen comes from the Messages Table defined for the application.

If you wish to create an entry that is displayed on all message screens, you must leave the Screen-ID/ViewName and Function key fields blank. The Field Name field may then contain DESC0.

To define description text for one particular messages screen, you must specify the Screen-ID. The Screen-ID field may contain the name of any screen that currently exists in tablesONLINE. If you wish to further refine the description by function, you can complete the Function field. The Function field may contain the name of any function that is currently defined to tablesONLINE. If you wish to create a new function, you may do so by inserting the function name in the Function field of the menu table and then creating description text entries for that new function.

The description on the messages screens in tablesONLINE is determined by the values that may be used in the Field Name field. In this case, the valid values are DESC0 or FUNC-ID.

NO-DESCR

Whenever a description table lookup fails, text from this 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.

PFKS

Rows in the description table with PFKS as their Program/DBtype are used to define the display of the PF key screens. Entries may be defined so that they appear on all PF key screens, or on all PF Key screens for a particular function. The standard tablesONLINE PF key screen appears with the following format.

```
tablesONLINE 5.1.0 Administrator ------ Edit Rows --------------- PFKS/COMMANDS
COMMAND ===>


Program Function Key Assignments:

PF 1 HELP                - If errors, display messages else help/tutorial
PF 2 EXECUTE             - If changes have been made the row is updated
PF 3 END                 - Return to Previous Screen (If changes,row updated)
PF 4 GETCNT              - Get Row by Count (#, -#, +#, default next row)
PF 5 FINDFLD             - Find Next (First) Fieldname
PF 6 GETKEY              - Fetch Row by Key
PF 7 UP                  - n Fields (Default = 18 or cursor position)
PF 8 DOWN                - n Fields (Default = 18 or cursor postion)
PF 9 WINDOW              - Next Window (- for previous. NEW to create one)
PF10
PF11
PF12 CANCEL              - Cancel changes to row and return


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.
```

If you wish to create an entry that is displayed on all PF key screens, you must leave the Screen-ID/ViewName and Function key fields blank. The Field Name field may then contain any of the following values: DESC0, DESC1, DESC2, DESC3, DESC4, DESC5.

To define description text for one particular PF keys screen, you must specify the Screen-ID. The Screen-ID field may contain the name of any screen that currently exists in tablesONLINE. If you wish to further refine the description by function, you can complete the Function field. The Function field may contain the name of any function that is currently defined to tablesONLINE. If you wish to create a new function, you may do so by inserting the function name in the Function field of the menu table and then creating description text entries for that new function.

If the Screen-ID contains the literal 'MENU', the PF Key screen applies to a menu. In this case, the Function field is left blank since PF key operation is the same for all menus.

The content of the PF Key screens in tablesONLINE is determined by the values that may be used in the Field Name field. In this case, the valid values are DESC0 through DESC5, FUNC-ID or any command name that is defined on the XXXXPFKS table (i.e., CANCEL or END).

