tableBASE

Batch Utilities Guide

Release 6.0.3 /

Release 6.1.0

DataKinetics®

Document Number: TBM004-R6.0.36.1.0v2.0

Publication History

Maintenance Release 6.0.1—May 2004

Service Pack 2 Release 6.0.1 Version 1.1—November 2004

tableBASE Release 6.0.2 Version 1.0—May 2005
tableBASE Release 6.0.2 Version 1.1—April 2006
Service Pack 6 Release 6.0.2 Version 1.2—August 2006
tableBASE Release 6.0.3 Preliminary—January 2009
tableBASE Release 6.0.3 Version 1.0—June 2008

tableBASE Release 6.0.3 / Release 6.1.0 Version 2.0 - March 2010

Technical Support Hotline: (613) 523-5588

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

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

Facsimile: (613) 523-5533
Email: tableBASE@dkl.com
http://www.dkl.com

Table of contents

Preface

This guide describes tableBASE batch utility programs that are used to define and maintain tables and libraries, report on their contents, and generate COBOL copybooks or C structures from table definitions.

Who this guide is for

This guide is intended for:

- analysts who define tables
- programmers who maintain and test tables and generate COBOL copybooks or C structures
- systems personnel responsible for maintaining tables and transferring them from test to production

What you should know to use this guide

Familiarity with tableBASE concepts and facilities, the tableBASE application interface, tableBASE programming commands, and the batch environment aid in the understanding of this document.

Conventions used in this Guide

Convention	Description
`code examples and commands`	Code examples and commands appear in this type of font: `this is an example of the font.`
MAXNMTAB	Names of parameters appear in upper case simply for ease of reading; actual case used is upper or lower or a mixture.
Version	Following IBM standards, the term *version* refers to a generation of a software product that has significant new code or new functionality. *Version* is a more general term than *release*. For example, *Version 6* includes *Release 6.1* and *Release 6.2,* and is equivalent to *Release 6.x*.
Release	Following IBM standards, the term *release* refers to a program or set of programs which represent a specific revision to the base version of a software product. For example, *Release 6.0* is a term that is used to identify the first release of *Version 6*. Subsequent releases made available under the Version 6 umbrella, such as *Release 6.1*, will provide additional revisions to the base product.
Modification Level	Following IBM standards, the term *modification level* refers to the application of specific program enhancements and error corrections to the release of a software product. For example, *Release 6.0.3* is at *modification level* 3, and *Release 6.1.0* is at *modification level* 0.
MVS	MVS is a generic term which is used when referring to z/OS and other related IBM operating systems.

Glossary

The following terms are used throughout this guide. Descriptions are provided to make them more familiar.

Data Table	A Data Table is the actual raw data. Each Data Table has a table definition (DT-BLOCK) that is used to generate the Index for the Data Table.
Index	An Index is defined for each Data Table. A Data Table Index is generated dynamically when a table is opened or defined based on the information in the table definition (DT-BLOCK).
Alternate Index	An Alternate Index is an Index that may be defined for a Data Table. The Alternate Index has an Alternate Index definition (ALT-DEFINITION) that defines the key, organization, and search order. Alternate Indexes are optional, and there is no limit to the number of Alternate Indexes a Data Table may have.
Delivered defaults	The defaults that are delivered with the product. Also known as *factory defaults*.
Installation defaults	The defaults set at installation time by an administrator, which may or may not be the same as the delivered defaults. Defined using the TBOPTGEN file. (These defaults may be overridden by an individual application using the TBOPT file.
TSR	Table Space Region. A data space of up to 2 Gigabyte is used by tableBASE to house tables. The data space is owned by an application in the associated address space. The application uses tableBASE to access data within the tables.
Local TSR	As above.
Shared TSR	A VTS-TSR; see below.
Temporary Table	A temporary table exists only within a TSR, and is created by the DT command (or IA). It is never stored in a library. A temporary table can be distinguished from a library table using the GD command output—if found, a temporary table will show no dataset name.

Linked Table	A linked table (also known as a remote table) is created when a user issues a command to open a table that is already open in a VTS-TSR specified in the LIB-LIST. The table entry in the local TSR is linked to the existing open table in the VTS-TSR. No updates are allowed to a linked table.
Table Expansion	Dynamic allocation of space for tables in the TSR when the initial space allocated becomes insufficient.
Multitasking Batch	An MVS region that implements multitasking by attaching multiple TCBs. This can include a batch job that attaches several subtasks or a transaction processing region like DB2 stored procedures that implements multitasking through multiple TCBs.
View	A tablesONLINE view provides the field, edit and display attributes for a Data Table with its Index. In releases previous to Version 6 (and Version 5) the View was referred to as a Field Definition Table (FDT).
Alternate Index View	A tablesONLINE Alternate Index View is identical to a View but applies to a Data Table when access is through an Alternate Index.
VTS	Virtual Table Share. A VTS-TSR is a shared Data Space that provides a public location to share tables among application regions.
tableBASE VTS	The optional component that provides the shared VTS-TSR capability.
VTS-TSR	A Virtual Table Share (VTS) Table Space Region (TSR) is a shared TSR, and resides in a shared data space. Applications can access tables within a VTS-TSR, and use the information as if it were within their local TSRs. VTS TSRs are managed by the VTS Agent program. If tableBASE Process Manager is installed, VTS TSRs are managed by VTS Managers, and the VTS Agent is not required but still available for transition purposes.
tableBASE Process Manager	The optional component that extends the functionality of the tableBASE VTS shared TSR capabilities.
TPM	tableBASE Process Manager—the tableBASE Process Manager top tier component.
VTS Manager	The middle tier of tableBASE Process Manager which manages VTS-TSRs.
TPVM	This is synonymous to a VTS Manager. It is the middle tier of tableBASE Process Manager which manages VTS-TSRs.

Catalog	A catalog contains definitions of managed items in the next tier down in the hierarchy; i.e., TPVM definition information is contained in the TPM catalog, and VTS-TSR definition information is contained in the TPVM catalog. The catalog is contained within the LDS associated with the TPM / TPVM.
Cataloged VTS	A VTS-TSR that is managed by a user-defined VTS Manager under tableBASE Process Manager, as opposed to the *compat* VTS Manager.
LDS	Linear DataSet used for tableBASE Process Manager.
Alias name	An alternate name for a VTS-TSR that can be created, assigned, and used in lieu of the name assigned at VTS-TSR definition.
VTS switch	The action of switching an alias name from one VTS-TSR to another. This is a feature of tableBASE Process Manager.
***compat* VTS Manager**	The *compat* VTS Manager is the default VTS Manager that runs under tableBASE Process Manager.

In previous releases, tableBASE allowed two table types: True and Pointer. True tables were characterized as not having an Index and by being stored in contiguous memory. In Version 6, the concept of True tables still exists, however they are treated as Pointer tables, as all memory is segmented memory that requires Indexes.

What is covered in this guide

This guide identifies and describes each of the eight batch utilities that come with tableBASE.

Chapter 1 gives an overview of the tableBASE batch utility programs.

Chapter 2 discusses TBEXEC (DK1TEXEC), the main batch utility for tableBASE. The maintenance functions provided by TBEXEC are identified and described, JCL examples and report samples are shown, and a list of associated error messages is provided.

Chapter 3 describes the use of TBEXEC. It includes the Job Control Language (JCL) required to run TBEXEC, followed by descriptions of how TBEXEC handles errors. Standard reports produced are described, as are error and audit messages issued by TBEXEC.

Chapter 4 describes the use of TPDRIVER. It includes the JCL required to implement commands.

Chapter 5 describes TBPRINT (DK1TPTBL), the utility that prints formatted columnar listings using information from Views.

Chapter 6 describes the utility TBDEFPRT (DK1TVWPR) that prints Views of tableBASE tables.

Chapter 7 describes the utilities TBCOBFD (DK1TCOBF) and its companion DK1TCSTR that create API interfaces for tables.

Chapter 8 describes the table compare utility TBCOMP (DK1TCOMP).

Chapter 9 points to the library conversion utility DK1TCNV and its companion library version identification utility DK1TLCHK.

Naming protocol

Version 6 features the new tableBASE naming protocol.

All tableBASE executables begin with DK1 for easy identification, a prefix that has been reserved with IBM for exclusive use by DataKinetics.

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

What is new in version 6

Version 6 introduces the ability to create C structures from tablesONLINE Views.

Additional tableBASE references

This guide is one of a series that describes tableBASE and tablesONLINE:

- *tableBASE Release Notes*
- *tableBASE Installation Guide*
- *tableBASE Concepts and Facilities Guide*
- *tableBASE Administration Guide*
- *tableBASE Programming Guide*
- *tableBASE Quick Reference Guide*
- *tablesONLINE/CICS User's Guide*
- *tablesONLINE/ISPF User's Guide*
- *tableBASE Library Bridge Manual, Release 5.B*

1 Introduction

The Batch Utility Programs

The utility programs supplied with tableBASE simplify the maintenance of the tableBASE environment.

