tableBASE

Batch Utilities Manual

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 tableBASE Batch Utilities used to define and maintain tables and libraries, report on their contents and generate COBOL copybooks from table definitions.

Who This Manual is For

This manual is intended for:

- Analysts who define tables
- Programmers who maintain and test tables and generate COBOL copybooks
- Systems personnel responsible for maintaining tables and transferring them from test to production.

What You Should Know to Use This Manual

You should be familiar with tableBASE Concepts and Facilities, the tableBASE Programmer's Guide and the batch MVS and/or VSE environments.

Additional tableBASE References

This manual is one of several that describe tableBASE. The others are:

- Concepts and Facilities Manual
- tableBASE Programmer's Guide
- tableBASE Administrator's Guide
- tableBASE Installation Guide
- tablesONLINE/CICS User's Manual
- tablesONLINE/ISPF User's Manual
- tableBASE Workshop Manual

Chapter 1 Introduction

The Batch Utility Programs

There are five batch utility programs supplied as part of tableBASE. These utilities are designed to simplify your task in maintaining your tableBASE environment. The five programs are:

- **TBEXEC**
  This is the primary tableBASE batch utility. It provides basic housekeeping facilities such as initializing new table libraries, defining tables, deleting tables, mass updates to tables, copying tables among libraries, etc. TBEXEC also provides a print facility especially designed for the printing of tables used in testing.
- **TBPRINT**
  This utility produces columnar listings of data tables, based on formatting information contained in the View.
- **TBDEFPRT**
  This utility prints Views and field definitions.
- **TBCOBFD**
  This utility generates COBOL data description copybooks for use in user-written tableBASE applications.
- **TBCOMP**
  This utility compares tables and identifies any differences for your review.

In addition to the five utilities provided by Data Kinetics, other utilities have been made available by tableBASE users. These are described in Appendix A.

How this Manual is Organized

The maintenance functions provided by TBEXEC are described in Chapter 2. Chapter 3 describes the use of TBEXEC; it includes examples of JCL, report samples and lists of error messages. TBPRINT is described in Chapter 4. Chapter 5 describes the use of TBDEFPRT. The fourth utility, TBCOBFD, is described in Chapter 6. TBCOMP is described in Chapter 7.

A note for users of previous releases of tableBASE

Release 5.0 introduces a change in terminology for certain tableBASE facilities. The term ‘Field Definition Table’ or ‘FDT’ is now referred to as ‘View’.

Chapter 2
TBEXEC Commands

TBEXEC is a batch tableBASE utility program. It allows users to perform a variety of maintenance functions against tableBASE libraries and tables.

Introduction

TBEXEC is used to manipulate tableBASE libraries. These libraries reside on external devices and must be initialized by TBEXEC, tablesONLINE/ISPF or a batch program which calls tableBASE. A tableBASE library has a Directory that allows for up to nine generations of each table.

TBEXEC manages the tables that are placed in tableBASE libraries, keeping track of the characteristics of each table, such as the number of rows in the table, the row size and the table organization.

Different functions, such as defining, copying, loading, unloading and clearing a table, are easily performed by TBEXEC, using keyword commands.

TBEXEC also makes available a date-variable Master Password facility which allows access to tableBASE tables on occasions when passwords have been forgotten. Details of the operation of the Master Password facility have been supplied to your tableBASE Administrator under separate cover.

The remainder of this chapter is organized as follows. First, we present a brief summary of each command. An alphabetical listing of the parameters used in the TBEXEC commands follows. The chapter concludes with a detailed definition of each command.

Command Summary

The following is a summary of the commands available in TBEXEC. Subsequent sections provide details of their use.

Commands	Command Description
CHANGE	Change Table characteristics
Option 1:	Change the Definition of a Table
Option 2:	Change an Alternate Table Definition
CLEAR	Clear a Table
COPY	
Option 1:	Copy a Table Generation
Option 2:	Copy a Table (all Generations)
Option 3:	Copy a Library
DEFINE	
Option 1:	Initialize a new tableBASE Library
Option 2:	Define a New Table
Option 3:	Create an Alternate Definition
DELETE	
Option 1:	Delete a Table
Option 2:	Delete all Generations of a Table
EXPAND	Expand a tableBASE Library
EXPORT	Export a tableBASE Library to another tableBASE environment (MVS, VSE, PC)
IMPORT	Import a tableBASE Library from another tableBASE environment (MVS, VSE, PC)
LOAD	Create a new Table Generation from an Input Dataset
PRINT	
Option 1:	List Directory
Option 2:	Print Table Information
Option 3:	Print Table Contents
RENAME	Rename a Table
REORG	Reorganize a Paged Table
SET	Set Default Values for Keywords
UNLOAD	Write a specified Table Generation to an Output Dataset
UPDATE	Update a Table from an Input Dataset

TBEXEC commands are specified in control statements which are read by TBEXEC from a file identified by the DDname 'CNTLCARD'.

A TBEXEC command sequence consists of a command, a list of parameters specified via keyword and value combinations and a semicolon ';'.

Each command has the following appearance:

COMMAND KEYWORD1=VALUE KEYWORD2=VALUE;

'KEYWORD=VALUE' is the representation of a keyword-value combination and may be repeated for each parameter of the command.

The following rules apply:

1. Each command sequence must start on a new line.
2. A command sequence must be terminated by a semicolon (;). Anything following a semicolon is ignored.
3. The command and each keyword-value combination must be followed by at least one blank.
4. Keyword-value combinations may appear in any order.
5. Columns 73 - 80 of a statement are ignored.
6. Blanks may occur freely, provided that they do not break up commands, keywords or value fields.
7. The character set for unquoted fields is limited to the alphanumeric characters: 0 - 9 and A - Z.
8. Values that contain special characters must be enclosed in quotes (').

TBEXEC terminates when end-of-file is reached on the input CNTLCARD file.

Interpreting Command Parameters

Each TBEXEC command has one or more parameters. Some are defaulted, others are optional. The rules for interpreting the various options of the parameters are:

- A slash (/) indicates a choice among alternatives.
- An asterisk (*) is used to set a field to blanks.
- The parentheses '( )' are not coded on the command; they are used to designate optional parameters.

Parameter Descriptions

Most of the commands used in TBEXEC require parameters. An alphabetic listing describing these parameters follows.

ALLGEN
Denotes that all generations of a table are to be processed by the command. When used, the value must be YES.

ALT
The name of the alternate view.

BASENAME
The name of the data table for which this is an alternate index. The name has a maximum length of 8 bytes. If the name contains non-alphanumeric characters, it must be enclosed in quotes (').

BYTCOUNT
The number of bytes to be used from the table row. Default is the row size.

DEF
The name of the table which is to be used.

EST
An estimate of the number of rows that will be loaded into the table the first time it is used. The default value depends on the table organization and row size and is explained in the tableBASE Programmer's Guide.

EXPFACTR
An integer from 1 to 999 representing the number of tenths of a percent in the Expansion Factor. An Expansion Factor of 250, for example, will be treated as 25%. The default is 200. The Expansion Factor is described in the tableBASE Programmer's Guide.

FROM
The DDNAME of the dataset from which data is to be taken.

GEN
A number from -8 to +255 identifying the generation of a table. If this parameter is zero or is omitted, the most recent generation will be assumed; if positive, it refers to an absolute generation number; if negative, to a relative generation number.

INDEX
A one character code with value of 'T' or 'P' to indicate whether the table data is to be maintained as a True table (T) or as a Pointer table (P). The default is 'T'.

'P' is used to gain efficiency when inserting and deleting rows in sequential tables, or to achieve space savings on Hash tables where the size of the data portion of a row is large compared with the eight bytes used by the index.

ISZ
A number from 1 to 32767 that identifies the length of the row. The default value is 1.

KSZ
A number from 1 to 256 that identifies the length of the key. The default value is 1.

KLOC
Identifies the starting position of the key within the row. The default of 1 represents a key starting at the beginning of the row.

LIB
The name of a DD statement which is connected to a tableBASE library. Unless changed by the SET command, the default for LIB is 'MAINLIB'.

LINSIZ
A number from 1 to 100 that represents the number of bytes of table data to be printed per line. If omitted, the default is 100 or the value specified on the most recent SET command.

LOWERDEN
An integer from 1 to 999 representing, in thousandths, the lowest density a hash table may have. A Lower Density of 700, for example, will be treated as 70%. The default is 600. Density is discussed in the tableBASE Programmer's Guide.

MAXGEN
A number from 1 to 9 that represents the number of generations to be kept for the table. The default value is 1.

MTHD
A one character code indicating the search method to be used for the table:

S	-	Serial
Q	-	Queued Sequential
B	-	Binary
C	-	Address-tree Binary
H	-	Hash

The default value is S, i.e., serial.

NEWLIB
The ddname of a new tableBASE Library to be initialized or the target of an EXPAND command.

NEWNAME
A new table name for a copied or renamed table. The name has a maximum length of 8 bytes. If the name contains non-alphanumeric characters, it must be enclosed in quotes (').

ORG
A one character code indicating how the data in a table is to be organized:

R	-	Random
U	-	User Controlled
S	-	Sequential
D	-	Descending Sequential
H	-	Hash

The default value is R.

PRNT
A one character code indicating the format to be used to print the table contents:

C	-	Character
H	-	Hexadecimal
B	-	Both

If omitted, the default is 'C' or the value specified by the most recent SET command.

PSWD
The READ or WRITE PASSWORD used to gain access to a protected table. It will be used in commands where a password is required. If '*' is specified in the SET command, the default will be restored to the normal default value, which is eight blanks. The maximum length of a password is eight bytes.

RECCOUNT
A number used to limit the number of rows loaded to or taken from a table.

REMOVE
The DDNAME of the dataset which contains the keys of table entries to be removed.

REPLACE

A parameter to indicate that an existing table is to be replaced by a table named in the COPY function.

RPSWD
READ PASSWORD, if READ PASSWORD protection is desired.

STARTBYT
A number which indicates the location in the table entry or input record from which data is to be retrieved. The default value is 1.

STARTREC
A number which indicates the record at which printing, loading or unloading is to start. The default value is 1.

SMC
A one character code that denotes the table's STORAGE MODE CODE:

R	-	Resident
P	-	Paged

The default is 'R'.

TBL
Denotes the name of the table to be used for the command. The name has a maximum length of 8 bytes. If the name contains non-alphanumeric characters, it must be enclosed in quotes (').

TO
Indicates the DDNAME of the dataset into which data is to be moved.

TTYPE
This parameter can be used to convert the name of a data table to the name of a View. Views have the first character of their table name as a lower case letter. A table with the name ABC will have a View named aBC. (All upper case letters begin with the binary digits 11; all lower case letters begin with the binary digits 10.)

blank	The table name refers to a data table name.
F	The table name refers to a View name.
A	The table name refers to an Alternate Table.
X	The table name refers to an Alternate View.

When TTYPE is specified with commands using the parameters TBL, ALT and NEWNAME, it will cause each to be converted.

UPPERDEN
An integer from 1 to 999 representing, in thousandths, the highest density a hash table may have. An Upper Density of 850, for example, will be treated as 85%. The default is 800. Density is discussed in the tableBASE Programmer's Guide.

WITH
Indicates the ddname of the dataset from which data is to be retrieved for updating a table.

WPSWD
WRITE PASSWORD, if WRITE PASSWORD protection is desired. If not specified, it will be set to the READ PASSWORD.

CHANGE Command

The CHANGE command modifies elements of a table definition, subject to the constraints specified for all change commands as defined in the description of TBLBASE. (See the tableBASE Programmer's Guide for details on TBLBASE).

Option 1: Change the Definition of a Table

This option is chosen by the presence of the keyword TBL. Changes made to the definition can cause the contents to be physically reorganized. If the table named is an alternate, changes that affect the alternate definition will be made there, other changes which do not affect the alternate definition will be made to the data table.

CHANGE

TBL	=	(table name) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
GEN	=	(generation number)
PSWD	=	(write password)
ORG	=	(R/U/S/D/H)
MTHD	=	(S/Q/B/C/H)
MAXGEN	=	(new max. no. of generations)
INDEX	=	(T/P)
SMC	=	(new storage mode code - R/P)
ISZ	=	(new row size)
KSZ	=	(new key size)
KLOC	=	(new key location)
RPSWD	=	(new read password/*)
WPSWD	=	(new write password/*)
UPPERDEN	=	(new upper density)
LOWERDEN	=	(new lower density)
EXPFACTR	=	(new expansion factor);

Action

If the password matches the write password for the table, the generation indicated will be opened, changed according to the specified keywords and stored as a new generation. Previous generations will be aged according to normal generation practice.

The existing READ and WRITE PASSWORDS can be removed by setting RPSWD=* and WPSWD=*.

Note: The CHANGE of a paged Hash table requires the allocation of SORT work files.

Note: In VSE the specification SORT=YES will enhance performance when changing a table to a paged table. In order to use this facility, the SORT program must be relocatable.

Note: The WRITE PASSWORD of a table cannot be changed if the other keywords cause a reorganization of the table. Changing the Storage Mode Code (SMC) will result in a reorganization as will, for a paged table, changing the row size, key size or key location.