TABLE

Rows in the description table that have TABLE as their Program/DBtype are used to describe the display of the table/view editing/browsing screens. This includes the screens used to edit the tablesONLINE system tables as well as the utility screens. Entries defined in this manner appear on all table or view editing and browsing screens, or on all editing screens for a particular function. The following Copy Table utility screen is an example of a screen that is constructed using the Program/DBtype with the literal 'TABLE'.

```
tablesONLINE 5.1.0 Administrator ------ Copy Table ----------------------------
COMMAND ===>



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

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

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

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

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

In addition to allowing you to define the lines of descriptive text and the screen title, this Program/DBtype also allows you to define the grids required to create and edit lines of text.

To define description text for a specific table, the View Name must contain the View of the table being edited or browsed. If you wish to further refine the descriptions by function, you may specify the Function field. The Function field may contain the name of any table editing function that is currently defined to tablesONLINE. If you wish to create a new function, you may do so by inserting the function name in the Function field of the menu table and then creating description text entries for that new function.

The content of the table/view editing and browsing screens is largely determined by the values of the table that is being edited. However, descriptive information can be associated on a field by field basis by entering the Field Name of the View used in editing the table in the Field Name of the field. This will cause descriptions to be placed on the Edit Row screen as described in Chapter 4, Defining Views. The only values that may be used for Field Name are FUNC-ID or a Field Name in a View.

TBONLINE

Rows in the description table that have TBONLINE as their Program/DBtype are used to specify the following:

1. The Application ID which appears in the left half of the first line on every screen.

   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 4 character application name that is currently defined to tablesONLINE. If you wish to create a new function, you may do so by

inserting the function name in the Function field of the menu table and then creating description text entries for that new function.

2. The Function ID that appears in the right half of the first line on all three of the table editing/browsing screens: Identify, Edit-Tbl and Edit-Row. The Function ID on these table editing screens may be overridden with the TABLE entries described above.

   To define description text for one particular help screen, you must specify the screen-ID. The Screen-ID field must contain one of the following literals 'EDIT-TBL', 'EDIT-ROW' or 'IDENTIFY'. If you wish to further refine the description by function, complete the function field. The Function field may contain the name of any function that is currently defined to tablesONLINE. If you wish to create a new function, insert the function name in the Function field of the menu table and then create description text entries for that new function.

3. 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'.

4. All special descriptions that are used on all the table editing/browsing screens.

The Field Names that may be used with this Program/DBtype are as follows:

DESCn-xxx

where n is an integer between zero and seven, essentially a field number, and xxx is a three-letter suffix indicating what the field does. For example:

DESC7-DUP

identifies the "Duplicate Keys Allowed?" prompt given on some Identify Table/Row screens. The following is a complete list of these special field names.

DESC0	Identify screen - table entry instruction
DESC0-BRM	Browse mode
DESC0-KPM	Key protect mode
DESC0-NWM	New row mode
DESC0-UPM	Update mode
DESC0-ROW	Row/field location
DESC0-TBL	Table/View name
DESC1-LIB	Data table library
DESC2-LIB	View library
DESC3-OBJ	Table object name
DESC3-TBL	View name
DESC4-TBL	Data table
DESC5-PSW	Password
DESC6-PSC	Read/Write label
DESC7-GEN	Generation number
DESC8	Identify screen - row/key entry instruction
DESC9-DUP	Duplicate keys

You may wish to browse through the tablesONLINE description table (TBOLDESC) to see how these fields are used.

Displaying Description Text

The balance of the fields on this screen are used to specify description text and how it is displayed. The first field, display length, indicates the total length of the description text. The next field, display offset, indicates the location where the first character of the text is to be positioned on the line.

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 in this screen is the Description Data field. This is where you insert the text that will be displayed in the location you have specified using the fieldname, display length and display offset.

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 by a user more easily than the library dataset name. These aliases are stored in the Library Table. To edit this table, select 7, Edit Libr Table, from the Application Developer's Menu and press <Enter>. This will refresh the screen, displaying a field where you can insert an alias. If this field is left blank, the screen below will be displayed. If you wish to edit the information for a particular command alias, insert the command alias and command parameter for the command and that row will be opened in the Edit Row screen.