1. TBEXEC (DK1TEXEC) is the primary tableBASE batch utility. It provides basic housekeeping facilities such as initializing new table libraries, defining tables, deleting tables, making mass updates to tables, and copying tables among libraries. TBEXEC also provides a print facility designed for the printing of tables used in testing.

2. TPDRIVER is the tableBASE Process Manager batch utility. It provides VTS-TSR management features such as defining VTS-TSRs, assigning alias names, auto-starting VTS-TSRs, switching VTS-TSRs, and more.

3. TBPRINT (DK1TPTBL) produces columnar listings of Data Tables, based on formatting information contained in the View.

Note: Views are a facility of tablesONLINE, an optional tableBASE component.

4. TBDEFPRT (DK1TVWPR) prints Views and field definitions.

5. TBCOBFD (DK1TCOBF) generates COBOL data description copybooks for use in user-written tableBASE applications.

6. DK1TCSTR generates C structures for use in user-written tableBASE applications.

7. TBCOMP (DK1TCOMP) compares tables and identifies any differences for examination.

8. DK1TCNV is used to convert tableBASE libraries between Version 5 (V5), Library Bridge (Bridge), Version 6 Transition (V6TRANS), and Version 6 (V6).

9. DK1TLCHK is used to identify the version of a tableBASE library.

2 TBEXEC (DK1TEXEC) commands

In this chapter, a brief summary of each command is presented, followed by an alphabetical listing of the parameters used in the TBEXEC commands. The chapter concludes with a detailed definition of each command.

The TBEXEC (DK1TEXEC) program is a tableBASE batch utility program that allows users to perform a variety of maintenance functions for tableBASE libraries and tables. This chapter identifies and describes each of the available commands and provides examples of how they are used.

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

The TBEXEC program 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 its table organization. Defining, copying, loading, unloading, and clearing a table are functions easily performed by TBEXEC using keyword commands.

Note: TBEXEC does not work with or use VTS-TSRs, and it cannot be run with the multi-tasking switch (MULTITASKING=Y) turned on.

If passwords are forgotten, TBEXEC can make a date-variable Master Password facility available that allows access to tableBASE tables. DataKinetics Ltd. (DKL) has supplied details of the operation of the Master Password facility to your tableBASE administrator in the *tableBASE Administration Guide*.

TBEXEC accepts only DDnames that are alphanumeric, and conform to standard IBM naming conventions.

Command summary

Table 2-1 is a summary of the commands available in TBEXEC, listed in alphabetical order. Subsequent sections provide details of their use.

Note: Any input statement starting with an asterisk (*) in column one is treated as a comment.

Table 2-1: Command summary

Command	Command description	Options
CHANGE	change table characteristics	Option 1: Change the definition of a table Option 2: Change an Alternate Index definition
CLEAR	clear or empty a table	-
COPY	copy a table, table generation, or a library	Option 1: Copy a table generation Option 2: Copy a table (all generations) Option 3: Copy a library
DEFINE	initialize a new library, define a new table, define an Alternate Index.	Option 1: Initialize a new tableBASE library Option 2: Define a new table Option 3: Create an Alternate Index definition
DELETE	delete one or more table generations	Option 1: Delete a table generation Option 2: Delete all generations of a table
EXPAND	increase size of a tableBASE library	-

Table 2-1: Command summary (Continued)

Command	Command description	Options
EXPORT	convert a tableBASE library or table to a sequential dataset	Option 1: Convert entire table library Option 2: Convert a table to a sequential dataset
IMPORT	copy an exported sequential dataset into a tableBASE library	-
LOAD	load a table generation from a sequential dataset	-
PRINT	prints information about a directory, table generation, and/or table contents	Option 1: List Directory Option 2: Print Table Information Option 3: Print Table Contents
RENAME	rename table generations	-
SET	establishes default values for keywords	-
UNLOAD	write a table generation to an output dataset	-
UPDATE	edit a table from an input dataset	-

Notes:

i. TBEXEC commands are specified in control statements read by TBEXEC from a file identified by the DDNAME CNTLCARD.

ii. A TBEXEC command sequence consists of a command and a list of parameters specified by keyword and value combinations terminated by a semicolon (;).

iii. 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 to TBEXEC:

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 alphanumeric characters 0-9 and uppercase 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 required, others are optional. The conventions 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
- parentheses ( ) are used to designate optional parameters and are not coded on the command

Parameter descriptions

Most of the commands used in TBEXEC require parameters. Table 2-2 lists these parameters in alphabetical order.

Table 2-2: Parameter descriptions

Parameter	Description
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 Index.
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. The default is the row size.
DEF	The name of the table 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, which is explained in greater detail the *tableBASE Programming Guide*.
EXPFACTR	An integer between 1-999 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 field is described in the *tableBASE Programming 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 • positive, it refers to an absolute generation number • negative, it refers 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 P. With Version 6, all tables are Pointer tables. The value T is retained for backwards compatibility.

Table 2-2: Parameter descriptions (Continued)

Parameter	Description
ISZ	A number between 1-32767 that identifies the length of the row. The default value is 1. ISZ is a synonym for RSZ.
KSZ	A number between 1-256 that identifies the length of the key. The default value is 1.
KLOC	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 connected to a tableBASE library. Unless changed by the SET command, the default is MAINLIB.
LINSIZ	A number between 1-100 that represents the number of bytes of table data to be printed per line. If omitted, or if 0 is entered, the default is 100 or the value specified on the most recent SET command.
LOWERDEN	An integer between 1-999 representing, in the thousands, the lowest density a Hash table may have. A lower density of 700, for example, will be treated as 70%. Density is discussed in the *tableBASE Programming Guide*.
MAXGEN	A number between 1-9 that represents the number of generations to be kept for the table. The default value is 1.
MTHD	A one-character code indicating the table search method to be used: S (Serial), Q (Queued Sequential), B (Binary), C (Address Tree Binary), and H (Hash). The default value is S or depends on the ORG parameter setting; see below.
NEWID	This COPY command parameter indicates whether the update history of the table should change when a table is copied (default is YES). Alternate Indexes are always copied as NEWID=YES, as the setting of NEWID is ignored. If NEWID is NO, the userid and timestamp are not updated when the table is copied.
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), or H (Hash). The default is R.

Table 2-2: Parameter descriptions (Continued)

Parameter	Description
PRNT	A one-character code indicating the format to be used to print the table contents: C (Character), H (Hexadecimal), or 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 8 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.
RSZ	A number between 1-32767 that identifies the length of the row. The default value is 1. RSZ is a synonym for ISZ.
STARTBYT	A number that 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 that 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. Version 6 supports only R (Resident). Since this is the default value, there is no need to use this keyword, and is only provided for backwards compatibility.
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 copied.

Table 2-2: Parameter descriptions (Continued)

Parameter	Description
TTYPE	This parameter can be used to specify 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 may have a View named aBC. If TTYPE is blank the table name refers to a Data Table name. If TTYPE is V (or F), the table name refers to a View name. F is supported for backwards compatibility. When TTYPE is specified with commands using the parameters TBL, DEF, and NEWNAME, it will cause each name to be converted.
UPPERDEN	An integer between 1-999 representing, in the thousands, 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 Programming Guide*.
WITH	Indicates the DDNAME of the dataset from which data is to be retrieved for the purpose of updating a table.
WPSWD	Write password, if write password protection is desired. If not specified on the DEFINE command, 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. For further details on TBLBASE, see the *tableBASE Programming Guide*.

Option 1: Change the definition of a table

Option 1 (see Table 2-1 on page 18) is indicated by the keyword TBL. Changes made to the definition can cause the contents to be physically reorganized. If the table named is an Alternate Index, changes that affect the Alternate Index definition are made to the Alternate Index definition; changes that do not affect the Alternate Index definition are made to the Data Table.

CHANGE

TBL=(table name) **Required keyword**
TTYPE=(table name conversion - blank/V)
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) **all tables are P internally in Version 6**
SMC=(storage mode code - R) **not required in Version 6**
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 a table's password and write password match, the generation indicated will be opened and changed according to the specified keywords. It is then 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=*.

Option 2: change an Alternate Index definition

Option 2 (see Table 2-1 on page 18) is indicated by the keyword ALT. The definition specified must be an Alternate Index definition.

CHANGE

ALT=(alternate index name) **Required keyword**
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 data table name);

Action

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

CHANGE command use

The following two examples show two different ways of using of the CHANGE command.

Example 1—change table organization from Sequential to Hash

This change will be used to see whether Hash searching is more efficient than Binary searching. 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 Data Table

Change the name of the Data Table for the Alternate Index definition DISTRICT.REGION is the new contents table name. The alternate table is in the tableBASE library identified by the DDNAME MAINLIB.