Option 2: Change an Alternate Table Definition

This option is chosen by the presence of the keyword ALT. Only keywords applicable to an alternate definition are allowed to be changed. The definition specified must be an alternate.

CHANGE

ALT	=	(alternate name) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
ORG	=	(R/U/S/D/H)
MTHD	=	(Q/S/B/C/H)
KSZ	=	(new key size)
KLOC	=	(new key location)
BASENAME	=	(new contents table name);

Action

The alternate definition identified will be deleted and a new definition created using the new values specified and retaining the old values for any values not specified.

Use of a CHANGE Command

The following examples illustrate both options of the CHANGE command.

Example 1

Change a table organization from Sequential to Hash. This change will be used to see whether hash searching is more efficient than binary searching for this table. The table name is COUNTIES and it is on the tableBASE library identified by the ddname TESTLIB.

```
CHANGE      LIB=TESTLIB  TBL=COUNTIES
            ORG=H  MTHD=H;
```

Example 2

Change the name of the contents table for the alternate definition DISTRICT. The new contents table name is REGION. The alternate table is on the tableBASE library identified by the ddname MAINLIB.

```
CHANGE      ALT=DISTRICT  BASENAME=REGION;
```

CLEAR Command

The CLEAR command will empty a table of all its rows, resulting in the number of rows in the table being zero.

CLEAR

TBL	=	(table name) **Required keywor**d
TTYPE	=	(table name conversion - blank/F);
LIB	=	(library name)
GEN	=	(generation number)
PSWD	=	(write password);

Action

Creates a new generation with no rows, using the definition of the named generation.

Use of a CLEAR Command

The following example illustrates the use of the CLEAR command.

Example 1

Delete all the rows in the latest generation of table PARTNO on tableBASE Library MAINLIB (default).

CLEAR TBL=PARTNO;

COPY Command

The COPY command copies a table generation, a table or a library.

Note: In VSE the specification SORT=YES will enhance performance when copying a paged table. In order to use this facility, the SORT program must be relocatable.

Option 1: Copy a Table Generation

This option of the COPY command copies one generation of a table from one tableBASE library to another tableBASE library. This option is specified through the presence of the keyword TBL and the absence of the ALLGEN keyword.

COPY

TBL	=	(table name)	**Required keyword**
TO	=	(destination library)	**Required keyword**
TTYPE	=	(table name conversion - V/F);	
LIB	=	(library name)	
GEN	=	(generation number)	
PSWD	=	(read password/write password)	
NEWNAME	=	(table name)	
REPLACE	=	(YES/NO)	
WPSWD	=	(new write password/*);	

Action

If the password of the table to be copied is correct, the specified generation will be copied to the library named by 'destination library'. The copy will not be carried out if there is a table with the same name on the destination library unless the REPLACE parameter has the value of YES. All table characteristics, except for date and time, will be copied to the new table along with the contents. Today's date and time will be entered in the definition of the new table.

The parameter TTYPE is supplied for your use when you are copying a view. If you are copying a view, enter either 'V' or 'F' as the value of TTYPE. This will ensure that the view name is created properly.

Note: The existing WRITE PASSWORD can be removed by setting WPSWD to *.

Option 2: Copy a Table (all generations)

This option of the COPY command copies all generations of a table from one tableBASE library to another tableBASE library. This option is specified through the presence of the keywords TBL and ALLGEN=YES.

COPY

TBL	=	(table name)	**Required keyword**
TO	=	(destination library)	**Required keyword**
TTYPE	=	(table name conversion - blank/F)	
ALLGEN	=	YES	**Required keyword**
LIB	=	(library name)	
PSWD	=	(read password/write password)	
NEWNAME	=	(table name)	
REPLACE	=	(YES/NO)	
WPSWD	=	(new write password/*);	

Note**:** Only the most recent generation of each paged table will be copied.

Note: If the TTYPE keyword is used, it will cause both the table names specified in the TBL and the NEWNAME parameters to be converted.

Action

If the password of the table to be copied is correct, all the generations of the table will be copied to the library named by 'destination library'. The copy will not be carried out if there is a table with the same name on the destination library unless the REPLACE parameter has the value of YES. All table characteristics, except for date and time, will be copied to the new table along with the contents. Today's date and time will be entered in the definition of the new table.

Option 3: Copy a Library

This option of the COPY command copies one tableBASE library to another.

COPY

FROM	=	(library #1)	**Required keyword**
TO	=	(library #2);	**Required keyword**

Action

"Library" is the name of a DD statement that points to the relevant tableBASE library. The contents of tableBASE Library #1 will be copied to tableBASE Library #2. Library #2 must exist or have been initialized by the DEFINE Command. Tables which already exist on Library #2 will not be copied.

Note**:** Only the most recent generation of a paged table will be copied.

Use of COPY command

The following examples illustrate the three options of the COPY command:

Example 1

Copy the latest generation of table name TABLE01 on the tableBASE library MAINLIB (default) and its associated View to the tableBASE library PRODLIB. Replace the existing TABLE01.

```
COPY      TBL=TABLE01 TO=PRODLIB  REPLACE=YES;
COPY      TBL=TABLE01 TTYPE=F TO=PRODLIB REPLACE=YES;
```

Example 2

Create a copy of a table, TABLE05, on the tableBASE library, MAINLIB, so that the copy may be used in testing without changing TABLE05.

```
COPY      TBL=TABLE05  TO=MAINLIB  NEWNAME=TABLE05X;
```

Example 3

Copy all generations of table name TABLE01 on the tableBASE library MAINLIB (default) to the tableBASE library PRODLIB. The copy will not be performed if a table called TABLE01 already exists on PRODLIB.

```
COPY      TBL=TABLE01 TO=PRODLIB ALLGEN=YES;
```

Example 4

Copy the tableBASE library MAINLIB to a library identified by the ddname BACKUP. This must be a tableBASE library and any tables on MAINLIB that already exist on BACKUP will not be copied.

```
COPY      FROM=MAINLIB TO=BACKUP;
```

DEFINE Command

The DEFINE command initializes a new library, defines a new table or defines an alternate view.

Option 1: Initialize a New tableBASE Library

This option is specified through the presence of the keyword NEWLIB.

DEFINE

NEWLIB	=	(library name);	**Required keyword**

Action

The dataset connected to the ddname 'library name' will be initialized as a tableBASE library.

Option 2: Define a New Table

This option is specified through the presence of the keyword TBL. The library in which this table is to be defined must already be defined.

DEFINE

TBL	=	(table name) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
ORG	=	(R/U/S/D/H)
MTHD	=	(S/Q/B/C/H)
MAXGEN	=	(max. no. of generations allowed)
INDEX	=	(T/P)
SMC	=	(storage mode code - R/P)
ISZ	=	(row size)
KSZ	=	(key size)
KLOC	=	(key location)
EST	=	(estimated number of rows)
RPSWD	=	(read password)
WPSWD	=	(write password)
UPPERDEN	=	(upper density)
LOWERDEN	=	(lower density)
EXPFACTR	=	(expansion factor);

Action

A table is defined on the named library.

Option 3: Create an Alternate Definition

This option is specified by the presence of the keyword ALT and the absence of the keywords NEWLIB and TBL.

DEFINE

ALT	=	(table name)	**Required keyword**
BASENAME	=	(contents table name)	**Required keyword**
TTYPE	=	(table name conversion - blank/F)	
LIB	=	(library name)	
ORG	=	(R/U/S/D/H)	
MTHD	=	(S/Q/B/C/H)	
KSZ	=	(key size)	
KLOC	=	(key location);	

Action

Creates an alternate definition on the library specified.

Note**:** If the parameter TTYPE=F is used, it will cause the names of the tables specified in the ALT and BASENAME parameters to be converted.

Use of a DEFINE Command

The following examples illustrate the use of all options of the DEFINE command:

Example 1

Initialize a tableBASE library identified by the ddname PRODLIB.

```
DEFINE      NEWLIB=PRODLIB;
```

Example 2

Define a new table, named TABLE02, on tableBASE library PRODLIB. The table is to be in ascending sequence by key and will be searched using a binary search. Two generations of the table are to be kept to provide a backup, and the table will contain approximately 300 rows when full. Each row will be 120 characters long and the 18 character key starts in position 5.

```
DEFINE      LIB=PRODLIB  TBL=TABLE02  ORG=S
            MTHD=B MAXGEN=2  ISZ=120  KSZ=18
            KLOC=5 EST=300;
```

Example 3

Create an alternate definition called EMPNAME for the data table EMPNUM. The alternate view will use a 40 byte key starting in position 18. The alternate view will describe a sequential table searched with a binary search.

```
DEFINE      ALT=EMPNAME  BASENAME=EMPNUM
            KSZ=40  KLOC=18  ORG=S  MTHD=B;
```

DELETE Command

The DELETE command deletes a specified or all generations of a table.

Option 1: Delete a Single Table Generation

This option is specified through the **absence** of ALLGEN=YES.

DELETE

TBL	=	(table name)	**Required keyword**
TTYPE	=	(table name conversion - blank/F)	
LIB	=	(library name)	
GEN	=	(generation number)	
PSWD	=	(write password);	

Action

If the WRITE PASSWORD is correct, the generation specified will be deleted.

Option 2: Delete all Generations of a Table

This option is specified through the **presence** of ALLGEN=YES.

DELETE

TBL	=	(table name)	**Required keyword**
TTYPE	=	(table name conversion - blank/F)	
LIB	=	(library name)	
ALLGEN	=	YES	
PSWD	=	(write password);	

Action

If the WRITE PASSWORD is correct, all generations of the table will be deleted.

Use of a DELETE Command

The following examples illustrate the use of both options of the DELETE command:

Example 1

Delete generation -1 (one prior to the current generation) of the table TABLE01 on the tableBASE library identified by the ddname PRODLIB. The table is write password protected.

```
DELETE   LIB=PRODLIB TBL=TABLE01 GEN=-1
         PSWD=BBBBBBBB;
```

Example 2

Delete TABLE01 and its associated View from the tableBASE library identified by the ddname PRODLIB. The table is write password protected.

```
DELETE      LIB=PRODLIB TBL=TABLE01 ALLGEN=YES
            PSWD=BBBBBBBB;

DELETE      LIB=PRODLIB TBL=TABLE02  TTYPE=F;
```

EXPAND Library Command

The EXPAND command increases the size of a library. This operation performs a block by block copy of the original contents. Whenever the size of a tableBASE library needs to be changed, a new library must be allocated and initialized, and the contents of the old one copied into it using the EXPAND command. Having been replaced in this manner, the original tableBASE library may then be deleted.

Unlike the COPY command, which modifies the date/time stamp for each table to the time of the copy operation, this command copies the LIB library to the NEWLIB library leaving the date/time stamps at the original table creation setting.

EXPAND

NEWLIB	=	(library name) **Required keyword**
LIB	=	(library name);

Action

The contents of tableBASE library (LIB) will be copied to tableBASE library 'NEWLIB' without change of the date/time stamp.

Note: The target library must be initialized and empty.

Note: Both datasets (LIB and NEWLIB) must have a DISP of OLD.

Note The default value for LIB is MAINLIB unless this has been overridden by a SET command.

Note: This command will not change the size of the NEWLIB's internal free space directory. An existing library may be EXPANDed to a new library only if the file system directories of the two libraries are compatible, i.e., they have the same number of file system directory blocks. A library with up to 168,000 blocks has seven directory blocks; one directory block is added for every 24,000 blocks above 168,000. This means that a library of fewer than 168,000 blocks can be expanded to only 168,000 blocks; a library of size between 168,000 and 192,000 can be expanded to only 192,000 blocks, and so on.

Note: This facility is only available for a library created with Release 4.2 or higher. If an older release needs to be increased in size, it will be necessary to create a new library using the DEFINE command followed by a COPY (Option 3) command, both described above.

Use of an EXPAND Command

The following example illustrates the use of the EXPAND command.

Example 1

Expand a tableBASE library MAINLIB into a library identified by the ddname LARGELIB. The target library has been previously initialized and is empty.

```
EXPAND      NEWLIB=LARGELIB;
```

EXPORT Library Command

The EXPORT command converts a tableBASE library to a sequential dataset that can be transported to another operating system. It provides a mechanism to synchronize tableBASE tables and libraries residing on different platforms. It is most commonly used in moving tables between the mainframe and the PC; this is described in detail in the tableBASE/PC User's Manual.

Export

LIB	=	(library name)	**Required keyword**
TO	=	(ddname);	**Required keyword**

Action

The library specified by the DDname defined in the LIB keyword will be copied to the file defined by the DDname specified in the TO keyword. The output dataset specified via the TO keyword should be defined as containing fixed length records of 512 bytes in length. The command will also list the number of tables exported.

Use of the EXPORT Command

The following example illustrates the use of the EXPORT command:

Example 1

Export a tableBASE library to a file to be downloaded to the PC.

EXPORT LIB=PRODLIB TO=DOWNLOAD;

IMPORT Library Command

The IMPORT command moves a sequential dataset created by the EXPORT command to a tableBASE library. This command, which is used in conjunction with the EXPORT command, provides a mechanism to synchronize tableBASE tables and libraries residing on different platforms. It is most commonly used in moving tables between the mainframe and the PC.