```
tablesONLINE 5.1.0 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
```

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 (TBOLACT) must have at least one corresponding entry in the TBOLLIBR table. If this is not done, a warning message, TB-5719, results.

Chapter 7 tablesONLINE/CICS Messages

TB-2000	**I**	The program TBDRIVC terminated successfully.
TB-2010	**E**	Transfer Error: Transfer transaction TRIN was not found.
TB-2011	**E**	Transfer Error: Invalid transaction.
TB-2012	**E**	Transfer Error: User not authorized to execute the transaction.
TB-2013	**E**	Transfer Error: Transaction is currently disabled.
TB-2014	**E**	Transfer Error: TWA size too small.
TB-2015	**E**	Transfer Error: Cannot retrieve PCT entry (Lock/Busy condition).
TB-2016	**E**	Transfer Error: Target program not found or is currently disabled.
TB-5000	**E**	Unexpected tableBASE error xxxx occurred while processing 'xx'.
TB-5001	**I**	Enter Down command to view additional rows.
TB-5002	**I**	Enter Up command to view additional rows.
TB-5003	**W**	Enter Help command for a display of # lines of messages.
TB-5004	**I**	Enter Up or Down command to view additional rows.
TB-5005	**W**	You only have the authority to browse. Edit capability disabled.
TB-5006	**W**	You do not have the authorization to perform this function.
TB-5007	**E**	Indirect help could not be done because table could not be opened.
TB-5008	**E**	Indirect help could not be done because key could not be found.
TB-5009	**E**	Command 'XXXXXXXXXXXXXXXXXXXX' is not supported for hash organization.
TB-5010	**E**	Duplicate keys allowed accepts only a X or X response
TB-5011	**E**	The generation number supplied is incorrect.
TB-5012	**E**	Library: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx not on XXXXXXXX table. Contact your system administrator.

TB-5013	**E**	Library selection failed. You are not authorized to modify 'xxxxxxxx'. Please contact the tableBASE system administrator.
TB-5014	**E**	Table open failed, you may not access tableBASE library:
TB-5015	**E**	Indirect help not followed. Indirect points to indirect help.
TB-5017	**W**	There are no other windows established, enter 'NEW' to create another.
TB-5020	**W**	The last field has been searched. To repeat from start press FINDFLD
TB-5021	**W**	The field name starting with ' ' is not found.
TB-5023	**E**	View key field only accepts an 'X' or 'X' entry.
TB-5029	**I**	Changes to the view 'xxxxxxxx' have been cancelled.
TB-5030	**E**	The count entered is outside the table. Count of last row is
TB-5031	**I**	Changes to table 'xxxxxxxx' have been cancelled.
TB-5032	**I**	The previous row examined has been deleted.
TB-5033	**I**	The row has been added to the table.
TB-5034	**I**	A row with this key is already on the table. The row has been added.
TB-5035	**I**	The row has been updated.
TB-5036	**I**	A row with this key is already on the table. The row has been updated.
TB-5037	**W**	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.
TB-5040	**E**	The hook program '********' could not be loaded.
TB-5041	**W**	The count entered is outside the table. You are positioned at the top.
TB-5042	**W**	The first available row is:
TB-5043	**W**	The last available row is:
TB-5044	**I**	Table already in use. Multiple user access in effect.
TB-5045	**W**	This row is currently in use by 'xxxxxxxxxxxxxxxxxxxxxxxx'.
TB-5046	**I**	Table accessed by multiple users has been closed.
TB-5047	**I**	Multiple user access in effect. Other users may be browsing.

TB-5049 **E** Cannot close table 'xxxxxxxx'. The table is open for write.

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'.

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 cancelling

Enter 'RESHOW'.

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'.

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'.

TB-5055 **I** Table 'XXXXXXXX' has been updated.

TB-5056 **I** View 'XXXXXXXX' has been updated.