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


CLEAR command

The CLEAR command empties a table of all its rows, leaving it with zero table rows.

CLEAR

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

Action

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

CLEAR command use

The following shows one use of the CLEAR command.

Example 1—delete all rows in latest generation

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.

Option 1: copy a table generation

This option (see Table 2-1 on page 18) of the COPY command copies one generation of a table from one tableBASE library to another tableBASE library. This option is indicated by the keyword TBL and by the absence of the ALLGEN keyword.

COPY

TBL=(table name) **Required keyword**
TTYPE=(table name conversion - blank/V)
TO=(destination library) **Required keyword**
LIB=(library name)
GEN=(generation number)

PSWD=(read password/write password)
NEWNAME=(table name)
NEWID=(YES/NO) Default is YES
REPLACE=(YES/NO) Default is NO
WPSWD=(new write password/*);

Notes:

i. The parameter TTYPE=V is supplied when copying a View. This will ensure that the View name is used for both the TBL and NEWNAME parameters.

ii. NEWID applies only to data tables; it does not apply to alternates.

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 is YES.

When the NEWID parameter is YES (default), all table characteristics will be copied to the new table along with the contents, except for the date, time, and user ID of the last change. The new table definition will be updated with the date, time, and user ID of the copy operation. When the NEWID parameter is NO, all table characteristics including the original date, time and user ID of last change will be copied along with the contents. This permits a table to be restored with original update history information preserved.

Note: The existing write password can be removed by setting WPSWD to *.

Option 2: copy a table (all generations)

This option (see Table 2-1 on page 18) of the COPY command copies all generations of a table from one tableBASE library to another tableBASE library. This option is indicated by the keyword TBL and ALLGEN=YES.

COPY

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

Notes:

i. The parameter TTYPE=V is supplied when copying a View. This will ensure that the View name is used for both the TBL and NEWNAME parameters.

ii. NEWID applies only to data tables; it does not apply to alternates.

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 is YES.

When the NEWID parameter is YES (default), all table characteristics will be copied to the new table along with the contents, except for the date, time, and user ID of the last change. The new table definition will be updated with the date, time, and user ID of the copy operation. When the NEWID parameter is NO, all table characteristics including the original date, time and user ID of last change will be copied along with the contents. This permits a table to be restored with original update history information preserved.

Note: The existing write password can be removed by setting WPSWD to *.

Option 3: copy a library

This option (see Table 2-1 on page 18) of the COPY command copies one tableBASE library to another.

COPY

FROM=(library #1)	**Required keyword**
TO=(library #2);	**Required keyword**
NEWID=(YES/NO)	Default is YES
REPLACE=(YES/NO)	Default is NO

Note: NEWID applies only to data tables; it does not apply to alternates.

Action

Library is the DDNAME 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.

When the NEWID parameter is YES (default), all table characteristics will be copied to the new tables along with the contents, except for the date, time, and user ID of the last change. The new table definitions will be updated with the date, time, and user ID of the copy operation. When the NEWID parameter is NO, all table characteristics including

the original date, time and user ID of last change will be copied along with the contents. This permits a table to be restored with original update history information preserved.

COPY command use

The following are four examples of using the COPY command:

Example 1—copy the latest generation

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=V TO=PRODLIB REPLACE=YES;
```

Example 2—create a copy of a table

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

Copy all generations of 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 MAINLIB to DDNAME BACKUP

Copy the tableBASE library MAINLIB to a library identified by the DDNAME BACKUP. This must be a tableBASE library. 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 Index.

Option 1: initialize a new tableBASE library

This option (see Table 2-1 on page 18) is indicated by 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 (see Table 2-1 on page 18) is indicated by the keyword TBL. The library in which this table is to be defined must already exist.

DEFINE

TBL=(table name) **Required keyword**
TTYPE=(table name conversion - blank/V)
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)
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 in the named library.

Option 3: create an Alternate Index definition

This option (see Table 2-1 on page 18) is indicated by the keyword ALT and the absence of the keywords NEWLIB and TBL.

DEFINE

ALT=(alternate index name)	**Required keyword**
BASENAME=(data table name)	**Required keyword**
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 Index definition on the library specified.

DEFINE command use

The following examples show all of the ways to use the DEFINE command:

Example 1—initialize a tableBASE library

Initialize a tableBASE library identified by the DDNAME PRODLIB.

```
DEFINE NEWLIB=PRODLIB;
```

Example 2—define a new table

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 Index

Create an Alternate Index definition called EMPNAME for the Data Table EMPNUM. The Alternate Index will use a 40 byte key starting in position 18. The Alternate Index 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 one or more specified generations of a table.

Option 1: delete a single table generation

This option (see Table 2-1 on page 18) is indicated by the absence of ALLGEN=YES.

DELETE

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

Action

The specified generation will be deleted.

Option 2: delete all generations of a table

This option (see Table 2-1 on page 18) is indicated by ALLGEN=YES.

DELETE

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

Action

All generations of the table will be deleted.

DELETE command use

Example 1—delete table generation

Delete generation -1 (one generation 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 table and view

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=V;

Note: Views are a function of tablesONLINE, an optional tableBASE component.

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 tableBASE library size 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. 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 a change to the date/time stamp.

The target library must be initialized and emptied by using, for example, the TBEXEC DEFINE library command.

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

The default value for LIB is MAINLIB, unless overridden by a SET command.

In Version 6 the EXPAND command can increase the size of the NEWLIB's internal free-space directory. In previous versions, the EXPAND command could not increase the size of the NEWLIB's internal free-space directory.

Note: Once converted, Version 6 format libraries may be expanded. However, expansion will mean that they can no longer be converted back to the Release 5.x format.

Note: tableBASE Version 6 libraries are not compatible with previous releases. For more information see Appendix B of the *tableBASE Installation Guide*.

EXPAND command use

The following example shows how to use the EXPAND command.

Example 1—expand a library

Expand tableBASE library MAINLIB into a library identified by the DDNAME LARGELIB. The target library has been previously initialized and is empty by having used, for example, the TBEXEC DEFINE Library command.

EXPAND NEWLIB=LARGELIB;

EXPORT command

The EXPORT command converts either individual tables or an entire tableBASE library to a sequential dataset. It provides a mechanism to synchronize tableBASE tables and libraries residing on different platforms.

Option 1: export an entire library

EXPORT

LIB=(library name) **Required keyword**
TO=(DDNAME) **Required keyword**
PSWD=(read password/write password);

Action

The library specified by the DDNAME defined in the LIB keyword field will be copied to the file defined by the DDNAME specified in the TO keyword field. The output dataset

specified in the TO keyword should be defined as a sequential dataset (DSORG=PS) containing fixed-length records that are 512 bytes in length. The command also lists the number of tables exported. If any tables are password protected, the PSWD keyword with the appropriate password must be used. The PSWD keyword with the master password will allow all password protected tables to be exported.

If multiple libraries are to be exported, a different TO=(DDNAME) must be used for each library.

Option 2: export a specific table

EXPORT

TBL=(table name) **Required keyword**
TTYPE=(tablename conversion—blank/V)
LIB=(library name) **Required keyword**
TO=(DDNAME) **Required keyword**
PSWD=(read password/write password);

Action

The latest generation of the table will be copied to the file defined in the DDNAME specified in the TO keyword field. The output dataset specified in the TO keyword should be defined as a sequential dataset (DSORG=PS) containing fixed-length records that are 512 bytes in length.

If multiple tables are to be exported to the same DDNAME, the EXPORT by Table control cards MUST be grouped together. The DDNAME specified will be closed when a different DDNAME is specified in the TO keyword field on any subsequent EXPORT command and at the end of the TBEXEC processing for all commands. Thus if an EXPORT to a previously referenced DDNAME occurs tableBASE will re-open the output file and start from the beginning. Any data that was put in the file previously will be overwritten.

If the table is password protected, the PSWD keyword with the appropriate password must be used. The PSWD keyword with the master password will allow all password protected tables to be exported.

EXPORT command use

The following example illustrates the use of the EXPORT command:

Example 1—export a tableBASE library to a sequential dataset

EXPORT LIB=PRODLIB TO=DOWNLOAD;

Example 2—export a table to a sequential dataset

EXPORT LIB=PRODLIB TBL=TABLE01 TO=DOWNLOAD;

IMPORT command

The IMPORT command copies a sequential dataset created by the EXPORT command to a tableBASE library. Used in conjunction with the EXPORT command, the IMPORT command provides a mechanism to synchronize tableBASE tables and libraries residing on different platforms.

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 (DSORG=PS) 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 also lists the number of tables imported. Table passwords in the originating library are retained.

IMPORT command use

The following example illustrates the use of the IMPORT command:

Example 1—import a sequential dataset

IMPORT LIB=PRODLIB FROM=UPLOAD REPLACE=NO;

LOAD command

The LOAD command loads a table from a sequential dataset (DSORG=PS) into a tableBASE library, creating a new generation of the table. To perform this function, the table must already exist.

LOAD

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

Action

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

The UNLOAD command copies a table to a sequential dataset.

LOAD command use

The following example illustrates the use of the LOAD command:

Example 1—load a sequential dataset into a tableBASE library

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 information about the directory, a table generation, and/or the contents of a particular generation of a specified table.

Option 1: list directory

The List Directory option (see Table 2-1 on page 18) 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

DETAIL=(YES/NO) Default is NO
LIB=(library name);

Action

The PRINT command provides the following information for each table in the directory of the named tableBASE library:

- name
- table type
- absolute generation number of each generation present
- date and time that each generation was created
- library dataset name
- amount of space allocated/remaining

When the DETAIL parameter is YES, the following information for each table in the directory of the named tableBASE library will also be provided:

- table organization
- table search method
- index code
- storage mode code (obsolete, printed for reference)
- number of rows in the table
- row size
- user id that last stored the table

The library Dataset Name is also printed, along with the amount of space allocated to the library and the amount of space that remains for use.

Option 2: print table information

The Print Table Information option (Table 2-1 on page 18) of the PRINT command lists a table definition. This option is indicated by the keyword DEF.

PRINT

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

Note: PRINT DEF will print the definition if the table has a READ password; PRINT TBL will not print anything for a password-protected table, unless the password is supplied on the PRINT command.

Action

The following table information is printed for each generation of the named table that resides in the specified tableBASE library:

- absolute generation number
- the maximum number of generations to be retained
- number of generations existing at present
- relative generation number
- date and time created
- number of rows in the table
- number of rows allocated in the table
- table organization
- table search method
- index code
- storage mode code (obsolete, printed for reference)
- level of password protection

  Note: READ/WRITE indicates password security; it does not identify the type of password security (RO, R/W); NONE indicates no password security. (See example in Figure 3-5 on page 56)
- row size
- key length
- key location
- expansion factor
- upper density

- lower density
- actual density of the table
- View version
- user id that last stored the table
- View name of an associated View

Note: Views are a function of tablesONLINE, an optional tableBASE component.

Option 3: print table contents

The Print Table Contents option (see Table 2-1 on page 18) is indicated by the keyword TBL.

PRINT

TBL=(table name) **Required keyword**
TTYPE=(table name conversion - blank/V)
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, the Print Table Contents option lists the contents of the table in character (C), hexadecimal (H), or both (B) modes depending on the value of the PRNT keyword selected. STARTREC and RECCOUNT limit the portion of each table printed. The definition information for the specified table generation will be output in a format similar to that described in option 2, above.

PRINT command use

The following examples illustrate the use of the PRINT command:

Example 1—list the directory of a tableBASE library

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