Import

LIB	=	(library name)	**Required keyword**
FROM	=	(ddname);	**Required keyword**
REPLACE	=	NO/YES	Replace tables on the library, NO is the default value

Action

The fixed length, sequential file defined by the DDname specified in the FROM keyword will be written to the library specified by the DDname defined in the LIB keyword The command will also list the number of tables imported.

Use of the IMPORT Command

The following example illustrates the use of the IMPORT command:

Example 1

Import a file created on the PC to a tableBASE library on the mainframe. Do not replace any tables with the same name on the file and the library.

IMPORT LIB=PRODLIB FROM=UPLOAD REPLACE=NO;

LOAD Command

The LOAD command loads a table from a sequential dataset into a tableBASE library, creating a new generation of the table. In order to perform this function, the table must already exist.

LOAD

TBL	=	(table name) **Required keyword**
FROM	=	(ddname) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
GEN	=	(generation number)
PSWD	=	(write password)
STARTREC	=	(record number at which loading starts)
RECCOUNT	=	(maximum rows to be loaded)
STARTBYT	=	(byte within each record at which loading starts);

Action

If the PSWD is the correct WRITE PASSWORD, a new generation of the named table will be created from the records of the dataset connected to the ddname of the FROM parameter. The definition of the generation specified will determine how the data will be stored. STARTREC and RECCOUNT can be used to control the records loaded; STARTBYT can be used to control the portion of each record loaded.

Note: The LOAD of a paged Hash table requires the allocation of SORT work files.

Note: In VSE the specification SORT=YES will enhance performance when loading a paged table. In order to use this facility, the SORT program must be relocatable.

Note: See the UNLOAD command which enables you to copy a table to a sequential dataset.

Use of LOAD Command

The following example illustrates the use of the LOAD command:

Example 1

Load table PARTNO from the file identified by the ddname PARTFLE. The table must have been DEFINEd prior to this command.

LOAD TBL=PARTNO FROM=PARTFLE;

PRINT Command

The PRINT command prints directory information, information about each generation of a table or the contents of a particular generation of a specified table.

Option 1: List Directory

This option of the PRINT command prints directory information for each table in a tableBASE library. This option is specified by the **absence** of the keywords DEF and TBL.

PRINT

LIB = (library name);

Action

This command will provide the following information for each table in the directory of the named tableBASE library:

a) Name
b) Table type
c) Absolute generation number of each generation present
d) The date and time that each generation was created.

In addition, the library dataset name will be printed along with the space allocated to the library and the amount remaining for use.

Option 2: Print Table Information

This option of the PRINT command lists a table definition. This option is specified through the presence of the keyword DEF.

PRINT

DEF = (table name) **Required keyword**
LIB = (library name)
TTYPE = (table name conversion - blank/F);

Action

For the named table (on the specified tableBASE library) the following table information will be printed for each generation:

a) The absolute generation number
b) The maximum number of generations to be retained
c) The number of generations existing at present
d) The relative generation number
e) The date and time created
f) The number of rows in the table
g) The number of rows allocated in the table

h) The table organization
i) The table search method
j) The index code
k) The storage mode code
l) The level of password protection
m) The row size
n) The key length
o) The key location
p) Expansion Factor
q) Upper Density
r) Lower Density
s) The actual density of the table
t) View version of a view
u) User id who last stored the table
v) View name of an associated view.

In addition, the library dataset name will be printed along with the space allocated to the library and the amount remaining for use.

Option 3: Print Table Contents

This option is specified through the presence of the keyword TBL.

PRINT

TBL	=	(table name) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
GEN	=	(generation number)
PSWD	=	(read password/write password)
PRNT	=	(print option - C/H/B)
LINSIZ	=	(print line size)
STARTREC	=	(row number at which printing starts)
RECCOUNT	=	(maximum rows to be printed)
STARTBYT	=	(byte number at which printing starts)
BYTCOUNT	=	(maximum bytes to be printed);

Action

For the table generation specified, this option of the PRINT command lists the contents of the table in character (C), hexadecimal (H) or both (B) character and hexadecimal modes depending on the value of the PRNT keyword selected. STARTREC and RECCOUNT will limit the portion of each table printed. In addition, the definition information for the specified table generation will be output in a format similar to that described in option 2.

Use of a PRINT Command

The following examples illustrate the use of the PRINT command:

Example 1

List the Directory of a tableBASE library whose ddname is MAINLIB.

```
PRINT     LIB=MAINLIB;
```

or

```
PRINT;
```

Example 2

List the definition of TABLE01 on tableBASE library MAINLIB.

```
PRINT     LIB=MAINLIB DEF=TABLE01;
```

Example 3

List the contents of Generation 2 of TABLE01 on tableBASE library PRODLIB. The listing is to be in hexadecimal format, 80 characters to a line and the table is not password protected.

```
PRINT     LIB=PRODLIB TBL=TABLE01 GEN=2
          PRNT=H LINSZ=80;
```

RENAME Command

The RENAME command renames all generations of a table on a given tableBASE Library.

RENAME

TBL	=	(table name)	**Required keyword**
NEWNAME	=	(new table name)	**Required keyword**
TTYPE	=	(table name conversion - blank/F)	
LIB	=	(library name)	
PSWD	=	(write password);	

Action

Renames all generations of a table on a given tableBASE Library.

Note**:** The RENAME will not be performed if a table with the 'new table name' already exists on the library.

Note**:** If the TTYPE parameter is used, it will cause both the table names specified in the TBL and the NEWNAME parameters to be converted.

Use of a RENAME Command

The following example illustrates the use of the RENAME command:

Example 1

Rename all generations of TABLE01 to TABLE02 and the associated View Table on the tableBASE library MAINLIB (default).

RENAME TBL=TABLE01 NEWNAME=TABLE02;

RENAME TBL=TABLE01 TTYPE=F NEWNAME=TABLE02;

REORG Command

The REORG command reorganizes a paged table when it has become too dense (see the tableBASE Programmer's Guide for a discussion of density).

REORG

TBL	=	(table name) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
GEN	=	(generation number)
PSWD	=	(write password);

Action

The specified generation of the paged table will be used as input to the reorganization. A new generation will be created with the density of the table set to the lower density that had been previously defined for this table. Distribution statistics for the access efficiency to this hashed table will be output to the TBRPT file.

Note**:** The REORG of a paged Hash table requires the allocation of SORT work files.

Note: In VSE the specification SORT=YES will enhance performance when reorganizing a paged table. In order to use this facility, the SORT program must be relocatable.

Note**:** A table will be reorganized automatically when the Storage Mode Code is changed. In addition, when a paged table has its KLOC, KSZ or ISZ CHANGEd in a TBEXEC run, the table will be reorganized to the lower density.

Note**:** A Hash table that is resident (SMC=R) will automatically be reorganized to its lower density at the first row insertion that causes the upper density to be exceeded.

Use of a REORG Command

The following example illustrates the use of the REORG command:

Example 1

Reorganize the latest generation of TABLE01 on the tableBASE library MAINLIB (default).

REORG TBL=TABLE01;

SET Command

The SET command sets default values for selected keywords.

SET

LIB	=	(library name)
PSWD	=	(password/*)
PRNT	=	(print option)
LINSIZ	=	(line size);

Action

Establishes default values for LIB, PSWD, PRNT and LINSIZ that will be used in subsequent commands, unless explicitly overridden.

Use of a SET Command

The following example illustrates the use of the SET command:

Example 1

Print the hexadecimal contents of several tables on library TESTLIB.

SET	LIB=TESTLIB PRNT=H;
PRINT	TBL=TABLE01;
PRINT	TBL=TABLE02;
PRINT	TBL=TABLE03;

UNLOAD Command

The UNLOAD command copies a specified generation of a table to an output dataset.

UNLOAD

TBL	=	(table name) **Required keyword**
TO	=	(ddname) **Required keyword**
TTYPE	=	(table name conversion - blank/F)
LIB	=	(library name)
GEN	=	(generation number)
PSWD	=	(read password/write password)
STARTREC	=	(record number at which unloading starts)
RECCOUNT	=	(maximum rows to be unloaded)
STARTBYT	=	(byte number in record at which unloading starts);

Action

If the PASSWORD is correct, the generation named will be output to the dataset connected to the DD statement named 'ddname'. STARTREC and RECCOUNT can be used to limit the records unloaded; STARTBYT can be used to control the portion of each record unloaded.

The dataset defined by the TO keyword contains fixed length records whose minimum length is the length of a table row.

Note: Hash tables will not be in key sequence when they are UNLOADed. Empty rows will be dropped.

Note: See the LOAD command which enables you to copy a sequential dataset to a table.

Use of an UNLOAD Command

The following example illustrates use of the UNLOAD command:

Example 1

Copy the contents of table PARTNO on the tableBASE Library MAINLIB to the dataset identified by the ddname PARTFLE.

UNLOAD LIB=MAINLIB TBL=PARTNO TO=PARTFLE;

UPDATE Command

The UPDATE command updates a table using data from a sequential dataset. Two files are available, one for 'adds' and 'changes', the other for 'deletes'.

UPDATE

TBL	=	(table name)	**Required keyword**
WITH	=	(ddname)	**WITH and/or**
REMOVE	=	(ddname)	**REMOVE are required**
TTYPE	=	(table name conversion - blank/F)	
LIB	=	(library name)	
GEN	=	(generation number)	
PSWD	=	(write password);	

Action

The key of the record is assumed to be in the same position as the key in the table.

For each record on the WITH file, if the key is matched in the table, the corresponding entry will be replaced in its entirety by the data record. If the key is not matched, the data record will be inserted in the table.

For each record on the REMOVE file, the table will be searched for a row with a matching key and if found the row will be deleted.

All actions taken during the update command will be logged on the TBRPT file.

Note: Only the first row found on the table with a key matching an input record will be updated or removed.

Note: If both WITH and REMOVE keywords are present, the rows are 'removed' before the WITH rows are processed.

Use of an UPDATE Command

The following examples illustrate for the use of the UPDATE command:

Example 1

Update table TABLEXX with the data on the file identified by the ddname UPDFILE, removing those rows whose keys are found on DELFILE.

```
UPDATE   TBL=TABLEXX WITH=UPDFILE
         REMOVE=DELFILE;
```

Example 2

A file is available containing a record for each product ordered in a company. The supplier is contained in position 20 for a length of 10. A list of all suppliers on the file is required. The following code will eliminate duplicate suppliers should there be multiple occurrences and then delete the temporary table.

```
DEFINE    TBL=SUPPLIER ORG=S KLOC=20 KSZ=10
          ISZ=30;
UPDATE    TBL=SUPPLIER WITH=PRODUCT;
PRINT     TBL=SUPPLIER;
DELETE    TBL=SUPPLIER ALLGEN=YES;
```

Chapter 3
Using TBEXEC

This chapter describes the use of TBEXEC. It begins by specifying the Job Control Language (JCL) required to run TBEXEC in MVS and VSE environments. Next, it describes how TBEXEC handles errors. Then, the standard reports produced by TBEXEC - the audit report (TBMSG) and the table listing report (TBRPT) - are described. The chapter concludes with a listing of the error and audit messages issued by TBEXEC.

Job Control Language

MVS Job Control Language

```
//STEP1         EXEC    PGM=TBEXEC
//STEPLIB       DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//TBMSG         DD SYSOUT=*
//TBRPT         DD SYSOUT=*
//TBDUMP        DD SYSOUT=*
//MAINLIB       DD DSN=*YOUR.PREFIX*.TBASE.LIBRARY,DISP=SHR
//CNTLCARD      DD *
TBEXEC command sequences;
/*

//SORTLIB       DD (these sort statements are required when
//SORTMSG       DD (LOADing, COPYing, CHANGEing or REORGanizing
//SORTWK01      DD (paged tables, or when COPYing a library
//SORTWK02      DD (containing one or more paged tables.  In these
//SORTWK03      DD (cases the SORTWKnn files must be large enough
                   (to contain all the rows in the largest paged table.)
```

The ddname TBRPT is required only if reports are to be produced.

To create a new tableBASE library:

1) Allocate the library data set either by JCL (BDAM, VSAM) or by IDCAMS (VSAM).

2) Format the data set with a TBEXEC DEFINE NEWLIB operation.