TB-5057 **W** 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, you will have to edit the table again.

In order to save your changes and any changes made by other users:

Select 'ENTER' or 'END'.

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'.

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'.

TB-5060 **W** You are not authorized to use the M2M command. See tableBASE support.

TB-5061 **W** A range of 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'.

TB-5062 **E** Listing the contents of the VTS attached to xxxxxxxx is not supported.

TB-5063 **E** Internal error '9999' creating a directory listing.

TB-5500 **W** Exit program called with invalid indicators:

TB-5501	**W**	Field name is spaces.
TB-5502	**W**	Key must be Y, N or blank. If the field is a dynamic suffix location, code S. If it is also a key field, code B.
TB-5503	**E**	Invalid display or field format.
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.
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.
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.
TB-5510	**E**	Invalid feature, must be D, N, Y, ':' or blank.
TB-5511	**E**	Invalid action code. It must be Y, N, R, E, I, D, C, U, or blank.
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.
TB-5514	**E**	Invalid timing, must be B for before system action or A for after.
TB-5515	**E**	Key fields must be contiguous. Check field ====>
TB-5516	**E**	Display length must equal field length.
TB-5517	**W**	Display length should equal field length or field length + 1.
TB-5518	**W**	Display length should equal (2 * field length). Subtract 1 if no sign.
TB-5519	**E**	Display length must equal 2 * field length.
TB-5520	**W**	Display length should equal field length + 1 or field length + 2.
TB-5521	**W**	Display length should equal (2 * field length). Add 1 if signed.
TB-5522	**W**	Exit program entered without setting exit indicators.
TB-5523	**E**	You may not move a field to a position before a key field.
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:
TB-5531	**E**	The data table is currently in use.
TB-5532	**E**	Data table 'xxxxxxxx' requires RSZ, KSZ, KLOC: nnnnnnnnnnnnnnnnnn. This view points to a data table that does not match its definition. Row size, key size & location must be changed to conform to the view.
TB-5533	**E**	Data table 'xxxxxxxx' is in use - try again later.
TB-5534	**E**	You cannot change the keys or suffix in display mode.
TB-5535	**E**	You cannot change fields to be key fields while in display mode.
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.
TB-5539	**E**	The display mask must have 'X' decimal places.
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.
TB-5542	**E**	Invalid decimal point encountered at position 'XX'.
TB-5543	**W**	Edit pattern only allowed for display format of 'X'.
TB-5544	**E**	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.
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
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.
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.
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.
TB-5562	**E**	tableBASE error 'XXXX' occurred creating internal table 'RSXXXXXX'.
TB-5563	**E**	tableBASE error 'XXXX' occurred storing internal table 'RSXXXXXX'.
TB-5564	**E**	The total length of all fields: 99999 exceeds 32,763 bytes.
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.
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-5600	**E**	Exit called for a table it doesn't know about. Table is xxxxxxxx
TB-5601	**E**	You cannot delete your own session. Action ignored.

TB-5700	**W**	Utility processing is bypassed as the current operating mode is 'X'.
TB-5701	**W**	Invalid action. Use S to select a utility.
TB-5702	**W**	Table 'xxxxxxxx' has been copied as 'xxxxxxxx'.
TB-5703	**W**	Table 'xxxxxxxx' has been renamed to 'xxxxxxxx'.
TB-5704	**W**	Table 'xxxxxxxx' has been deleted.
TB-5705	**W**	View 'xxxxxxxx' has been copied as 'xxxxxxxx'.