```
PRINT LIB=MAINLIB;
or
PRINT;
```

Example 2—list the definition of a table

List the definition of TABLE01 on tableBASE library MAINLIB.

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

Example 3—list table contents

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 LINSIZ=80;
```

RENAME command

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

RENAME

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

Action

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

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

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

RENAME command use

The following example illustrates the use of the RENAME command:

Example 1—Rename Table Generations

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=V NEWNAME=TABLE02;
```

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.

SET command use

The following example illustrates the use of the SET command:

Example 1—set default values

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 a sequential output dataset.

UNLOAD

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

Action

The specified generation is written to the dataset connected to the DD statement named DDNAME in the TO parameter field. STARTREC and RECCOUNT can be used to limit the rows unloaded; STARTBYT can be used to control the portion of each row unloaded.

If the record length is greater than the row size, the remainder of the record is filled with spaces.

If the record length of the dataset defined by the TO keyword is less than the ROWSIZE of the table, the unloaded rows will be truncated to the record length of dataset.

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

UNLOAD command use

The following example demonstrates how to use the UNLOAD command:

Example 1—copy a table generation to an output dataset

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;
```


Target dataset rules

The rules for the target dataset are:

- QSAM DS with DISP=NEW or OLD are allowed; SHR is not allowed and will return error 33.
- BDAM DS is not allowed.
- PDS member with DISP=SHR or OLD is allowed
- PDSE member with DISP=SHR or OLD is allowed
- VSAM DS is not allowed
- DUMMY DD statements are allowed.

UPDATE command

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

UPDATE

TBL=(table name) **Required keyword**
TTYPE=(table name conversion - blank/V)
WITH=(DDNAME) **WITH and/or**
REMOVE=(DDNAME) **REMOVE are required**
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 REMOVE file, the table will be searched for a row with a matching key and if found the row will be deleted.

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.

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

If both WITH and REMOVE keywords are present, the rows are inserted/replaced before the REMOVE rows are deleted.

All actions taken during the UPDATE command are logged on the TBRPT file.

UPDATE command use

The following examples illustrate the use of the UPDATE command:

Example 1—update a table

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—eliminate duplicates

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 eliminates duplicate suppliers if there are multiple occurrences, prints the resulting table contents, and then deletes 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;
```

Note: For each record on the WITH file, the corresponding record will be replaced in its entirety, provided that the key is matched in the table. Therefore, the resultant table will not contain any duplicate supplier records.

3
Using TBEXEC (DK1TEXEC)

TBEXEC executes in a single-threaded batch environment against a local TSR. It issues tableBASE commands internally to verify or control its environment. It sets the WAIT switch internally to avoid unexpected results.

The TBEXEC is a batch tableBASE application. TBOPT can be used in a TBEXEC execution, just as it can be used in any batch job accessing tableBASE. For instance, it could be needed for TSRSIZE if the batch default is not large enough. For further information on the TBOPT dataset, see Appendix B on page 129.

This chapter describes the use of TBEXEC (DK1TEXEC). It begins by specifying the JCL required to run TBEXEC, followed by descriptions of how TBEXEC handles errors. Next, the standard reports produced by TBEXEC, audit report (TBMSG) and table listing report (TBRPT) are described. The chapter concludes with a listing of the error and audit messages issued by TBEXEC.

JCL to run TBEXEC

```
//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 *
command keyword = value ... keyword = value;
command keyword = value ... keyword = value;

/*
```

Note: TBRPT is required only if reports are to be produced

Create a new tableBASE library

To create a new tableBASE library:

1. Allocate the library dataset either by JCL for BDAM or by IDCAMS for VSAM.

2. Format the dataset with a TBEXEC DEFINE NEWLIB operation.

Below is JCL to create both a VSAM and BDAM tableBASE library:

```
//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;
/*
```

An equally valid variation is to allocate BDAMLIB as DISP=(,CATLG) in Step 1 (above) and reference it as DISP=OLD in Step 2 (above). If your installation uses only BDAM, only STEP2 is required.

Note: Experience has shown 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, a size of 3584 bytes is recommended, thus the allocation of VSAM libraries by an IDCAMS step (as shown above) is also recommended.

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

- Nine blocks (of 3120) for the space directory, larger for libraries greater than 168K blocks.
- 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.
- One block per table to hold the contents of all generations (step 2) for overhead for definitions of the tables.

- Space for one generation of the largest table. This is needed when updating a table, because the new generation is stored on the library prior to deletion of the oldest generation.
- Room for short-term growth.
- An average of one block per 20 tables for the tables directory.

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

EXPORT and IMPORT a tableBASE library

The following sample JCL shows the steps involved in copying a tableBASE library to a sequential dataset, and then copying the exported dataset back into the tableBASE libary:

```
//STEP1    EXEC PGM=TBEXEC
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//TBMSG    DD SYSOUT=*
//TBRPT    DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//TBDUMP   DD SYSOUT=*
//MAINLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LIBRARY
//EXPORT   DD DSN=*your.sequential.file*,DISP=(NEW,PASS),
//         UNIT=SYSDA,SPACE=(TRK,15),
//         DCB=(RECFM=FB,LRECL=512,BLKSIZE=5120,DSORG=PS)
//CNTLCARD DD *
PRINT  LIB=MAINLIB DETAIL=YES;
EXPORT LIB=MAINLIB TO=EXPORT;
IMPORT LIB=MAINLIB FROM=EXPORT REPLACE=YES;
PRINT  LIB=MAINLIB DETAIL=YES;
/*
```


UNLOAD a table

The following sample JCL creates a sequential output dataset and copies a tableBASE table to that output dataset:

```
//STEP01    EXEC PGM=TBEXEC
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//TBRPT    DD SYSOUT=*
//TBMSG    DD SYSOUT=*
//TBDUMP   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//MAINLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LIBRARY
//TESTDATA DD DSN=*your.sequential.file*,
//         DISP=(NEW,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(3120,(500,20),RLSE),
//         DCB=(RECFM=F,LRECL=3120,BLKSIZE=3120,DSORG=PS)
//CNTLCARD DD *
UNLOAD TBL=EXAMPLE TO=TESTDATA LIB=MAINLIB;
/*
```

LOAD a table

The following sample JCL creates a new tableBASE library and a new table, and then copies data from a sequential dataset into the new table:

```
//STEP01    EXEC PGM=TBEXEC
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//TBRPT    DD SYSOUT=*
//TBMSG    DD SYSOUT=*
//TBDUMP   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//TESTDATA DD DISP=SHR,DSN=*your.sequential.testdata.file*,
//TESTLIB  DD DISP=SHR,DSN=*your.test.library*,
//         UNIT=SYSDA,SPACE=(3120,(15,1)),
//         DCB=(RECFM=F,LRECL=3120,BLKSIZE=3120,DSORG=PS)
//CNTLCARD DD *
DEFINE NEWLIB=TESTLIB;
DEFINE TBL=TABLE1 MAXGEN=3 LIB=TESTLIB ORG=S MTHD=B INDEX=T
SMC=R ISZ=40 KSZ=4 KLOC=1 EXPFACTR=0150 LOWERDEN=900 UPPERDEN=999;
LOAD TBL=TABLE1 FROM=TESTDATA LIB=TESTLIB;
/*
```


Error diagnosis

TBEXEC recognizes three classes of errors: command interpretation errors, command processing errors, and TBLBASE processing errors.

Command interpretation errors

Command interpretation errors are errors detected during the interpretation of command sequences.

Examples

Examples of command interpretation errors are:

- invalid command
- invalid keyword
- missing delimiter

Action

When a command interpretation error occurs, an error message is printed and TBEXEC skips to the next command sequence.

Command processing errors

Command processing errors are errors detected by TBEXEC during command processing.

Example

An example of a command processing error is:

- A DDNAME is missing in the JCL.

Action

When a command processing error occurs, an error message is printed and processing resumes with the next command sequence.

TBLBASE processing errors

TBLBASE processing errors are errors detected within TBLBASE, the application programming interface to tableBASE. See the *tableBASE Programming Guide* for detailed information about TBLBASE.

Examples

Examples of a TBLBASE processing errors are:

- invalid key size
- invalid password

Action

When a TBLBASE processing error occurs, an error message is printed along with an interpretation of the error code set by TBLBASE. Processing resumes with the next command sequence. A list of all error codes is found in the *tableBASE Programming Guide*.

Note: While TBEXEC executes in an environment, the step return code is set to 16 if an error is encountered during interpretation or execution of any command sequence. A return code of zero means all command sequences have been processed successfully.

Report descriptions

TBEXEC produces two report files: TBMSG, and TBRPT.

TBMSG report

The TBMSG report displays the control statements, associated messages, and any TBLBASE error codes returned while attempting to execute a command. A sample of the TBMSG report is shown in Figure 3-1.

```
     DATE/TIME: 2003/11/04 10:44     TABLE BASE AUDIT REPORT           PAGE:   1

     TABLEBASE V601  LICENSED TO:
(1)



     *** REQUEST:  DEFINE  NEWLIB=TESTLIB;
         INITIALIZATION SUCCESSFUL
(2)
     *** REQUEST:  SET     LIB=TESTLIB;
(3)      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
(4)      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
(5)  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
(6)
```

Figure 3-1: tableBASE audit report

Notes:

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. Although proper syntax was used, an error occurred which prevented the successful completion of the command TBLBASE attempted to process it. The correct password needs to be provided in this case. tableBASE error codes are further described in the *tableBASE Programming Guide*.

5. A syntax error was detected by TBEXEC. The programmer misspelled STARTBYT in this case.

6. This line is printed after all commands have been processed.

TBRPT report

The TBRPT report prints selected tables and their related definitions in character format, hexadecimal format, or both. To print columnar-style reports based on the formatting information contained in the Views, see Chapter 5 "Table printing utility: TBPRINT (DK1TPTBL)" on page 71.

Figure 3-2 to Figure 3-6 are samples of the reports produced by TBEXEC.

Figure 3-2 and Figure 3-3 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. Two logs are produced if an UPDATE command uses both the WITH and REMOVE files.

```
DATE/TIME: 2003/11/04 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 3-2: Sample tableBASE update log report, with additions


```
DATE/TIME: 2003/11/04 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 3-3: Sample tableBASE update log report, with deletions

Figure 3-4 shows a sample directory listing produced by TBEXEC for the tableBASE library TESTLIB, which is on the dataset SYS03309.T104402.RA000.DKLS04X.LIB. As shown, the library contains only the three generations of TBL01, and lists their creation date. The report indicates the number of blocks allocated to the library and how many remain for additional tables. By default, each block contains 3120 characters of information, which can be set to a different value when the library is defined.

Possible values for table type:

- blank Data table
- VIEW View

```
DATE/TIME: 2003/11/04 10:44    TABLE BASE OPTIONAL REPORTS         PAGE:  1

DIRECTORY LISTING FOR LIBRARY TESTLIB  ON DATASET SYS03309.T104402.RA000.DKLS04X.L

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 3-4: tableBASE directory listing

Figure 3-5 and Figure 3-6 on page 57 are samples that show the definition and contents of table TBL01.

Figure 3-5 is a summary report that identifies: table type, absolute generation number, date created (YYYY/MM/DD), and time created (HH:MM) for each table generation in the library.


```
DATE/TIME: 2003/11/04 10:44   TABLE BASE OPTIONAL REPORTS                                          PAGE: 1

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    DKLS04
```

Figure 3-5: Sample tableBASE table definition report

The following information is provided reading from left to right on the report:

- absolute generation number
- maximum number of generations
- number of generations currently on the library
- relative generation number
- creation date and time
- actual number of rows
- estimate number of rows for which space is allocated when loaded in the TSR
- table organization
- table search method
- True or Pointer table type
- storage mode code (obsolete)
- if read and write passwords have been used

  Note: READ/WRITE indicates password security; it does not identify the type of password security (RO, R/W); NONE indicates no password security. (See example in Figure 3-5)
- row size
- key length
- key location
- expansion factor
- upper density
- lower density
- actual density of Hash tables
- the version of the View (if this table is a View)
- the user id
- associated View name if specified

Note: Views are part of tablesONLINE, an optional tableBASE component.

Figure 3-6 shows the contents of one generation of a table in character format. It shows:

- the name of the table being printed TBLO1
- the library where it resides.