Typical MVS JCL to accomplish a) and b) is

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
       DEFINE CLUSTER ( NAME(your.vsam.library) -
              NUMBERED SPEED REUSE SHR(3) -
              RECSZ(3120 3120) CISZ(3584) -
              RECORDS(nnnn) -
              VOLUMES(yourvol) )
/*
//STEP2 EXEC PGM=TBEXEC
//STEPLIB  DD DSN=*YOUR.PREFIX*.TBASE.LOAD,DISP=SHR
//TBMSG    DD SYSOUT=*
//TBRPT    DD SYSOUT=*
//BDAMLIB  DD DSN=your.bdam.library,DISP=(,CATLG),
//            UNIT=yourunit,SPACE=(3120,nnnn,,,ROUND)
//VSAMLIB  DD DSN=your.vsam.library,DISP=OLD
//CNTLCARD DD *
       DEFINE NEWLIB=BDAMLIB;
       DEFINE NEWLIB=VSAMLIB;
/*
```

Equally valid variations are

Allocate BDAMLIB as DISP=(,CATLG) in STEP1 and reference it as DISP=OLD in STEP2

Allocate your VSAM library as DISP=(,CATLG) using VSAM JCL allocation in STEP2. and omit STEP1

Note: Our experience is that the control interval size cannot be set as desired using JCL to allocate a VSAM dataset. Although tableBASE VSAM supports control intervals of any allowable size, our recommendation is a size of 3584; thus we recommend the allocation of VSAM libraries by an IDCAMS step as shown above.

The library size (nnnn) may be estimated by summing the following:

a) Nine blocks (of 3120) for the library directory

b) Space to hold the contents of all generations of all tables. This value can be computed from the number of records and the size of each record in each table. Note that hash tables occupy twice as much space as non-hash tables.

c) 10 percent of (b) for overhead for definitions of the tables

d) Space for the largest table (one generation only). This is needed for updating of a table, as the new generation is stored on the library prior to removal of the oldest generation

e) Room for short term growth.

Once the library has been DEFINEd, the NEWLIB DD statement is replaced by:

```
//OLDLIB      DD DSN=*YOUR.PREFIX*.TBASE.LIBRARY,DISP=SHR
```

Note: In the JCL above, the DDnames MAINLIB, NEWLIB, and OLDLIB can be any name satisfying normal JCL rules. The names actually chosen are communicated to TBEXEC in the control statements.

When tableBASE LOADs a table from a dataset, the BLKSIZE of the input dataset does not matter; RECFM=FB must be specified. If the LRECL is greater than the row size, the record will be truncated. If the LRECL is less than the row size, the row will be padded with spaces. The dataset must have fixed length records.

When tableBASE UNLOADs a table to a dataset, either the dataset must exist or a complete DCB parameter must be provided. The table row will be truncated or padded with spaces to fit the LRECL of the output dataset.

VSE Job Control Language

The following example shows TBEXEC defining a new library and then defining and loading a new table:

```
*       $$ JOB JNM=TBEXECO,CLASS=A,DISP=D,USER=TABLEBASE
*       $$ LST JSEP=2,CLASS=Z,LST=cua1
*       $$ LST JSEP=2,CLASS=Z,LST-cua2
*       $$ LST JSEP=2,CLASS=Z,LST=cua3
*       $$ LST JSEP=2,CLASS=Z,LST-cua4
//JOB   TBEXEC  ***** TBEXEC RUN *****
//OPTION        PARTDUMP        Dump partition, if abends occurs
//PAUSE                 Point to tableBASE CIL
//ASSGN SYS001,cua                      SORT WORK AREA
//ASSGN SYS002,cua                      SORT WORK AREA
//ASSGN SYS003,cua                      SORT WORK AREA
//ASSGN SYS006,SYSIPT                   INPUT COMMANDS
//ASSGN SYS007,cua2                     OUTPUT (MSGFILE)
//ASSGN SYS008,cua3                     OUTPUT (RPTFILE)
//ASSGN SYS009,cua4                     OUTPUT (TBTSRPT)
//DLBL  SORTWK1,'your.sortwork.area.1',0
//EXTENT        SYS001,1,0,start,end
//DLBL  SORTWK2,'your.sortwork.area.2',0
//EXTENT        SYS002,1,0start,end
//DLBL  SORTWK3,'your.sortwork.area.3',0
//EXTENT        SYS003,1,0,start,end
//DLBL   IJSYSUC,'your.vsam.user.catalogue',,VSAM
//DLBL  TBOPT,'your.tbopt.file',,VSAM
//DLBL   MAINLIB,'your.tablebase.library',,VSAM
//DLBL  SEQFILE,'your.seq.file',,VSAM
//EXTENT        SYS002,PACK03
//EXEC  TBEXEC,SIZE=600K          *** TBEXEC statements after here ***
                DEFINE NEWLIB=MAINLIB;
                DEFINE TBL=TABLE01 ISZ=50  KSZ=5  KLOC=1
                        ORG=S  MTHD=Q  MAXGEN=5
                        RPSWD=RRRRR  WPSWD=WWWWW;
                LOAD TBL=TABLE01  FROM=SEQFILE  PSWD=WWWWW;
/*
/&
* $$ EOJ
```

Note: In the above example, the filenames MAINLIB and SEQFILE could have been any name satisfying normal Control Statement rules. The names actually chosen are communicated to TBEXEC via the control statements. The User Catalogue must be assigned if the VSAM files are not defined in the Master Catalogue.

When tableBASE LOADs a table from a dataset, the record length of the dataset does not matter. If the record length is greater than the row size, the record will be truncated. If the record length is less than the row size, the row will be padded with blanks. The dataset must have fixed length records.

When tableBASE UNLOADs a table to a dataset, the dataset must exist but must not contain any data (i.e., the dataset must have been DEFINEd with IDCAMS but never used). The table row will be truncated or padded with blanks to fit the record length of the output dataset.

Error Diagnosis

TBEXEC recognizes three classes of errors:

Command Interpretation Errors

Errors detected during interpretation of command sequences

Example

Invalid command, invalid keyword, missing delimiter.

Action

An error message is printed and TBEXEC skips to the next command sequence.

Command Processing Errors

Errors detected by TBEXEC during command processing

Example

Generation requested does not exist

Action

An error message is printed and processing resumes with the next command sequence.

TBLBASE Processing Errors

Errors detected within TBLBASE (TBLBASE is described in the tableBASE Programmer's Guide)

Example

Invalid key size, invalid password

Action

An error message is printed along with an interpretation of the return code set by TBLBASE. Processing resumes with the next command sequence. A list of these return codes can be found in the Programmer's Guide.

Note**:** If during the execution of TBEXEC running under MVS an error is encountered during interpretation or execution of any command sequence, the step return code is set to 16. A return code of zero means all command sequences were processed successfully.

Report Descriptions

TBEXEC produces two Report files, TBMSG and TBRPT.

The TBMSG Report

The TBMSG report displays the control statements, associated messages and any TBLBASE error codes returned while attempting to execute a command.

Figure 3a contains a sample of the TBMSG report. The following is a legend for its interpretation:

1. This area contains the name and address of the tableBASE client. It appears on the first page of the TBMSG Report.
2. '***REQUEST:' is followed by the command as input on the control card(s).
3. A message follows each command explaining what action was taken.
4. A syntax error was detected by TBEXEC. In this case the programmer misspelled 'STARTBYT'.
5. Although proper syntax was used, when the command was executed by TBLBASE an error occurred which prevented the successful completion of the command. In this case the correct Password needs to be provided. (See tableBASE error codes in the tableBASE Programmer's Guide).
6. This line is printed after all commands have been processed.

```
DATE/TIME: 1996/11/04 10:44                          TABLE BASE AUDIT REPORT
                     PAGE:   1

TABLEBASE R5.1 LICENSED TO:




*** REQUEST:  DEFINE  NEWLIB=TESTLIB;
    INITIALIZATION SUCCESSFUL

*** REQUEST:  SET     LIB=TESTLIB;
    DEFAULTS HAVE BEEN SET

*** REQUEST:  DEFINE  TBL=TBL01 ORG=S MTHD=B  MAXGEN=9  WPSWD=****

*** REQUEST:                    ISZ=40  KSZ=5 KLOC=3   EST=50;
    TABLE DEFINITION SUCCESSFUL

*** REQUEST:  LOAD    TBL=TBL01 FROM=ORIGINAL PSWD=****;
    NEW GENERATION LOADED

*** REQUEST:  CHANGE  TBL=TBL01 KLOC=38;
    TABLE *TBL01   * COULD NOT BE OPENED         PASSWORD IS NOT VALID FOR THIS TABLE
    CHANGE NOT MADE FOR REASONS ABOVE

*** REQUEST:  UPDATE  TBL=TBL01 WITH=ADD REMOVE=DELETE PSWD=****;
    TABLE UPDATED SUCCESSFULLY

*** REQUEST:  PRINT   TBL=TBL01 STARTBYTE=10 PSWD=****;
             STARTBYT... FIELD IS GREATER THAN 8 CHARACTERS
                     COMMAND REJECTED FOR REASON ABOVE     - SKIPPING TO NEXT COMMAND

*** REQUEST:  PRINT   TBL=TBL01 PSWD=****;
    PRINT REQUEST COMPLETED SUCCESSFULLY

*** REQUEST:  PRINT   DEF=TBL01 PSWD=****;
    PRINT REQUEST COMPLETED SUCCESSFULLY

*** REQUEST:  PRINT  ;
    PRINT REQUEST COMPLETED SUCCESSFULLY
                     END OF DATA - TABLE BASE UTILITY ENDED
```

FIGURE 3a

The TBRPT Report

The TBRPT report prints selected tables (in character, hexadecimal or both formats) and their related definitions. For printing columnar style reports based on the formatting information in the Views, refer to Chapter 4. Figures 3b - 3g are samples of the reports produced by TBEXEC.

Figures 3b and 3c are samples of update logs produced by TBEXEC when an UPDATE command is executed. Each log contains the key of each record on an update file and the action taken with each record. If an UPDATE uses both the 'WITH' and 'REMOVE' files, two logs will be produced.

```
DATE/TIME: 1996/11/04 10:44              TABLE BASE OPTIONAL REPORTS                    PAGE:   2

TABLE BASE UPDATE LOG FOR TABLE TBL01    ON LIBRARY TESTLIB  USING FILE ADD

   ACTION      RECORD  FIRST 100 BYTES OF KEY
                  SEQ
-----------  --------  ------------------------------------------------------------------------

INSERTED             1  BBB
REPLACED             2  GGG
INSERTED             3  OOO
INSERTED             4  ZZZ
END OF DATA              ********************************************************************************
```

FIGURE 3b

```
DATE/TIME: 1996/11/04 10:44              TABLE BASE OPTIONAL REPORTS                    PAGE:   3

TABLE BASE UPDATE LOG FOR TABLE TBL01    ON LIBRARY TESTLIB  USING FILE DELETE

   ACTION      RECORD  FIRST 100 BYTES OF KEY
                  SEQ
-----------  --------  ------------------------------------------------------------------------

DELETED              1  CCC
NOT DELETED          2  EEE
DELETED              3  NNN
DELETED              4  TTT
END OF DATA              ********************************************************************************
```

FIGURE 3c

Figures 3d and 3e are samples of directory listings that are produced by TBEXEC. Figure 3d is a Directory Listing for the tableBASE library 'TESTLIB', which is on the dataset 'SYS96309.T104402.RA000.DKLS04X.LIB'. Figure 3e provides details as to the contents of the TESTLIB library. As we see, the library contains only the three generations of TBL01. The listing shows when the generations were created, how many blocks in the library are currently being used and how many blocks remain.

```
DATE/TIME: 1996/11/04 10:44              TABLE BASE OPTIONAL REPORTS                    PAGE:   1

 DIRECTORY LISTING FOR LIBRARY TESTLIB  ON DATASET SYS96309.T104402.RA000.DKLS04X.LIB

    BLOCKS IN LIBRARY TESTLIB  : 00000026    BLOCKS REMAINING: 00000018
```

FIGURE 3d

```
DATE/TIME: 1996/11/04 10:44                    TABLE BASE OPTIONAL REPORTS                    PAGE:   6

DIRECTORY LISTING FOR LIBRARY TESTLIB  ON DATASET SYS96309.T104402.RA000.DKLS04X.LIB

TABLE NAME   TABLE TYPE   GENERATION NO.    DATE CREATED    TIME CREATED

 TBL01                          3          1996/11/04       10:44
 TBL01                          2          1996/11/04       10:44
 TBL01                          1          1996/11/04       10:44

   BLOCKS IN LIBRARY TESTLIB  : 00000026    BLOCKS REMAINING: 00000013
```

FIGURE 3e

Figures 3f and 3g are samples showing the definition of a table, TBL01.

Figure 3f is a summary report which identifies the table type, the absolute generation number, the date created (YYYY/MM/DD), and the time created (HH:MM), for each table generation on the library. Possible values for table type:

blank	Data table
VIEW	View
ALT	Alternate table index
ALT VIEW	Alternate view index.