TB-5706	**W**	View 'xxxxxxxx' has been renamed to 'xxxxxxxx'.
TB-5707	**W**	View 'xxxxxxxx' has been deleted.
TB-5708	**W**	A generation of table 'xxxxxxxx' has been deleted.
TB-5709	**W**	User profile has been updated.
TB-5710	**W**	Application table 'xxxxxxxx' has been copied as 'xxxxxxxx'.
TB-5711	**E**	Function cannot be performed: Table 'xxxxxxxx' not found.
TB-5712	**E**	Utility failed: Table 'xxxxxxxx' is currently in use.
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:
TB-5719	**W**	DDNAME xxxxxxxx not on 'xxxxxxxx' - contact your system administrator.
TB-5720	**E**	Utility failed: Invalid password supplied.
TB-5721	**W**	Table 'xxxxxxxx' has been defined.
TB-5722	**I**	Table 'xxxxxxxx' definition has been updated.
TB-5723	**I**	Supplemental information of view 'XXXXXXXX' has been updated.
TB-5724	**W**	Reducing row size may result in data loss. Press execute to continue.
TB-5725	**E**	The passwords for table 'xxxxxxxx' have been changed.
TB-5726	**W**	Alternate table 'xxxxxxxx' has been defined (for 'xxxxxxxx').
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.
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.
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 table 'xxxxxxxx' has been updated (for '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.
TB-5742	**W**	Utilities function error. There is no processing logic for row ID 'X'.
TB-5743	**E**	Utility failed: Invalid password supplied for target.
TB-5744	**E**	The SYSIN parameter starting with ' xxxxxxxx ....' cannot be found.
TB-5745	**E**	The JES spool data set could not be opened, reason code '9999'.
TB-5746	**E**	The JES spool data set could not be closed, reason code '9999'.
TB-5747	**E**	The JES spool data set could not be written to, reason code '9999'.
TB-5748	**E**	The JCL template table, 'xxxxxxxx' does not exist. Please re-enter.
TB-5749	**E**	The variable ' ' in JCL statement 00000000 was not replaced.

TB-5750	**E**	The JCL statement starting with '//xxxxxxxx ....' cannot be found.
TB-5751	**I**	Job 'xxxxxxxx' has been submitted to generate copybook 'mmmmmmmm'.
TB-5752	**E**	The job that was just submitted for table/view 'tvtvtvtv' has been submitted with an error. Please examine the JES queue and determine whether the job needs to be resubmitted.
TB-5753	**I**	Job 'xxxxxxxx' has been submitted to print table 'tttttttt'.
TB-5754	**E**	Alternate definition failed: Table 'xxxxxxxx' already exists. Use
TB-5755	**E**	Copy cannot be performed: Table 'xxxxxxxx' exists on target library.
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'.
TB-5759	**I**	The alternate definition name 'xxxxxxxx' is unavailable - (not found).
TB-5760	**E**	Enter the name of a table that contains the restructuring rules.
TB-5761	**E**	The name 'xxxxxxxx' used to save the restructuring rules is in use. Re-enter a different name.
TB-5762	**E**	The restructuring table, 'xxxxxxxx' does not exist. Please re-enter.
TB-5763	**E**	Enter Y or N to indicate the generations you want to restructure.
TB-5764	**I**	The restructuring table, XXXXXXXX is not a valid restructuring table.
TB-5765	**E**	Restructuring tbl: XXXXXXXX requires RSZ/KSZ/KLOC:nnnnnnnnnnnnnnnnnnn. The data table that has been entered cannot be restructured with this restructuring rules table.
TB-5766	**E**	Generation (-x) of the data has RSZ/KSZ/KLOC: nnnnnnnnnnnnnnnnnnn. 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.

TB-5767	**I**	X generation(s) of the table 'XXXXXXXX' has been restructured.
TB-5768	**I**	The restructuring rules are saved in the table 'xxxxxxxx'.
TB-5769	**E**	If the restructuring rules are to be saved for future use with other tables, enter a restructuring table name.
TB-5770	**I**	The data is already restructured. Use END or CANCEL to exit utility.
TB-5771	**E**	Although you have elected not to restructure the data table(s) at this time, if you want to restructure the data at a later time, it is essential to enter a restructuring table name to save the restructure rules. If this is not a requirement, enter CANCEL to avoid a repeat of this message.