The definition of the generation being printed is displayed in the same format described above.

```
DATE/TIME: 2003/11/04 10:44                TABLE BASE OPTIONAL REPORTS                  PAGE:   1

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 U
ABS M A REL      CREATED        PRESENT ALLOCATED R/E/N/M             SIZE SIZE  LOC FACTR UPPER LOWER  ACT VS

  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    D

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 3-6: Sample tableBASE table contents report

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 described

All error and audit messages generated by batch utility programs are identified and described in alphabetical order in Appendix A on page 99.

4

Using tableBASE Process Manager: TPDRIVER

TPDRIVER is the interactive component of the tableBASE Process Manager product. It is the conduit through which a user can control VTS management features. Further information (including sample JCL) is provided in the *tableBASE Installation Guide* and the *tableBASE Administration Guide*.

Access to TPDRIVER

Recall that using ISPF, you can communicate with tableBASE via the TBDRIVER utility to make changes to VTS table data. In a similar fashion, you can use the TPDRIVER utility, again via ISPF, to make administration and management changes to the management structures, tableBASE Process Manager, the various VTS Managers and the VTS-TSRs themselves.

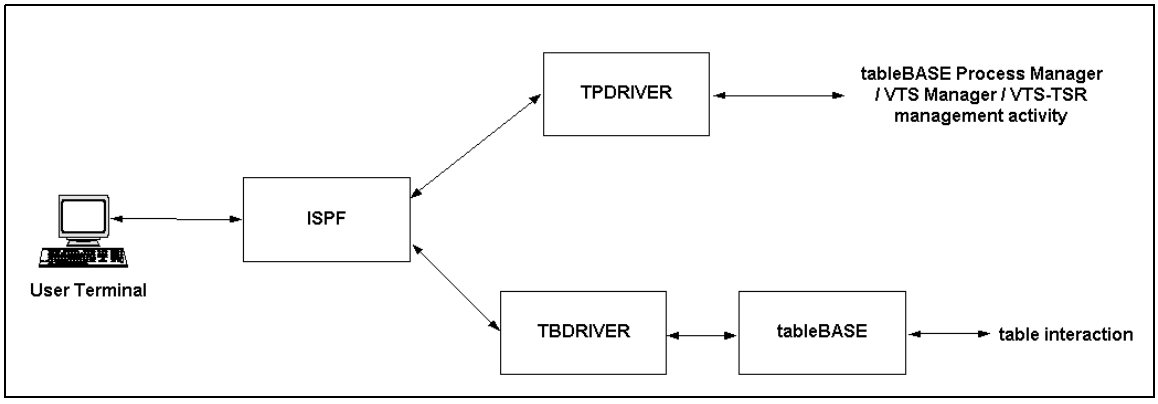

Figure 4-1: User communication with tableBASE and tableBASE Process Manager

Using TPDRIVER

TPDRIVER provides the following capabilities:

- To create, delete and modify VTS Manager defaults
- To start and shutdown VTS Managers
- To create, delete and modify VTS-TSR definitions
- To start and shutdown VTS-TSRs
- To define, switch and delete alias names applied to VTS-TSRs

You can run the TPDRIVER interactively.

TPDRIVER input and output files

TPDRIVER accepts one file for input, and produces one file for output. The input file has the DDNAME of TPDRCNTL with 80-byte fixed length control statements. The output file has the DDNAME of TPDRRPT with 132-byte fixed length output lines.

TPDRIVER characteristics

The following TPDRIVER characteristics are noteworthy:

- Commands are not case sensitive; all input is translated into uppercase before processing.
- Commands can transcend a single line. When this occurs, a comma is needed as the last valid character to indicate that the following line is part of the same command; for example:
```
SHOW VTS,
 NAME=myVTS
```
- Blanks can be inserted anywhere in a command line; for example:
```
Show VTS , NAME =  myVTS
```
- The default value for a parameter is used when the parameter is not explicitly included in the command line. For further information, see the specific commands in the section "tableBASE Process Manager commands" on page 61.
- A line is identified as a comment if its first non-blank character is an asterisk (*); for instance:
```
* this is a valid comment line
```
Note that inline comments are not allowed, i.e., this is invalid:
```
SHOW TPVM   * if no NAME, then use compatibility TPVM.
```


tableBASE Process Manager commands

tableBASE Process Manager uses only a handful of commands to execute its functionality. The following rules apply to all, and are enforced by TPDRIVER.

- All names must be from one to eight characters in length
- Names shorter than eight characters are automatically extended to eight using trailing blank spaces
- The first character of a name must be alphabetic (A-Z) or National (@, $ or #)
- Subsequent characters within a name must be either Alphanumeric (A-Z, 0-9) or National (@ or #)
- No other characters are allowed (+, -, =, punctuation, etc.), and will produce an error
- There is no case-sensitivity—both uppercase and lowercase alphabetic characters (A-Z, a-Z) are allowed; lowercase characters are translated, and are always displayed in uppercase.
- Examples:
  The following names are allowed: *A*, *VTS699*, *@TPVM3*, *########*, *asdf$1@#*
  The following are not: *4$vtsX*, *VTS_TEST*, *V%:!?*, *, *tpvmTest66*

Define ALIAS

The syntax of the **DEFINE ALIAS** command is:

```
DEFINE    ALIAS,NAME=xxxxxxxx,TPVM=xxxxxxxx,
          DESC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Required:

NAME	An alias name—it can be up to 8 characters in length.
TPVM	An existing VTS Manager (TPVM) must be included; its name can be up to 8 characters in length. Alias names are not applicable to the *compat* VTS Manager. The TPVM must be running.

Optional:

DESC	User comments; up to 40 characters. Double quotes can be used to enclose the description, if non-alphanumeric characters are used.

Define TPVM

The syntax of the **DEFINE TPVM** command is:

```
DEFINE    TPVM,AUTOSTART=Y|N,AUTOSHUT=Y|N,LDS=DSNAME,
          NAME=xxxxxxxx,PROC=xxxxxxxxx,JOBNAME=xxxxxxxx,
          DESC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Required:

NAME	The name of the newly defined VTS Manager (TPVM)—it can be up to 8 characters in length.
LDS	The VSAM Linear Dataset name must be included; the pathname can be up to 44 characters in length.
PROC	The name of the procedure stored in the system PROCLIB—it can be up to 8 characters in length.

Optional:

AUTOSTART	Determines whether or not the VTS Manager will start automatically when tableBASE Process Manager starts. A single character; either ‘Y’ or ‘N’. Default value is N.
AUTOSHUT	Determines whether or not the VTS Manager will automatically shut down in the event of a tableBASE Process Manager shutdown. A single character, either ‘Y’ or ‘N’. Default value is N.
JOBNAME	An 8 character name that replaces the STEPNAME of the procedure. This allows you to differentiate between multiple jobs assigned to the same TPVM, and to help identify the job in use. Default is the procedure name.
DESC	User comments; up to 40 characters. Double quotes can be used to enclose the description, if non-alphanumeric characters are used.

Define VTS

The syntax of the **DEFINE VTS** command is:

```
DEFINE    VTS,AUTOSTART=Y|N,AUTOSHUT=Y|N,LDS=DSNAME,LISTOPT=Y|N,MAXNMTAB=0-
          9999999,NAME=xxxxxxxx,PROC=xxxxxxxx,
          TPVM=xxxxxxxx,TSRACCESS=RO|RW,TSRSIZE=40K-2G,
          JOBNAME=xxxxxxxx,DESC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Required:

NAME	The name of the newly defined VTS-TSR—it can be up to 8 characters in length.
PROC	The name of the procedure stored in the system PROCLIB—it can be up to 8 characters in length.

Optional:

AUTOSTART	Determines whether or not the VTS-TSR will start automatically when VTS Manager starts. A single character; either ‘Y’ or ‘N’. Default value is N.
AUTOSHUT	Determines whether or not the VTS-TSR will automatically shut down in the event of a VTS Manager shutdown. A single character, either ‘Y’ or ‘N’. Default value is N.
LDS	The VSAM Linear Dataset name must be included if the VTS-TSR is mapped to an LDS; the pathname can be up to 44 characters.
LISTOPT	Determines whether or not to list tableBASE initialization options when starting the VTS-TSR—a single character ‘Y’ or ‘N’. Default is ‘N’.
MAXNMTAB	Defines the number of tables allowed in the TSR—a range of values from 0 to 999999. The default is 0, in which case the value is calculated by the given TSRSIZE and VTS startup.
TPVM	The name of an existing VTS Manager (TPVM); it can be up to 8 characters in length. When this parameter is not supplied, the default is the *compat* VTS Manager.
TSRACCESS	Two characters, ‘RO’ or ‘RW’ to define the VTS-TSR as read-only or read/write. Default is ‘RW’.
TSRSIZE	The TSR size—between 40K to 2G. Enter K for Kilobytes, M for Megabytes or G for Gigabytes. Default is 40K.
JOBNAME	An 8 character name that replaces the STEPNAME of the procedure. This allows you to differentiate between multiple jobs assigned to the same VTS-TSR, and to help identify the job in use. Default is the procedure name.
DESC	User comments; up to 40 characters. Double quotes can be used to enclose the description, if non-alphanumeric characters are used.

Note: It is good practice to ensure that the defined TSR size (TSRSIZE parameter) is the same as the associated LDS size. If a VTS-TSR is mapped to an LDS, and the TSRSIZE is inconsistent with the LDS size, the TSR size will be automatically adjusted to the LDS size.

If an LDS already has data in it, the value of MAXNMTAB and TSRSIZE will be ignored.

Delete ALIAS

The syntax of the **DELETE ALIAS** command is:

```
DELETE    ALIAS,NAME=xxxxxxxx,TPVM=xxxxxxxx
```

Required:

NAME	The name of the existing alias within an existing VTS Manager catalog—it can be up to 8 characters in length.
TPVM	The name of the existing VTS Manager (TPVM) where the alias was defined—can be up to 8 characters in length. Alias names are not applicable to the *compat* VTS Manager.

Delete TPVM

The syntax of the **DELETE TPVM** command is:

```
DELETE      TPVM,NAME=xxxxxxxx
```

Required:

NAME	The name of the existing VTS Manager within the tableBASE Process Manager (TPM) catalog—it can be up to 8 characters in length. It is not possible to delete the *compat* VTS Manager.

Delete VTS

The syntax of the **DELETE VTS** command is:

```
DELETE      VTS, NAME=xxxxxxxx,TPVM=xxxxxxxx
```

Required:

NAME	The name of the existing VTS-TSR—it can be up to 8 characters in length.

Optional:

TPVM	The name of an existing VTS Manager —it can be up to 8 characters in length. If absent, the *compat* VTS Manager is used.

Note: When the definition for an aliased VTS-TSR is deleted, the ALIAS will still be associated with the deleted VTS-TSR. If the VTS-TSR is then defined again, and the VTS-TSR started up, it would still be running under the alias name. This preserves the association of the VTS-TSR to the Alias.