```
DATE/TIME: 1996/11/04 10:44                    TABLE BASE OPTIONAL REPORTS                    PAGE:   5

DEFINITION FOR TABLE TBL01    ON LIBRARY TESTLIB  ON DATASET SYS96309.T104402.RA000.DKLS04X.LIB

 GENERATION  DATE AND TIME       NUMBER OF ROWS   O M I S  SECURITY    ROW  KEY   KEY  EXP  ----DENSITY----- VW USERID     VIEW
ABS M A REL      CREATED       PRESENT  ALLOCATED R/E/N/M             SIZE SIZE   LOC FACTR UPPER LOWER  ACT VS            NAME

  3 9 3  0 1996/11/04 10:44         10         13 S B T R READ/WRITE    40   5     3  20.0% 80.0% 50.0%  N/A    DKLS04
  2 9 3 -1 1996/11/04 10:44         10         50 S B T R READ/WRITE    40   5     3  20.0% 80.0% 50.0%  N/A    DKLS04
  1 9 3 -2 1996/11/04 10:44          0         50 S B T R READ/WRITE    40   5     3  20.0% 80.0% 50.0%  N/A    ________
```

FIGURE 3f

The report indicates the number of blocks allocated to the library and how many remain for additional tables. Each block contains 3120 characters of information.

This report also lists the Definition for each generation of TBL01. From left to right on the report are the absolute generation number, the maximum number of generations, the number of generations currently on the library, the relative generation number, the creation date and time, the actual number of rows, the number of rows for which space has been allocated, the table organization and search method, whether the table is true or accessed through an index, the storage mode code, whether read and write passwords have been used, the size of each row, the length and location of the key, the expansion factor, the upper and lower densities, the actual density of hash tables, the version of the View and the user id.

Figure 3g lists the contents of one generation of a table in character format. It shows the name of the table being printed, TBLO1, and the library where it resides. The Definition of the generation being printed is displayed in the same format described above.

```
DATE/TIME: 1996/11/04 10:44                TABLE BASE OPTIONAL REPORTS                PAGE:   4

DEFINITION FOR TABLE TBL01    ON LIBRARY TESTLIB  ON DATASET SYS96309.T104402.RA000.DKLS04X.LIB

GENERATION  DATE AND TIME        NUMBER OF ROWS   O M I S  SECURITY    ROW KEY   KEY  EXP  ----DENSITY----- VW USERID     VIEW
ABS M A REL      CREATED        PRESENT ALLOCATED R/E/N/M             SIZE SIZE  LOC FACTR UPPER LOWER  ACT VS            NAME

  3 9 3  0 1996/11/04 10:44          10        13 S B T R READ/WRITE    40   5     3 20.0% 80.0% 50.0%  N/A    DKLS04

CONTENTS OF TABLE TBL01    ON LIBRARY TESTLIB

   ITEM NO.    LOCN      CONTENTS

       1       1      AAAAA  ORIGINAL TEST DATA
       2       1      BBBBB  NEW TEST DATA
       3       1      DDDDD  ORIGINAL TEST DATA
       4       1      GGGGG  NEW TEST DATA
       5       1      JJJJJ  ORIGINAL TEST DATA
       6       1      MMMMM  ORIGINAL TEST DATA
       7       1      OOOOO  NEW TEST DATA
       8       1      RRRRR  ORIGINAL TEST DATA
       9       1      YYYYY  ORIGINAL TEST DATA
      10       1      ZZZZZ  NEW TEST DATA
```

FIGURE 3g

For each Line output, the location indicates the position within the row at which printing begins. It is possible for longer rows to require several lines to report.

Error and Audit Messages with Explanation

AVERAGE EXCPS EXPECTED ON RETRIEVAL =
This is an estimate of the number of I/O operations on the tableBASE library required to LOAD, COPY or re-organize a paged table.

CHANGE DEFINITION FAILED
The attempt to change the definition of the table has failed for the reasons noted above the message.

CHANGE IS NOT FOR ALTERNATE DEFINITION
The 'CHANGE ALT=' command specified a table that is not an alternate table. Use the 'CHANGE TBL=' version of the command.

CHANGE KEY SIZE/LOCATION FAILED
The attempt to change the key size and/or key location has failed for reasons noted in preceding messages or to the right of this message.

CHANGE MAXGEN FAILED
The attempt to change the number of generations to be kept has failed for reasons noted in preceding messages or to the right of this message.

CHANGE NOT MADE FOR REASONS ABOVE
None of the changes in the above CHANGE were performed. The reason for the rejection of this command is noted above the message.

CHANGE SUCCESSFUL
The table definition has been changed as requested.

CHANGE WRITE PASSWORD INVALID ON REORG
None of the changes in the above CHANGE were performed. The WRITE PASSWORD cannot be changed in a command that requires the reorganization of a paged table.

COMMAND IS INVALID
The specified command is not a valid TBEXEC command.

COMMAND REJECTED FOR REASON ABOVE
The preceding TBEXEC Command was rejected for the reasons noted above the message.

COMMAND IS INVALID FOR ALTERNATE
The command cannot be performed on an alternate index table.

CONTENTS TABLE IS NOT POINTER
TBEXEC attempted to invoke an alternate view of a table that is not defined to be type ‘P’ (indexed or pointer).

COPY COMPLETE
The requested operation completed successfully.

COPY COMPLETE EXCEPT WHERE INDICATED
The requested copy operation was successfully completed except as noted above the message.

COPY FAILED
The requested operation failed. Diagnostics messages are issued to give the reason(s) for the failure.

COPY OF TABLE (ALL GENERATIONS) COMPLETE
The requested copy operation of all generations of a table was successfully completed

COUNT IS NOT WITHIN RANGE OF TABLE
The count is less than one, or greater than the number of rows in the table.

CREATE ALTERNATE DEFINITION FAILED
The attempt to create an alternate definition failed for the reasons noted in preceding messages or to the right of this message.

CREATE ALTERNATE DEFINITION SUCCESSFUL
The Alternate definition has been created successfully.

DDNAME NOT ASSIGNED - CHECK JCL
The LIB=ddname is not in the JCL.

DEFAULTS HAVE BEEN SET
The defaults entered have been set.

DELETE FAILED
The requested delete operation has failed for reasons noted in preceding messages or to the right of this message.

DELIMITER IS INVALID, = WAS EXPECTED
The next symbol after a keyword must be an equal sign.

DENSITY PARAMETER IS INVALID
The density must be numeric and in the range 000 through 999.

DESTINATION LIBRARY TOO SMALL
The destination library is too small to hold the table(s) to be copied.

DIRECTORY IS EMPTY
The Directory on the specified library is empty.

DUPLICATE KEYWORD FOR THIS COMMAND
The same keyword was specified twice for a command; fix the command sequence and try again.

END OF DATA - TABLE BASE UTILITY ENDED
End of data has been reached on CNTLCARD, the TBEXEC input file.

EST PARAMETER IS INVALID
The estimated number of rows is non-numeric, or is too large.

EXPAND LIBRARY FAILED
The requested operation failed. Diagnostics messages are issued to give the reason(s) for the failure.

EXPAND LIBRARY SUCCESSFUL
The requested operation completed successfully.

EXPFACTR PARAMETER IS INVALID
The expansion factor must be numeric and in the range 000 through 999.

ERROR IN TABLE DEFINITION
An error has been detected in the table definition created from the parameters supplied. Refer to the TBCALL Error Code for the reason.

FIELD IS GREATER THAN 8 CHARACTERS
All of the keywords and keyword values have a maximum length of eight characters.

xxxxxxxx FIELD IS GREATER THAN 8 CHARACTERS
All keywords and keyword values must be eight characters or fewer.

GENERATION NUMBER IS INVALID
The generation must be numeric and in the range 0 through 255.

GENERATION REQUESTED HAS BEEN COPIED
The COPY operation has been completed successfully. A new table (generation number 1) has been added to the target library.

INDEX PARAMETER IS INVALID
The index field must be 'T' for true tables or 'P' for indexed (pointer) tables.

INITIALIZATION SUCCESSFUL
The new tableBASE library has been initialized successfully.

INSUFFICIENT SPACE ON NEW LIBRARY
There is not enough space on the target library for the tables being copied.

INVALID TABLEBASE LIBRARY
The library is pre-release 4.2 of tableBASE. It cannot be processed by the EXPAND command.

ISZ CANNOT BE CHANGED ON PAGED TABLES
The row size cannot be changed for a paged table.

ISZ CANNOT EXCEED 1000 FOR PAGED TABLES
The maximum row size for paged tables is 1000.

ISZ PARAMETER IS INVALID
The row size must be numeric in the range 1 through 32,767, inclusive.

KEY IS NOT CONTAINED IN ITEM
The end of the key cannot exceed the end of the row.

KEYWORD AND VALUE ARE INCOMPATIBLE
If the keyword requires a numeric value, the value specified with it must be numeric: alternatively, the ALLGEN keyword must be followed by '= YES'. In either case, failure to have an appropriately formatted value for the keyword will result in this message.

KEYWORD INVALID FOR THIS COMMAND
This keyword is not valid for this command.

KEYWORD IS INCOMPLETE
The keyword has not been followed by an '=' and a value.

KEYWORD IS INVALID
The KEYWORD specified was not recognized.

KLOC PARAMETER IS INVALID
The key location must be numeric in the range 1 to the row size, inclusive.

KSZ PARAMETER IS INVALID
The key size must be a number from 1 to 256, inclusive.

KSZ/KLOC CANNOT BE CHANGED ON PAGED TBLS
Neither the key size nor location can be changed for a paged table.

LIBRARY DISPOSITION IS INVALID
tableBASE library disposition must be NEW for new libraries, and SHR or OLD for existing libraries.

LIBRARY IS NOT EMPTY
The target or destination library of an EXPAND LIBRARY request contains at least one table. The target library must be empty for an EXPAND operation.

LIBRARY NOT INITIALIZED
For the reason cited to the right of this message, the library was not initialized.

LOAD FAILED
The requested operation failed. Diagnostics messages are issued to give the reason(s) for the failure.

MAXGEN PARAMETER IS INVALID
The maximum generations to be kept must be numeric in the range 1 through 9, inclusive.

MTHD PARAMETER IS INVALID
The search method must be one of: S, Q, B, C or H.

MTHD/ORG PARAMETERS ARE INCOMPATIBLE
This combination of search method and organization will not work. Valid combinations are

Organization	Method
R,U	S
S,D	S,B,C
H	H

NEW GENERATION LOADED SUCCESSFULLY
A new generation of this table has been created from the contents of the FROM dataset.

NEWLIB DD NAME NOT ASSIGNED -- CHECK JCL
The requested initiation of the ddname specified by the NEWLIB parameter could not be performed since the ddname referred to is not defined in the Job Control Language.

NEW NAME ALREADY EXISTS
The table was not renamed because a table with the new name already exists on the library.

NO CHANGES SPECIFIED
A change command did not give any fields to be changed.

NOT COPIED, NEW NAME SAME AS OLD NAME
A COPY table request specified the same value for NEWNAME as for TBL.

NOT ENOUGH SPACE FOR DUMP/RESTORE
During a table LOAD operation, tableBASE needed to expand the table, but could not save the table temporarily on the tableBASE library. This message implies that the target library may not be large enough to hold the table once all rows have been LOADed. Either use a larger library, or free some library space by deleting obsolete tables.

NOT ENOUGH SPACE ON LIBRARY
There is not enough space on the tableBASE library for the table being defined, copied, loaded or expanded.

OLD ALTERNATE DEFINITION DELETED
During a COPY of an alternate definition, the old definition was deleted from the target library, but the new definition could not be copied from the source library to the target library.

ORG PARAMETER IS INVALID
The Organization must be one of R, U, S, D or H.

ON DEST LIB - INSUFFICIENT SPACE FOR COPY
The copy operation has been requested to a destination library with insufficient space on it to receive all generations of the tables on the "FROM" library.

PAGED TABLE CANNOT BE DIVERTED
A paged table can only be copied to a new library by using TBEXEC. This error should not occur when using TBEXEC.

PAGED TABLE IS FULL
A Paged table cannot be expanded after it is defined. TBEXEC attempted to add more rows than the table would hold.

PAGED TABLE : LATEST GEN. COPIED
Paged tables with multiple generations share their contents. If these tables are copied, they can no longer share their contents and consequently may take up far more space than originally anticipated.

For this reason, only one generation of a paged table can be copied. When all generations of a paged table have been requested for copy or when the entire library is being copied, only the most recent generation of a paged table will be copied.

PAGED TABLE MUST HAVE HASH ORGANIZATION
A paged table can only be organized as a hash table.

PASSWORD IS NOT VALID FOR THIS TABLE
An incorrect password has been specified.

POINTER INDEX NOT ALLOWED ON PAGED TABLE
Paged tables must have Storage Mode Code = 'R' (real). They cannot be indexed (pointer) tables).

PRINT REQUEST COMPLETED SUCCESSFULLY
The requested print operation has finished with no errors. The listing can be found on the file TBRPT.

REGION SIZE TOO SMALL
The MVS region available to TBEXEC is too small to hold the program and the requested TSR (Table Space Region) (or the requested table if there is no TSR). Specify a larger region on the JCL EXEC card.

RENAME FAILED
The requested operation failed. Diagnostics messages are issued to give the reason(s) for the failure.

RENAME SUCCESSFUL
The requested RENAME operation was successfully performed.

REORGANIZATION ATTEMPT FAILED
For the reasons noted to the right of this message, the requested reorganization of a paged table could not be successfully completed.

REORGANIZATION OF TABLE SUCCESSFUL
The requested reorganization was successful.

REQUESTED GENERATION(S) DELETED
The requested operation completed successfully.

REQUESTED GENERATION HAS BEEN CLEARED
A new generation of this table has been created using the definition of the generation specified. This new generation contains no items.

REQUESTED GENERATION HAS BEEN DELETED
The generation of the table specified has been deleted.

REQUIRED KEYWORD MISSING FOR THIS COMMAND
One of the required keywords for this command was not contained in the command sequence. Fix the input and try again.

SEMI-COLON IS MISSING FROM COMMAND
The command and its keywords must be followed by a semi-colon (;) to indicate the end of the keyword list for the command sequence.

SMC PARAMETER IS INVALID
The storage mode code (SMC) must be R or P.

SOURCE LIBRARY IS EMPTY
The source library in a COPY operation contains no tables.

TABLE ALREADY EXISTS ON LIBRARY
The table could not be defined on, or copied to, the new library because a table of the same name already exists on the target library.

TABLE BASE ERROR DETECTED
The requested operation failed. Diagnostics messages are issued to give the reason(s) for the failure.

TABLE BASE INTERNAL PROGRAM ERROR - XXXX
The requested operation failed. Diagnostics messages are issued to give the reason(s) for the failure.

TABLE DEFINITION SUCCESSFUL
The table was defined successfully and generation 1 was stored on the library.

TABLE IN USE
TBEXEC could not store the table because it is locked by another application.

TABLE IS NOT CLOSED
This error should not occur when using TBEXEC.

TABLE IS NOT ON SPECIFIED LIBRARY
The table was not found on the given library.

TABLE IS NOT OPEN
This error should not occur when using TBEXEC.

TABLE NAME IS INVALID
tableBASE table names can be any eight characters, including non-display characters, except all spaces or all low-values or ':TMPNAME' or '$$$$$$$$'.

TABLE OPENED FOR READ CANNOT BE STORED
A table has been opened for read only and a store or write command has been issued.

TABLE SPACE REGION TOO SMALL
The TSR (Table Space Region) is not large enough to contain the table and tableBASE internal tables. (tableBASE uses a small part of the TSR for its own internal tables).

TABLE TO BE REORGANIZED IS NOT PAGED
The REORG command can only be used with paged tables.

TABLE TYPE IS INVALID
The table type must be F, A or X.

TABLE UPDATED AS REQUESTED
The requested update operations on the specified table were performed successfully.

TABLE UPDATED SUCCESSFULLY
The requested operation completed successfully.

TABLE UNLOADED SUCCESSFULLY
The specified generation of the specified table has been unloaded to the "TO" dataset.

TABLE xxxxxxxx COULD NOT BE OPENED
For reasons noted in preceding messages or to the right of this message, the table xxxxxxxx could not be opened. Since the table must be opened to perform

this TBEXEC command, the TBEXEC command could not be successfully completed.

TABLE xxxxxxxx EXPORT FAILED
The export of table xxxxxxxx failed for the reasons given in other messages.

TABLE xxxxxxxx IMPORT FAILED
The import of table xxxxxxxx failed for the reasons given in other messages.

TABLE xxxxxxxx IMPORT REPLACE FAILED
The import of table xxxxxxxx failed for the reasons given in other messages.

TABLE xxxxxxxx SORT ERROR
The sort required for this operation failed for the reason noted.

TABLE xxxxxxxx WAS NOT STORED
The table xxxxxxxx was not stored for the reasons noted in preceding messages or to the right of this message.

nnnnn/mmmmm TABLES EXPORTED SUCCESSFULLY
nnnnn tables were exported from the tableBASE library successfully. mmmmm tables were requested to be exported.

nnnnn/mmmmm TABLES IMPORTED SUCCESSFULLY
nnnnn tables were imported to the tableBASE library successfully. mmmmm tables were requested to be imported.

tableBASE ERROR DETECTED
A TBLBASE type tableBASE error was detected in the processing of this command. The explanation of the error is found to the right of this message.

TOO FEW OPERANDS
TBEXEC called tableBASE incorrectly.

TOO MANY OPERANDS
TBEXEC called tableBASE incorrectly.

WRITE PASSWORD NOT SUPPLIED
The operation requires access to a write-protected table, but the write password has not been specified.

Chapter 4
Table Printing Utility
TBPRINT

TBPRINT prints formatted columnar listings using the formatting information in Views. This differs from the TBEXEC print command which is intended to supply whole or partial table dumps both in character and hexadecimal formats.

Introduction

This utility has free format input using keywords that invoke print functions. Some keywords may be followed by one or more values; others invoke print functions by themselves. The general format is:

PRINT KEYWORD value[,value] . . . KEYWORD ;

The keywords may be entered in any sequence. Should keywords be repeated in the same PRINT statement, the last occurrence of the keyword will be used. The semicolon is a terminating character for the PRINT statement. Many PRINT requests may be entered in the same input.

Completion Codes

Completion codes are set to indicate whether problems were encountered:

- Completion Code **0**
  All PRINT commands were successfully processed.
- Completion Code **4**
  A warning was issued in conjunction with one or more PRINT commands, but printing was able to be performed.
- Completion Code **16**
  One or more errors were encountered which prevented the printing of a table. Warnings may also have been issued.

Keyword Parameters

The input parameters for TBPRINT can be one of two forms: Keywords with assigned values and Keywords without assigned values.

Keywords with Assigned Values

The following delimiters can be used to separate multiple values for a keyword:

commas '**,**'

blanks ' '

left parenthesis '**(**'

right parenthesis '**)**'

equal sign '**=**'

double quotes ' " '.

Equal signs to separate the KEYWORD and VALUES, and commas to separate values lead to a readable style of code. It is this format that will be used in the examples that follow.

KEYWORD value[,value]... ;

LIB
Specify the libraries to be searched for a table. One to ten library names of up to eight characters each are allowed. If the keyword is omitted, searching will be restricted to MAINLIB

Example: LIB=TESTLIB1,TESTLIB2

TABLE
Indicate which tables are to be printed. One to eight character table name(s) are allowed.

Example: TABLE=CODE1,CODE2,CODE3

PASSWORD
Supply a one to eight character read password, if the table has one. If the keyword is omitted, the table(s) are assumed to have no password protection

Example: PASSWORD=SECRET

Note: Separate PRINT statements will have to be used if the data tables specified in the TABLE keyword have different read passwords.

GEN
Supply a number ranging from -8 through to 255 representing the relative or absolute generation number of the table(s) being printed. The default is zero, referring to the current generation.

Example: GEN=-1

Note: Separate PRINT statements must be used if more than one generation number is to be printed.