TB-5772	**E**	FLD: xxxxxxxxxxxxxxxxxxxx Error:
TB-5772	**E**	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.
TB-5774	**E**	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.
TB-5775	**E**	The variable ' ' in SYSIN statement 00000000 was not replaced.
TB-5776	**E**	tablesONLINE does not support copying a Paged table.
TB-5801	**E**	## - Action Error:
TB-5802	**E**	## - Import Error:
TB-5803	**E**	## - Duplication Error:
TB-5804	**E**	## - Mandatory Error: An EOF erase does not constitute a valid key stroke. Spaces are valid.
TB-5805	**E**	## - Pattern: 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.

TB-5806	**E**	## - Verification Error:
TB-5811	**E**	The source view 'XXXXXXXX' not found.
TB-5812	**E**	The source data table 'XXXXXXXX' not found.
TB-5813	**E**	The source table 'XXXXXXXX' is password protected.
TB-5814	**E**	tableBASE error 'XXXX' occurred opening source table 'XXXXXXXX'.
TB-5815	**E**	Source table, 'XXXXXXXX' must be in descending order for 'X' action.
TB-5816	**E**	The length of field 'XXXXXXXXXXXXXXXXXXXX' and source table 'XXXXXXXX' key field length do not match.
TB-5817	**E**	Source field 'XXXXXXXXXXXXXXXXXXXX' not found in table 'XXXXXXXX'.
TB-5818	**E**	Source field 'XXXXXXXXXXXXXXXXXXXX' in table 'XXXXXXXX' is not a key.
TB-5819	**E**	Source field 'XXXXXXXXXXXXXXXXXXXX' in source view 'XXXXXXXX' has a different length than the field defined in the view.
TB-5820	**E**	Source field 'XXXXXXXXXXXXXXXXXXXX' in source view 'XXXXXXXX' is not the entire key of the corresponding source data table.
TB-5821	**E**	Source field 'XXXXXXXXXXXXXXXXXXXX' in source view 'XXXXXXXX' has a different format than the field defined in the view.
TB-5822	**E**	tableBASE error 'XXXX' occurred creating internal table 'SPXXXXXX'.
TB-5823	**E**	tableBASE error 'XXXX' occurred emptying internal table 'ECXXXXXX'.
TB-5824	**E**	tableBASE error 'XXXX' occurred creating internal table 'ECXXXXXX'.
TB-5825	**E**	tableBASE error 'XXXX' occurred deleting internal table 'ECXXXXXX'.
TB-5826	**E**	tableBASE error 'XXXX' occurred deleting internal table 'SPXXXXXX'.
TB-5827	**E**	Upper range 'XXXXXXXXXXXXXXXXXXXX' in source view 'XXXXXXXX' has a different length than the lower range field.

TB-5828	**E**	Upper range 'XXXXXXXXXXXXXXXXXXX' in source view 'XXXXXXXX' has a different format than the lower range field.
TB-5829	**E**	Lower range 'XXXXXXXXXXXXXXXXXXX' in source view 'XXXXXXXX' has no corresponding upper range field for action R.
TB-5830	**E**	Internal get storage failed for data importation processing.
TB-5832	**E**	View data importation pointers corrupted.
TB-5833	**E**	View data duplication pointers corrupted.
TB-5834	**E**	Upper range in source data table 'XXXXXXXX' is too short or missing.
TB-5900	**W**	Help message unavailable, message 'XXXXXXX' not on table 'XXXXXXXX'
TB-5901	**E**	Unable to open table 'XXXXXXXX', reason code 9999.
TB-5902	**E**	Unable to invoke alternate table 'XXXXXXXX', reason code 9999.
TB-5903	**W**	Help information for table 'xxxxxxxx' is not found on table 'xxxxxxxx'
TB-5904	**W**	DESC: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' not on 'XXXXXXXX'
TB-5905	**A**	Error creating table 'XXXXXXXX', reason code 9999. Job aborted.
TB-5906	**W**	Tutorial data 'XXXXXXXXXXXXXXXXXXX' 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'
TB-5909	**W**	Help data 'XXXXXXXXXXXXXXXXXXX' not on table 'XXXXXXXX'
TB-5910	**E**	Transfer program 'XXXXXXXX' of menu option 'XXXX' not found/disabled.
TB-5911	**E**	Blank transfer parameters for menu option 'XXXX'
TB-5912	**E**	Invalid command or option: 'XXXXXXXXXXXXXXXXXXX', please re-enter.
TB-5913	**E**	Transfer option 'XXXXXXXXXXXX' is not defined on menu table 'XXXXXXXX'
TB-5914	**W**	The value entered for this command should be numeric.

TB-5915	**W**	Additional tutorial information is not found on table 'xxxxxxxx'.