Help

The **HELP** command does not take parameters. It lists the syntax of all commands:

```
DEFINE    ALIAS,NAME=xxxxxxxx,TPVM=xxxxxxxx,
          DESC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
DEFINE    TPVM,AUTOSTART=Y|N,AUTOSHUT=Y|N,LDS=DSNAME,
          NAME=xxxxxxxx,PROC=xxxxxxxxx,JOBNAME=xxxxxxxx,
          DESC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
DEFINE    VTS,AUTOSTART=Y|N,AUTOSHUT=Y|N,LDS=DSNAME,
          LISTOPT=Y|N,MAXNMTAB=0-9999999,NAME=xxxxxxxx,PROC=xxxxxxxx,
          TPVM=xxxxxxxx,TSRACCESS=RO|RW,TSRSIZE=40K-2G,
          JOBNAME=xxxxxxxx,
          DESC=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
DELETE ALIAS,NAME=xxxxxxxx,TPVM=xxxxxxxx
DELETE TPVM,NAME=xxxxxxxx
DELETE VTS,NAME=xxxxxxxx,TPVM=xxxxxxxx
HELP
SHOW TPVM,NAME=xxxxxxxx
SHOW VTS,NAME=xxxxxxxx,TPVM=xxxxxxxx
SHUTDOWN TPVM,NAME=xxxxxxxx
SHUTDOWN VTS,NAME=xxxxxxxx,TPVM=xxxxxxxx
START TPVM,NAME=xxxxxxxx
START VTS,NAME=xxxxxxxx,TPVM=xxxxxxxx
SWITCH ALIAS,NAME=xxxxxxxx,TPVM=xxxxxxxx,VTS=xxxxxxxx
```


SHOW TPVM

The syntax of the **SHOW TPVM** command is:

```
SHOW        TPVM,NAME=xxxxxxxx
```

Required: None.

Optional:

TPVM	The name of an existing VTS Manager from the tableBASE Process Manager catalog—it can be up to 8 characters in length. A wildcard character (*) can be used to list all VTS Manager names. If absent, only the *compat* VTS Manager is shown.

Output field descriptions:

TPVM Name	The name of the TPVM. (Should never be blank.)
Status	Status of the TPVM; one of: RUNNING, DEFINED (if it is defined, but not started), SHUTDOWN, REDEFINE (if it is defined, started, shut down, deleted, and defined again), STARTING, STOPPING. (Should never be blank.)
Version	The version of tableBASE Process Manager.
# of active VTSes	Number of VTS-TSRs currently running with this TPVM.
Time Start/Stop	The time and date stamp of the most recent start or shut down of the TPVM. (Blank if the TPVM is defined, but has never been started.)
Current # VTSes in Cat	Number of VTS-TSRs currently defined in the TPVM catalog. (Should never be blank.)
Max # VTSes in cat	The maximum number of VTS-TSRs that can be defined in the TPVM catalog. (Should be 64.)
Job Step Name	Job step name assigned at definition (if none specified, the name PROC is shown).
Startup Procedure	Name of the TPVM startup PROC. (Should never be blank.)
Update Time Stamp	The time and date stamp of TPVM definition. (Should never be blank.)
Updated by (username)	System / TSO username of user responsible for the most recent update of the definition. (Blank for TPVM *compat*.)
Startup Flag	Autostart setting, based on the AUTOSTART parameter setting at definition (Y or N).
Shutdown Flag	Auto-shutdown setting, based on the AUTOSHUT parameter setting at definition (Y or N).
Access Flag	Type of TPVM (Always *W).
LDS Name	The LDS associated with the TPVM. (Blank for TPVM *compat*.)
Description	The text description of the TPVM, as entered at definition. (Blank if there was no text entered at definition.)

SHOW VTS

The syntax of the **SHOW VTS** command is:

```
SHOW        VTS, NAME=xxxxxxxx,TPVM=xxxxxxxx
```

Required:

NAME	Will list a particular VTS-TSR or ALIAS definition whichever matches the specified name. A wildcard character (*) can be used, and will return a list of all VTS-TSRs and alias definitions. The name can be up to 8 characters in length. For aliases, both the alias name and the corresponding VTS-TSR will be returned. For alias names with no corresponding VTS-TSR, only the alias name will be returned.

Optional:

TPVM	The name of an existing VTS Manager from the tableBASE Process Manager catalog—it can be up to 8 characters in length. To list all VTS-TSRs in all VTS Managers, use a wildcard character (*) in the NAME parameter. If absent, only the VTS-TSRs under the *compat* VTS Manager are shown.

Note: When a VTS-TSR is defined using the **DEFINE VTS** command, without specifying a value for MAXNMTAB, the value will appear as 0 in the **SHOW** command after the VTS-TSR has been defined. However, when the VTS-TSR is started up, a calculation will be performed to determine what its value should be based on the TSR size that will be used at VTS startup. At this time, the defined MAXNMTAB value (of 0) will be replaced with the calculated value.

Output field descriptions:

TPVM Name	The name of the TPVM that manages the VTS-TSR. (Should never be blank.)
VTS Name	The name given to the VTS-TSR at definition. (Should never be blank.)
Alias Name	The alias currently associated with the VTS-TSR. (Blank if there is no association.)
Status	Status of the VTS-TSR; one of: RUNNING, DEFINED (if it is defined, but not started), SHUTDOWN, REDEFINE (if it is defined, started, shut down, deleted, and defined again), STARTING, STOPPING. (Should never be blank.)
Version	The version of tableBASE.
Time Start/Stop	The time and date stamp of the most recent start or shut down of the VTS-TSR. (Blank if the VTS-TSR is defined, but has never been started.)

TSR size	Actual size of the TSR. If the defined TSR size and the associated LDS size are different, the LDS size is shown; LDS size overrides defined size. (Should never be blank.)
Job Step Name	Job step name assigned at definition (if none specified, the name PROC is shown)
Startup Procedure	Name of the VTS startup PROC. (Should never be blank.)
Update Time Stamp	The time and date stamp of VTS-TSR definition. (Should never be blank.)
Updated by (username)	System / TSO username of user responsible for the most recent update of the definition. (Should never be blank.)
Startup Flag	Autostart setting, based on the AUTOSTART parameter setting at definition (Y or N).
Shutdown Flag	Auto-shutdown setting, based on the AUTOSHUT parameter setting at definition (Y or N).
Access Flag	Type of VTS-TSR, based on the TSRACCESS parameter setting at definition (RO or RW). (Should never be blank.)
Use LDS	Indicates whether or not an LDS was associated with the VTS-TSR at definition (Y or N).
ListOptions	Indicates whether or not initialization options are listed when a VTS-TSR is started (Y or N).
Max number of tables	Actual maximum number of tables for the VTS-TSR. Either the number specified at definition, or the calculated number in cases where the number was set to 0 at definition.
LDS Name	The LDS associated with the VTS-TSR. (Blank if there was no association.)
Description	The text description of the VTS-TSR, as entered at definition. (Blank if there was no text entered at definition.)

SHUTDOWN TPVM

The syntax of the **SHUTDOWN TPVM** command is:

```
SHUTDOWN    TPVM,NAME=xxxxxxxx
```

Required:

NAME	The name of an existing VTS Manager from the tableBASE Process Manager catalog—it can be up to 8 characters in length. It is not possible to shut down the *compat* VTS Manager. The specified TPVM must be currently running.

SHUTDOWN VTS

The syntax of the **SHUTDOWN VTS** command is:

```
SHUTDOWN    VTS, NAME=xxxxxxxx,TPVM=xxxxxxxx
```

Required:

NAME	The name of an existing VTS-TSR—it can be up to 8 characters in length. The specified VTS-TSR must be defined and currently running.

Optional:

TPVM	The name of an existing VTS Manager from the tableBASE Process Manager catalog—it can be up to 8 characters in length. If not used, the default is to shut down the specified VTS-TSR under the *compat* VTS Manager.

Note: When shutting down a VTS-TSR, you must use the name under which it is running. If the VTS-TSR has an alias name, you must use it; if it does not have an alias name, you must use the true name (the name assigned to it at definition).

START TPVM

The syntax of the **START TPVM** command is:

```
START       TPVM,NAME=xxxxxxxx
```

Required:

NAME	The name of an existing VTS Manager—it can be up to 8 characters in length. The specified VTS Manager must be defined in the tableBASE Process Manager catalog, and should not be running.

START VTS

The syntax of the **START VTS** command is:

```
START       VTS,NAME=xxxxxxxx,TPVM=xxxxxxxx
```

Required:

NAME	The name of an existing VTS-TSR—it can be up to 8 characters in length. The specified VTS-TSR must be defined in the VTS Manager catalog, and should not be running.

Optional:

TPVM	The name of an existing VTS Manager from the tableBASE Process Manager catalog—it can be up to 8 characters in length. The VTS Manager (TPVM) should be defined in the catalog, and it should be running. If not used, the default (spaces) will start the specified VTS-TSR under the *compat* VTS Manager.

Note: You can start a VTS-TSR using either the alias name (if assigned), or its true name (the name assigned to it at definition).