```
Example:  PRINT    TABLE=CODE1 GEN=-2;
          PRINT    TABLE=CODE1 GEN=-1;
          PRINT    TABLE=CODE1;
```

USING

Specify a one to eight character View name to be used. The default is to use the same name as the table being printed.

```
Example:  PRINT    TABLE=CODE1,CODE2,CODE3
```

TITLE

Supply a one to fifty character title, enclosed in double quotes OR
Supply a one to fifty character title with spaces filled with '-' (hyphen characters) or '_' (underscore characters), which are then replaced by blanks in TBPRINT. The title will be centered on the report page.

Note: Enclose the title in double quotes if hyphens or underscores are to appear in the title.

```
Example:  PRINT
          TABLE=CODE1
          TITLE=CODE_TABLE;
          PRINT
          TABLE=CODE2
          TITLE="PAY-CODE-TABLE";
```

ROWIDCOUNT

In order to identify the fields that overflow onto a second or subsequent page the Row Id Count 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.

Row ID Count is normally used to re-display the row keys on the overflow pages. See the formatting example in the next section.

The default is zero, indicating that no fields are to be repeated

```
Example:  PRINT    TABLE=CODE1 ROWIDCOUNT=2;
```

LINESPERPAGE

Specify the maximum number of lines to be printed on a page before a page break. The default is 60.

```
Example:  PRINT    TABLE=CODE1 LINESPERPAGE=9999;
```

PAGEWIDTH

Specify the maximum number of characters to be printed per line for a particular printer.

Do not include the carriage control character in the PAGEWIDTH. The default and the maximum are both 132.

```
Example:  PRINT    TABLE=CODE1 PAGEWIDTH=80;
```

FIELDS
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 can 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 can 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.

STARTREC
Specify the row in a data table from which reporting is to begin.

STARTKEY
Specify the key in a data table from which reporting is to begin. If the key is not found in an ordered table (either ascending or descending order), reporting will begin with the row referenced by the current value of the COUNT defined in the command area.

RECCOUNT
Specify the number of rows to print. If neither STARTREC nor STARTKEY is specified, reporting will begin at the first row of the table.

Keywords without Assigned Values

Keyword ;

FORMAT
Signals that format overrides follow. It is optional, serving only to make the PRINT command more readable.

One of:

HEADINGS	Title and column headings. (This is the **default.**)
NOHEADINGS	No title nor column headings.
TITLESONLY	Only title headings.
COLUMNHEADONLY	Only column headings.

One of:

VERTICAL	Data columns that cannot fit on one page, start after the entire table is printed. (This is the **default.**)
HORIZONTAL	Data columns that cannot fit on one page, start on the next page.
TRUNCATE	Data columns that cannot fit on one page are truncated.

Either of:

COLUMNSPLITON	For long fields, truncate last field on a page, if necessary.
COLUMNSPLITOFF	For long fields, maintain complete field. (This is the **default.**)

Either of:

DISPLAYCOUNTON	Use last 5 columns of report for displaying an row count number.
DISPLAYCOUNTOFF	Use full page for printing. (This is the **default**.)

Formatted Print Examples

To illustrate **FORMATTING** consider the situation where all fields of a table being printed do not fit on the page. (The page size is controlled by the **PAGEWIDTH** keyword.)

```
PAGE 1              TITLE
  F1      F2          F3              F4          F5

______  ______  ____________  ____________  ______
AAAAAA  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
BBBBBB  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
CCCCCC  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
DDDDDD  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
EEEEEE  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX

PAGE 2              TITLE
  F1      F2         F3             F4           F5

______  ______  ____________  ____________  ______
FFFFFF  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
GGGGGG  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
HHHHHH  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
IIIIII  XXXXXX  XXXXXXXXXXXX  XXXXXXXXXXXX  XXXXXX
```

Print EXAMPLE Table Using Defaults

PRINT TABLE=EXAMPLE; The defaults will be **VERTICAL, COLUMNSPLITOFF**

```
PAGE 1        TITLE
 F1        F2          F3
______   _______   _____________
AAAAAA   xxxxxx   xxxxxxxxxxxx
BBBBBB   xxxxxx   xxxxxxxxxxxx
CCCCCC   xxxxxx   xxxxxxxxxxxx
DDDDDD   xxxxxx   xxxxxxxxxxxx
EEEEEE   xxxxxx   xxxxxxxxxxxx


PAGE 2        TITLE
  F1        F2          F3
______   _______   _____________
FFFFFF   xxxxxx   xxxxxxxxxxxx
GGGGGG   xxxxxx   xxxxxxxxxxxx
HHHHHH   xxxxxx   xxxxxxxxxxxx
IIIIII   xxxxxx   xxxxxxxxxxxx


PAGE 1      TITLE
     F4           F5
_____________   ______
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx


PAGE 2      TITLE
     F4           F5
_____________   ______
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx
xxxxxxxxxxxx   xxxxxx
```

Print EXAMPLE with Row Identification

PRINT TABLE=EXAMPLE ROWIDCOUNT=1;

The defaults remain **VERTICAL** and **COLUMNSPLITOFF**, however F1 is repeated for identification.

```
PAGE 1        TITLE
  F1       F2          F3
_______  _______  _____________
AAAAAA   xxxxxx   xxxxxxxxxxxx
BBBBBB   xxxxxx   xxxxxxxxxxxx
CCCCCC   xxxxxx   xxxxxxxxxxxx
DDDDDD   xxxxxx   xxxxxxxxxxxx
EEEEEE   xxxxxx   xxxxxxxxxxxx


PAGE 2        TITLE
  F1       F2          F3
_______  _______  _____________
FFFFFF   xxxxxx   xxxxxxxxxxxx
GGGGGG   xxxxxx   xxxxxxxxxxxx
HHHHHH   xxxxxx   xxxxxxxxxxxx
IIIIII   xxxxxx   xxxxxxxxxxxx


PAGE 1         TITLE
  F1           F4           F5
_______  _____________  _______
AAAAAA   xxxxxxxxxxxx   xxxxxx
BBBBBB   xxxxxxxxxxxx   xxxxxx
CCCCCC   xxxxxxxxxxxx   xxxxxx
DDDDDD   xxxxxxxxxxxx   xxxxxx
EEEEEE   xxxxxxxxxxxx   xxxxxx


PAGE 2        TITLE
  F1           F4           F5
_______  _____________  _______
FFFFFF   xxxxxxxxxxxx   xxxxxx
GGGGGG   xxxxxxxxxxxx   xxxxxx
HHHHHH   xxxxxxxxxxxx   xxxxxx
IIIIII   xxxxxxxxxxxx   xxxxxx
```

Print EXAMPLE with Row Truncation

PRINT TABLE=EXAMPLE TRUNCATE;

The COLUMNSPLITOFF default keeps the last field from being split.