TB-5916	**E**	The keys are protected. The row cannot be deleted in this mode.
TB-5917	**E**	Command or option: 'XXXXXXXXXXXXXXXXXXXX', not supported for browse.
TB-5918	**E**	Command or option: 'XXXXXXXXXXXXXXXXXXXX', unsupported for hash tables
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.
TB-5923	**W**	Dynamic open of view 'XXXXXXXX' failed, reason code 9999.
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.
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:
TB-5931	**E**	## - Conversion Error:
TB-5932	**E**	Key field ## - Conversion Error:
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 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'.
TB-5935	**E**	Data table 'XXXXXXXX' is in use by another application. Try browse.

TB-5936	**E**	View 'XXXXXXXX' cannot be stored, library 'XXXXXX X' 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'.
TB-5937	**E**	View defines row size of 00000 which differs from table RSZ of 00000
TB-5938	**E**	View defines key location 00000 which differs from tbl location 00000
TB-5939	**E**	View defines key size of 0000 which differs from table key of 0000.
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.
TB-5944	**E**	View 'XXXXXXXX' cannot be found. Enter NEW command to create view.
TB-5945	**W**	The table in memory is not the latest generation
TB-5946	**E**	The table in memory is not the latest generation. Update access denied
TB-5947	**E**	The table is in use by another application. Changes can't be cancelled
TB-5948	**W**	Help information for menu 'xxxxxxxx' is not found on table 'xxxxxxxx'
TB-5949	**E**	Open of alternate not allowed. Data table/other alternate already open
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.
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)
TB-5961	**I**	Block command 'XXXX' at row count ' ' requires target A or B.
TB-5962	**I**	Block command 'XXXX' at row count ' ' is incomplete.
TB-5963	**E**	Invalid line command 'XXXX', please re-enter.
TB-5964	**E**	A sequential table 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.
TB-5965	**I**	The number of rows deleted is
TB-5966	**I**	The number of rows moved is to location
TB-5967	**I**	Line commands have been cleared.
TB-5968	**W**	Line commands following the selected row have been deleted.
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.
TB-5972	**E**	Multiple line commands are not supported for hash tables.
TB-5973	**E**	tableBASE error 'XXXX' occurred storing table/view 'XXXXXXXX'.
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.
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.
TB-5986	**A**	Table TBOLACT must be upgraded to Rel 5.0. See Installation Guide.

TB-5987	**A**	Table TBOLPROF must be upgraded to Rel 5.0. See Installation Guide.
TB-5988	**A**	Table TBOLCNST does not contain the default row, SIGNON terminated.
TB-5989	**A**	Table TBOLMRO does not contain the default row, SIGNON terminated.
TB-5990	**A**	An error occurred when attempting to terminate your session.
TB-5991	**A**	Unable to open table 'XXXXXXXX', reason code 9999. Job Aborted.
TB-5992	**A**	Transaction work area 99999999 too small. It should be 99999999. Abort
TB-5993	**A**	Your application is not defined to the system, SIGNON terminated.
TB-5994	**A**	Descript. '****NO DESCRIPTION ' not on tbl 'XXXXXXXX'
TB-5995	**A**	Error creating table 'XXXXXXXX', reason code 9999. Job aborted.
TB-5996	**A**	tablesONLINE incomplete environment, call Data Kinetics (613) 523-5588
TB-5997	**A**	Message '5900 W Help message unavailable, ...' not on tbl 'XXXXXXXX'
TB-5998	**W**	Unable to save user profile. tBASE COND:
TB-5999	**A**	CICS error: command code 'XX' response code 'XXXXXX'. Job aborted.
TB-9000	**W**	Exit program 'xxxxxxxx' called with invalid indicators: xxxxxxxxxxxx.
TB-9001	**W**	Table xxxxxxxx called exit with indicators: xxxxxxxxxxxx in test mode.
TB-9999	**A**	CICS error: command code 'XX' response code 'XXXXXX'. Job aborted.

Index

A

K

L

M

V

W