```
PAGE 1       TITLE
  F1      F2         F3
______  ______     ____________
AAAAAA  XXXXXX     XXXXXXXXXXXX
BBBBBB  XXXXXX     XXXXXXXXXXXX
CCCCCC  XXXXXX     XXXXXXXXXXXX
DDDDDD  XXXXXX     XXXXXXXXXXXX
EEEEE   XXXXXX     XXXXXXXXXXXX
PAGE 2       TITLE
  F1     F2          F3
______  ______     ____________
FFFFFF  XXXXXX     XXXXXXXXXXXX
GGGGGG  XXXXXX     XXXXXXXXXXXX
HHHHHH  XXXXXX     XXXXXXXXXXXX
IIIIII  XXXXXX     XXXXXXXXXXXX
```

PRINT TABLE=EXAMPLE TRUNCATE COLUMNSPLITON;

Fills the page.

```
PAGE 1              TITLE
  F1      F2         F3           F4
______  ______     ____________ ___
AAAAAA  XXXXXX     XXXXXXXXXXXX XXX
BBBBBB  XXXXXX     XXXXXXXXXXXX XXX
CCCCCC  XXXXXX     XXXXXXXXXXXX XXX
DDDDDD  XXXXXX     XXXXXXXXXXXX XXX
EEEEEE  XXXXXX     XXXXXXXXXXXX XXX

PAGE 2          TITLE
  F1      F2         F3           F4
______  ______     ____________ ___
FFFFFF  XXXXXX     XXXXXXXXXXXX XXX
GGGGGG  XXXXXX     XXXXXXXXXXXX XXX
HHHHHH  XXXXXX     XXXXXXXXXXXX XXX
IIIIII  XXXXXX     XXXXXXXXXXXX XXX
```

Print EXAMPLE with Rows Continuing on Next Page

PRINT TABLE=EXAMPLE HORIZONTAL ROWIDCOUNT=1;

F1 is repeated for identification.

```
PAGE 1          TITLE
  F1     F2          F3
______  ______   _____________
AAAAAA  xxxxxx     xxxxxxxxxxxx
BBBBBB  xxxxxx     xxxxxxxxxxxx
CCCCCC  xxxxxx     xxxxxxxxxxxx
DDDDDD  xxxxxx     xxxxxxxxxxxx
EEEEEE  xxxxxx     xxxxxxxxxxxx


PAGE 1          TITLE
  F1         F4           F5
______  _____________  ______
AAAAAA  xxxxxxxxxxxx   xxxxxx
BBBBBB  xxxxxxxxxxxx   xxxxxx
CCCCCC  xxxxxxxxxxxx   xxxxxx
DDDDDD  xxxxxxxxxxxx   xxxxxx
EEEEEE  xxxxxxxxxxxx   xxxxxx


PAGE 2        TITLE
  F1       F2          F3
______  ______  _____________
FFFFFF  xxxxxx  xxxxxxxxxxxx
GGGGGG  xxxxxx  xxxxxxxxxxxx
HHHHHH  xxxxxx  xxxxxxxxxxxx
IIIIII  xxxxxx  xxxxxxxxxxxx

PAGE 2        TITLE
  F1         F4           F5
______  _____________  ______
FFFFFF  xxxxxxxxxxxx   xxxxxx
GGGGGG  xxxxxxxxxxxx   xxxxxx
HHHHHH  xxxxxxxxxxxx   xxxxxx
IIIIII  xxxxxxxxxxxx   xxxxxx
```

MVS Job Control Language

An example of an execution of **TBPRINT** follows

```
//STEP1   EXEC  PGM=TBPRINT
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//*
//TBDUMP   DD SYSOUT=*
//TBLOG    DD SYSOUT=*
//TBREPORT DD SYSOUT=*
//*
//MAINLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.MAINLIB
//ABCXYZ   DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.ABCXYZ
//*
//*     This is sample input.
//*
//TBSYSIN  DD *
   PRINT TABLE=CODE1 LIB=MAINLIB;
   PRINT TABLE=CODE2 LIB=ABCXYZ GEN=-1 FORMAT HORIZONTAL;
/*
```

VSE Job Control Language

```
$$ JOB JNM=????????,CLASS=?,DISP=?
$$ LST DISP=?,PRI=?,CLASS=?
// JOB TBPRINT
// DLBL   DKLDIST,'VSE.YOUR-TABLEBASE.LIBRARY'
// EXTENT ,YOUR-VOLUME
// LIBDEF *,SEARCH=(DKLDIST.PRV42W51)
// DLBL MAINLIB,'YOUR-TABLEBASE-LIBRARY',,VSAM,CAT=YOUR-CATALOG
// ASSGN  SYS001,DISK,VOL=YOUR-VOLUME,SHR
// ASSGN  SYS006,SYSIPT
// ASSGN  SYS007,SYSLST
// ASSGN  SYS008,SYSLST
// ASSGN  SYS009,IGN
// EXEC   TBPRINT
(INSERT TBPRINT INPUT CARDS HERE)
/*
/&                                  /* END OF JOB                 */
* * $$ EOJ
```

Chapter 5
View Printing Utility
TBDEFPRT

Introduction

The View has a multi-purpose use in tableBASE. It is used in both the ISPF and CICS versions of the tablesONLINE applications. The utility program TBPRINT also uses the View.

TBDEFPRT allows users to selectively PRINT table definitions and Views. This documentation describes the batch operation of the utility. TBDEFPRT can also be invoked from a panel under tablesONLINE/ISPF.

Input

Control statements are input from a file with the DDname "COMMAND". Their free format syntax is described below.

Output

Reports and error messages (if any) are output to a file with a DDname of TBRPT.

General Syntax Rules for Input

The input is free format assigning values to keywords.

Each Keyword-Value combination must be followed by at least one blank or a comma.

Keyword-Value combinations may appear in any order.

An Input Sequence is a group of Keyword-Value combinations that identify printing of tables from a library.

Each Input Sequence must start on a new line and be terminated by a semicolon (;). Anything following a semicolon on a line will be ignored.

Multiple Input Sequences may be input one after the other. One Input Sequence will be processed at a time.

Within an Input Sequence some of the keywords are mutually exclusive with another keyword. (Both cannot be used in the same Input Sequence.)

Keyword Parameters

Each keyword is described in detail below:

Keyword value [,value]...;

FROMDDN
Identifies the DDNAME of the tableBASE library that contains the tables to be printed. A JCL statement must be included. Note that either the FROMDDN **or** FROMDSN keyword must be defined.

Example: FROMDDN=MAINLIB ;

FROMDSN
Identifies the Data Set Name of the tableBASE library that contains the tables to be printed. The JCL statement does not have to be included as a dynamic allocation will be done. Note that either the FROMDDN **or** FROMDSN keyword must be defined.

Example: FROMDSN=YOUR.PREFIX.TBASE.MAINLIB;

FDTTYPE
An optional parameter that identifies the type of View to be printed. FDTTYPE=ISPF is a request to print those Views that are in the "old" format. FDTTYPE=CICS is a request to print those View that are in the "new" format. If the Keyword is omitted or FDTTYPE=BOTH is specified, all Views are eligible for printing. If there is a data table with the same name as the View being printed, the data table definition will also be printed.

Example: FROMDDN=MAINLIB FDTTYPE=CICS;

SEQ
View rows will by default be printed in sequence by their field number. If the optional parameter SEQ=Y is specified, the View rows will be sequenced by field name.

Example: FROMDDN=MAINLIB FDTTYPE=CICS SEQ=Y;

SELECT
An optional parameter used to select a list of tables. To print all VIEWS, use the notation SELECT=ALL.

The table names (in conventional upper case) of the Views to be printed are enclosed in parentheses. A SELECT sequence cannot be continued over two lines. Multiple SELECT keywords can be input if all the table names to be selected for printing will not fit on a single line. Table names within parentheses can be separated by commas or spaces.

Examples:
```
FROMDDN=TBLIB1 SELECT=ALL ;
FROMDDN=TBLIB2 FDTTYPE=CICS
SELECT=CODE1;
FROMDDN=TBLIB3 SEQ=Y
SELECT=(CODE1,CODE2,CODE3) ;
```

Note: The keywords SELECT and EXCLUDE are mutually exclusive.

EXCLUDE
An optional parameter used to exclude a list of tables. All other Views will be printed. The syntax rules for EXCLUDE are the same as those for SELECT.

Example:
```
FROMDDN=TBLIB1
EXCLUDE=(CODE1,CODE2);
```

Note: The keywords SELECT and EXCLUDE are mutually exclusive.

PASSWORD
Required only if the tables to be printed are protected with read passwords. The password supplied is used for all tables to be printed.

Example:
```
FROMDDN=MAINLIB
PASSWORD=SECRET
SELECT=(CODE1,CODE2) ;
FROMDDN=MAINLIB
PASSWORD=FORGET
SELECT=CODE3 ;
```

MVS Job Control Language

An example of an execution of TBDEFPRT that prints the contents of LIB1, LIB2, and LIB3

```
//STEP1    EXEC  PGM=TBDEFPRT
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//SYSPRINT DD SYSOUT=*
//TBDUMP   DD SYSOUT=*
//TBRPT    DD SYSOUT=*
//*
//SORTMSGS DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5,1)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//LIB1     DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LIB1
//LIB2     DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LIB2
//LIB3     DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LIB3
//COMMAND  DD *
  FROMDDN=LIB1;
  FROMDDN=LIB2;
  FROMDDN=LIB3;
/*
```

VSE Job Control Language

```
$$ JOB JNM=????????,CLASS=?,DISP=?
$$ LST DISP=?,PRI=?,CLASS=?
// JOB TBDEFPRT
// LIBDEF *,SEARCH=(VSE.YOUR-TABLEBASE.LIBRARY)
// DLBL MAINLIB,'YOUR.PREFIX.MAINLIB',,VSAM,CAT=YOUR-CATALOG
// ASSGN  SYS001,DISK,VOL=YOUR-VOLUME,SHR
// ASSGN  SYS006,SYSIPT
// ASSGN  SYS007,SYSLST
// ASSGN  SYS008,SYSLST
// ASSGN  SYS009,IGN
// EXEC   TBDEFPRT
(INSERT TBDEFPRT INPUT CARDS HERE)
/*                                  /* END OF DATA FOR TBDEFPRT    */
/&                                  /* END OF JOB                  */
* * $$ EOJ
```

Chapter 6
Generating COBOL Copybooks

The utility program, TBCOBFD, generates COBOL copybooks and, optionally, the tableBASE COMMAND-AREA from views which describe the tableBASE row layout.

Coding Conventions

The table ROW-AREA and the tableBASE COMMAND-AREA are generated at the 01 level. Individual fields within these areas are generated as 05 levels.

DATE fields are divided into sub-fields (year, month, day). These sub-fields are defined at level 10; a suffix is appended to each sub-field to uniquely identify it. For example, an 'A' format DATE field (i.e, YYYMMDD) is redefined with three level 10 fields suffixed with '-YY', '-MM' and '-DD' in that order. If the DATE field name is longer than 16 characters, it is truncated to 16 characters to accommodate the suffixes in the generated level 10 sub-fields. (COBOL field names are limited to 30 characters in length.)

Generated field names are prefixed by the tableBASE View name, unless the field prefix parameter is used. In order to ensure that the generated field names are valid COBOL words embedded blanks are replaced by hyphens. The tableBASE Data table name is placed in positions 73 through 80 of each line generated.

Using TBCOBFD

You select the view for which a COBOL copybook is to be generated by means of the "PARM" field of the JCL used to invoke the program. The "PARM" field is also used to define a prefix to apply to generated COBOL data names and to generate a tableBASE COMMAND-AREA. An example of the specification of these parameters is shown below.

```
//STEP1  EXEC    PGM=TBCOBFD,REGION=256K,PARM='view,fldprfx,Y'
```

The parameters are positional. The first parameter, view, must be defined. The parameters have the following meanings:

VIEW
This first parameter defines the name of the view for which a COBOL copybook is to be generated.

FLDPRFIX
This second parameter is used to alter the field names of the generated copy books. If this parameter is not specified, the generated field names will be prefixed by the name of the View selected; otherwise, they will be prefixed by the value specified for this parameter. This parameter has a maximum length of eight characters.

COMMAND AREA
The third parameter controls the generation of the tableBASE COMMAND-AREA. It may have either of two values:

Y	to generate the tableBASE COMMAND-AREA for this view (this is the default)
N	to avoid the generation of the tableBASE COMMAND-AREA for this view

The COMMAND-AREA will be generated as a level 01 row immediately following the table ROW-AREA.

The JCL example on the following page will generate a copybook for the PAYROLL view. The generated field names will begin with the characters 'HR'. A tableBASE COMMAND-AREA will be generated.

MVS Job Control Language

```
//STEP1 EXEC    PGM=TBCOBFD,REGION=256K,PARM='PAYROLL,HR,Y'
//STEPLIB       DD      DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.USERLOAD
//              DD      DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//TBDUMP        DD      SYSOUT=*
//SYSOUT        DD      SYSOUT=*
//CBLOUT        DD      DSN=*YOUR.PREFIX*.COBOL.COPYLIB(MEMBER.NAME),DISP=SHR
//MAINLIB       DD      DSN=*YOUR.PREFIX*.TBASE.LIBRARY,DISP=SHR
/*
```

Chapter 7
Using TBCOMP

Comparing Tables - TBCOMP

This MVS batch utility, TBCOMP, compares tables and identifies any differences for your review. Tables are considered unequal if their definitions are significantly different or if the rows do not match.

Not all items in a table definition are compared. The following items are compared:

- ORG
- MTHD
- TYPE
- SMC
- ISZ
- KSZ
- KLOC
- EXP
- LDEN
- HDEN

Rows are considered unequal if any part of the row does not match.

TBCOMP is driven by statements contained in a card image input file. Two input lines are required to specify each pair of tables to be compared. Each table is defined on a separate input statement where the fields on each input statement must be given in a specific order with each field separated by a blank or a comma. The fields are:

Field	Description/Comments
id	Command identifier, may be COMPARE or CM or *
table	First table to be compared
library ddname	Ddname of library where table resides
read password	Read password, if any, of table
generation	Generation of table
view	View containing formatting information for table
unload ddname	DDname defining a flat file where the difference table(s) is to be unloaded

Notes: The command identifier must start in column 1.

An asterisk (*) in column 1 indicates a comment. TBCOMP ignores comment lines.

The mandatory fields in each statement are: id, table, library ddname.

The unload ddname must reference a sequential file or member of a PDS.

TBCOMP analyzes the differences between tables by treating the first table as the "new" table and the second table as the "old" table; i.e., it generates the differences required for making the changes to the second table. TBCOMP can report the differences in any of three formats: printed reports, flat files and/or difference tables. TBCOMP does not change any of the tables that it analyzes. You can change tables by:

- Inspecting the printed report and editing tables manually
- Using the flat files as input to TBEXEC
- Writing your own program to read the difference tables or the flat files

Using the printed reports allows you to inspect each change to ensure that it is correct for your system. This is useful for sites with customized tables.

Using the flat files is the simplest method. The first flat file contains rows which should be deleted from the target table. The second flat file contains rows which should be inserted/replaced in the target table. If rows with identical keys but different contents outside the keys are found in both the new and the old tables, then records will be created in both the first and second flat file. **Therefore, deletes must be applied before updates.** Please see the following example:

```
//   EXEC PGM=TBEXEC
//STEPLIB   DD  DISP=SHR,DSN=*YOUR TBASE.LOAD.LIB*
//NEWLIB1   DD  DISP=SHR,DSN=*YOUR.TBASE50.LIBRARY1*        <====
//TBSYSLIB  DD  DISP=SHR,DSN=*YOUR.TBASE.SYSTEM.LIBRARY*    <====
//UPDTTBL1  DD  DISP=SHR,DSN=*YOUR.PDS(MEMBER1)*            <====
//DELETBL1  DD  DISP=SHR,DSN=*YOUR.PDS(MEMBER2)*            <====
//TBMSG DD  SYSOUT=*
//TBRPT DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//TBDUMP    DD  SYSOUT=*
//CNTLCARD  DD  *
   UPDATE  LIB=NEWLIB1  TBL=TABLE1  REMOVE=DELETBL1;
   UPDATE  LIB=NEWLIB1  TBL=TABLE1  WITH=UPDTTBL1;
//
```

The difference tables are identical to the flat files except that they are stored in tableBASE libraries.

A report is always printed containing the results of each comparison. Each comparison is summarized on one line showing table and library names, generation numbers, result code for the comparison and a brief description. These result codes are:

Result Code	Meaning
0	Contents and definitions match
37	Definitions do not match (a list of mismatched definition fields is given)
100	Definitions match but contents differ
n	Other values are return codes from tableBASE, e.g., invalid password

TBCOMP will issue one of the following return codes:

Return Code	Meaning
0000	All pairs of tables are the same (with the possible exception of some acceptable parts of the definition).
0008	The contents or definitions of at least one pair differ.

Sample JCL follows on the next page.

```
//* Insert your job card here
//*******************************************************************************
//* '<====='  identifies JCL cards to be modified for your installation
//*
//STEP1 EXEC    PGM=TBCOMP,PARM='P,S,UPDT,DELE,123456,N'
//STEPLIB   DD  DISP=SHR,DSN=*YOUR.TBASE.LOADLIB*              <====
//*
//NEWLIB1   DD  DISP=SHR,DSN=*YOUR.TBASE50.LIBRARY1*           <====
//OLDLIB1   DD  DISP=SHR,DSN=*YOUR.TBASE42.LIBRARY1*           <====
//NEWLIB2   DD  DISP=SHR,DSN=*YOUR.TBASE50.LIBRARY2*           <====
//OLDLIB2   DD  DISP=SHR,DSN=*YOUR.TBASE42.LIBRARY2*           <====
//TBSYSLIB  DD  DISP=SHR,DSN=*YOUR.TBASE.SYSTEM.LIBRARY*       <====
//UPDTTBL1  DD  DISP=SHR,DSN=*YOUR.PDS(MEMBER1)*               <====
//DELETBL1  DD  DISP=SHR,DSN=*YOUR.PDS(MEMBER2)*               <====
//TBCMPRPT  DD  SYSOUT=*
//TBCMPLST  DD  SYSOUT=*
//TBREPORT  DD  SYSOUT=*
//TBLOG     DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//TBDUMP    DD  SYSOUT=*
//TBCMPCMD  DD  *
* EXAMPLE 1
COMPARE,TABLE1,NEWLIB1,RPSWD1,GEN1,VIEW1,UPDTTBL1
CM,TABLE1,OLDLIB1,RPSWD2,GEN2,VIEW2,DELETBL1
* EXAMPLE 2
CM,TABLE2,NEWLIB1
CM,TABLE3,OLDLIB1
* EXAMPLE 3
CM,TABLE4,NEWLIB2,,,VIEW5
CM,TABLE4,OLDLIB2,,,VIEW6
//*
```

The EXEC card PARM values are:

- P is a flag indicating whether to print the differences (value P or Y) or not (any other value). Optional. Default is to not print differences.
- S is a flag indicating whether to store the difference tables (value S or Y) or not (any other value). Optional. Default is to not store the difference tables.
- UPDT and DELE are prefixes for difference table names when difference tables are to be stored. TBCOMP will build the names of the difference tables by appending as many of the last non-blank characters in the names of the tables being compared as possible to these prefixes. (Table names may not exceed eight characters). Optional. Defaults are UPDT and DELE.
- nnnnnn is the master password. Optional. No default.
- N is a flag indicating that the data content comparison is Not case sensitive; upper case or lowercase is ignored in the test for equality. The default value is Y; the comparison is case sensitive.

The file specified by TBCMPRPT contains a summary of the differences found.

TBCOMP provides an optional parameter whereby you may define a view which contains the formatting information for this table. If you do not specify this view name, the list of differing rows will be contained in the file specified by TBCMPLST. If the view name is specified, the file specified by TBREPORT will

contain the list of differing rows and the file specified by TBLOG will contain a summary of the tables.

The sample job compares three pairs of tables.

Example 1 compares specific generations of TABLE1. Any differences will be printed using the formatting information in the corresponding views. The read passwords in this case are redundant since the master password is supplied on the EXEC card. The difference tables are unloaded to members of *YOUR.PDS*. Difference tables UPDTBLE1 and DELEBLE1 will be stored on NEWLIB1.

Example 2 compares the latest generation of TABLE2 and TABLE3. The master password will be used to open the tables if they are read-protected. Any differences will be printed without formatting. Difference tables UPDTBLE2 and DELEBLE3 will be stored on NEWLIB1. The difference tables are not unloaded to flat files.

Example 3 compares the latest generations of TABLE4. The master password will be used to open the tables if they are read-protected. Any differences will be printed using the formatting information in the corresponding views. Difference tables UPDTBLE4 and DELEBLE4 will be stored on NEWLIB2. The difference tables are not unloaded to flat files.

Appendix A
User Contributed Library

In this appendix you will find descriptions of programs submitted by other tableBASE users. These users have found these programs beneficial to them and wish to share them with other tableBASE users. The programs will be found in the User Contributed Library on the installation tape.

These programs are being made available on an "as is" basis. They are not supported by Data Kinetics and are not included in the licensed software.

Monitoring Library Space - LIBCNTL

This MVS utility, LIBCNTL, monitors the space available in tableBASE libraries. If all libraries being monitored contain sufficient unused space, the program will terminate without incident. If one or more libraries do not contain sufficient unused space, the program will log the results of its monitoring and then abend. This provides an appropriate alert to the operations staff that a potential problem exists and enables them to take corrective action prior to an actual production failure.

This program uses a control table, LIBCNTL, which is a normal tableBASE table with the following row structure:

Position	Contents	Format	Comments
1	DSName of library to be monitored	X	Entered by tableBASE administrator
45	Amount of space to be kept available for use	N	Entered by tableBASE administrator
49	Amount of space currently allocated	N	Entered by LIBCNTL
53	Amount of space currently free	N	Entered by LIBCNTL
57	Date examined	D	Entered by LIBCNTL
63	Attention indicator	X	Entered by LIBCNTL as an asterisk

Sample JCL

```
//*JOBNAME  JOB  ** YOUR JOB CARD **
//****
//*     MODIFY '*YOUR.PREFIX*' FOR YOUR INSTALLATION
//*
//* ===>    NOTE: THIS JOB REQUIRES TBL=LIBCNTL IN DDNAME=MAINLIB
//*
//****
//STEP1   EXEC  PGM=LIBCNTL,REGION=250K
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.USERLOAD
//         DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//MAINLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.MAINLIB
//TBDUMP   DD DUMMY
//SYSUDUMP DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
```

Creating Views Programmatically - TBCREFDT

tablesONLINE provides facilities to create Views via a controlled data entry process. TBCREFDT enables you to create Views programmatically, without the need for manual entry.

This MVS utility reads field descriptions from a card image input file with a DDNAME of SYSIN. Field descriptions have the following format:

Positions	Contents	Comments
1 - 8	table name	The name of the table to which his View applies
9 - 28	field name	Optional
29 - 29	key field indicator	Enter Y to denote a key field. Note that key fields must be contiguous
30 - 34	display length	The length of the field when printed by TBPRINT
35 - 35	display format	The format of the field when printed by TBPRINT
36 - 40	storage length	
41 - 41	storage format	
42 - 80	unused	

There are a fixed number of valid combinations of display and table formats:

Display Format	Displayed As	Table Format	Stored As
X	text	X/U	text
Y	hexadecimal	X	text
N	numeric (leading -)	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	MDDYY/MMDDYYYY	B	edited and
C	DMMYY/DDMMYYYY	C	converted to
D	YDDD/YYYYDDD	D	storage
E	DDMMMYYY/DDMMMYYY	E	format

For the following display formats, the table formats are set to these default values:

Display Format Entered	Default Storage Format	Default Rule
X	U	Character move
Y	X	Hex digits displayed from characters
N,0,1,2,..9	N	Numbers stored as zoned decimal
A	A	Dates displayed in table format;
B	B	If lengths are not entered, they
C	C	Will default to the shorter lengths
D	D	(e.g., year represented as YY)
E	E	

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

The final relationship between display format and table format is in the matter of length:

Display Format	Display Length	Storage Format	Storage Length
X	1 - 50	X	same as display
Y	2 - 50 (even lengths)	X	1 - 25
N	1 - 16 (inc. sign)	N	display digits
0	1 - 16 (no sign)	P	(display digits + 1)/2
1	1 - 17 (inc. sign and decimal)	F	4 (display length limited to 9 digits)
:			
9	1 - 17 (inc. sign and decimal)	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

When several codes reference the same underlying type, it is possible to see the same data in different ways by changing the display format. It is possible, for example, to show US users dates in format B while Canadian users get format C.

Notes:

1. Views are created on the tableBASE library allocated to MAINLIB.
2. Existing Views are replaced.
3. The edit table, TBOLHKFM, from tablesONLINE is required and is allocated to DDNAME=TBSYSLB. For installations without tablesONLINE, this table is provided as part of the install procedure.

Sample JCL

```
//*     Insert your job card here
//*******************************************************************************
//*     <=====
//*     The above symbol identifies JCL cards to be modified for your installation
//*
//STEP1 EXEC    PGM=T   TBCREFDT

//STEPLIB       DD      DSN=*YOUR.TBASE.LOADLIB*,DISP=SHR              <====
//TBDUMP        DD      SYSOUT=*
//TBRPT         DD      SYSOUT=*
//MAINLIB       DD      DSN=*YOUR.TBASE.LIBRARY*,DISP=SHR                    <====
//TBSYSLIB      DD      DSN=*YOUR.TBASE.SYSTEM.LIBRARY*,DISP=SHR       <====
//*
//*     This is sample input
//*             Note:   Table Test3 has invalid input
//*                     Display length and storage length contain 55
                //*
//SYSIN         DD      *
TEST1   DEPARTMENT CODE Y00005N00005N
TEST1   SECTION CODE            Y00004N00004N
        TYPE                    N00001X
        DESCRIPTION             N00010X000010X
TEST2   SOCIAL SECURITY NO.     Y00009N
        LAST NAME               N00010X00010X
        FIRST NAME              N00010X00010X
        DATE OF BIRTH           N           A
TEST3   S.I.N.                  Y00009N
        LAST NAME               N00055X00055X
        FIRST NAME              N00010X00010X
        DATE OF BIRTH           N           A
        SALARY                  N            Z
/*
```

Index

R