SWITCH ALIAS

The syntax of the **SWITCH ALIAS** command is:

```
SWITCH ALIAS,NAME=xxxxxxxx,TPVM=xxxxxxxx,VTS=xxxxxxxx
```

Required:

NAME	The alias name that you want to use—it can be up to 8 characters in length.
VTS	The VTS-TSR to which you want to apply this alias name—it can be up to 8 characters in length. Does not apply to VTS-TRS belonging to the *compat* VTS Manager.
TPVM	The VTS Manager to which the specified VTS-TSR belongs. Does not apply to the *compat* VTS Manager.

5 Table printing utility: TBPRINT (DK1TPTBL)

The Table Print Utility, TBPRINT (DK1TPTBL), is used to print formatted columnar listings using the formatting information contained in Views, used by tablesONLINE/ CICS and tablesONLINE/ISPF. The utility differs from the TBEXEC PRINT command which is intended to supply whole or partial table dumps both in character and hexadecimal formats.

This chapter describes TBPRINT input parameters, shows a variety of printed examples, and ends with an example of TBPRINT execution in JCL.

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.

Note: Views are a function of tablesONLINE, an optional tableBASE component.

Completion codes

Completion codes listed in Table 5-1 are set to indicate whether problems were encountered.

Table 5-1: Completion codes

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

Keyword parameters

Input parameters for TBPRINT can be one of two forms:

- keywords with assigned values
- keywords without assigned values

Keywords with assigned values

The delimiters listed in Table 5-2 can be used to separate multiple values for a keyword:

Table 5-2: Delimiters

Delimiter	Shown as
comma	,
blank	
left parenthesis	(
right parenthesis	)
equal sign	=
double quotes	"

Using equal signs to separate the keyword and values, and commas to separate values, lead to a readable code style—used in the examples that follow.

KEYWORD value[,value]... ;

LIB

Specify the libraries to be searched for a table. Between 1-10 library names consisting of up to 8 characters each are allowed. If the keyword is omitted, searching is restricted to MAINLIB.

Example: LIB=TESTLIB1,TESTLIB2

Note: DK1TPTBL does not use LIBnn specified in TBOPT

TABLE

Indicate which tables are to be printed. Table name(s) from 1-8 characters are allowed.

Example: TABLE=CODE1,CODE2,CODE3

PASSWORD

Supply a 1-8 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 are required if the Data Tables specified in the TABLE keyword have different read passwords.

GEN

Supply a number ranging between -8 and 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

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

Examples:
PRINT TABLE=CODE1 GEN=-2;
PRINT TABLE=CODE1 GEN=-1;
PRINT TABLE=CODE1;

USING

Specify a 1-8 character View name to be used. The default is the same name as the table being printed.

Example: PRINT TABLE=CODE1,CODE2,CODE3 USING=CODE;

TITLE

Supply a 1-50 character title, enclosed in double quotes, or, supply a 1-50 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.

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

ROWIDCOUNT

To identify the fields that overflow onto a second or subsequent page, the ROWIDCOUNT value 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.

ROWIDCOUNT 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.

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

Note: Currently in Version 6, the limit of PAGEWIDTH=131 needs to be specified to display the Display Count correctly.

Example: PRINT TABLE=CODE1 PAGEWIDTH=131;

FIELDS

Supply the field name(s) to be selected from the View for printing. The default is to print all fields. Use double quotes to enclose field names that contain embedded blanks.

Examples:
```
 FIELDS=FIRST,
 SECOND,
 "NEW CODE",
 "OLD CODE"
```

Field names can be selected using wildcards by placing an asterisk **(*)** after the significant characters of the field name. All fields beginning with the significant characters will be printed. Placing an asterisk only results in the selection all fields.

To print all fields beginning with D, for example:

```
FIELDS=D*
```

To print all fields and two again:

```
FIELDS=*,KEY1,KEY2
```

The first occurrence of a generic field name 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.

To print the first two fields beginning with D, for example:

```
FIELDS=D!,D!
```

Note: All fields in a View with ATTRIBUTE=SUPPRESS may be printed only if they are 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 with the next higher (ascending) or lower (descending) key.

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. The overrides are listed in Table 5-3. The FORMAT label is optional. It serves only to make the PRINT command specification more readable. Choose one option for each format override function.

Table 5-3: Format Overrides

Function	Options	Description
Title and column headings	HEADINGS	Title and column headings. This is the default.
	NOHEADINGS	Neither title nor column headings
	TITLESONLY	Only title headings
	COLUMNHEADONLY	Only column headings
Handling of long data columns	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.
Handling of long fields	COLUMNSPLITON	For long fields, truncate last field on a page if necessary.
	COLUMNSPLITOFF	For long fields, maintain complete field. This is the default.

Table 5-3: Format Overrides

Handling of row count display	DISPLAYCOUNTON	Use last five columns of report for displaying a row count number. This is the default. Currently in Version 6, the limit of PAGEWIDTH=131 needs to be specified to display the Display Count correctly.
	DISPLAYCOUNTOFF	Use to suppress the display of Display Count and allow full page for printing.

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 table using defaults

PRINT TABLE=EXAMPLE;

The defaults are 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 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 with row truncation

PRINT TABLE=EXAMPLE TRUNCATE;

The COLUMNSPLITOFF default prevents the last field from being split.

```
PAGE 1        TITLE
  F1      F2         F3

______  ____________________
AAAAAA  xxxxxxxxxxxxxxxxxx
BBBBBB  xxxxxxxxxxxxxxxxxx
CCCCCC  xxxxxxxxxxxxxxxxxx
DDDDDD  xxxxxxxxxxxxxxxxxx
EEEEEE  xxxxxxxxxxxxxxxxxx


PAGE 2        TITLE
  F1      F2         F3

______  ____________________
FFFFFF  xxxxxxxxxxxxxxxxxx
GGGGGG  xxxxxxxxxxxxxxxxxx
HHHHHH  xxxxxxxxxxxxxxxxxx
IIIIII  xxxxxxxxxxxxxxxxxx
```

PRINT TABLE=EXAMPLE TRUNCATE COLUMNSPLITON;

Fills the page.

```
PAGE 1          TITLE
  F1      F2         F3        F4

______  ____________________ ___
AAAAAA  xxxxxxxxxxxxxxxxxx xxx
BBBBBB  xxxxxxxxxxxxxxxxxx xxx
CCCCCC  xxxxxxxxxxxxxxxxxx xxx
DDDDDD  xxxxxxxxxxxxxxxxxx xxx
EEEEEE  xxxxxxxxxxxxxxxxxx xxx


PAGE 2           TITLE
  F1      F2         F3         F4

______  ____________________ ___
FFFFFF  xxxxxxxxxxxxxxxxxx xxx
GGGGGG  xxxxxxxxxxxxxxxxxx xxx
HHHHHH  xxxxxxxxxxxxxxxxxx xxx
IIIIII  xxxxxxxxxxxxxxxxxx xxx
```


Print with rows continuing on next page

PRINT TABLE=EXAMPLE HORIZONTAL ROWIDCOUNT=1;

F1 is repeated for identification.

```
PAGE 1          TITLE
  F1     F2          F3

______  ____________________
AAAAAA  xxxxxxxxxxxxxxxxxx
BBBBBB  xxxxxxxxxxxxxxxxxx
CCCCCC  xxxxxxxxxxxxxxxxxx
DDDDDD  xxxxxxxxxxxxxxxxxx
EEEEEE  xxxxxxxxxxxxxxxxxx


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
```


JCL

Here is an example of an execution of TBPRINT:

```
//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;
/*
```


6
View printing utility: TBDEFPRT (DK1TVWPR)

TBDEFPRT (DK1TVWPR) 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.

The View is multipurpose. It is used in both the ISPF and CICS versions of the optional tablesONLINE application. The utility program TBPRINT also uses the View.

Input

Control statements are input from a file with the DDNAME COMMAND. The 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

Each Keyword is separated from its value(s) by an equal sign (=).

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 is 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

FROMDDN identifies the DDNAME of the tableBASE library that contains the tables to be printed. A JCL statement must be included. Either the FROMDDN or FROMDSN keyword must be defined.

Example: FROMDDN=MAINLIB;

FROMDSN

Identifies the Dataset 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. Either the FROMDDN or FROMDSN keyword must be defined.

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

SEQ

View rows will be printed in sequence by their position in view by default. 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 of the Views to be printed are enclosed in parentheses in uppercase characters. 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 in parentheses can be separated by commas or spaces.

Example:
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.

Examples:
FROMDDN=MAINLIB
PASSWORD=SECRET
SELECT=(CODE1,CODE2) ;

FROMDDN=MAINLIB
PASSWORD=FORGET
SELECT=CODE3 ;

JCL

Here is 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=*
//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;
/*
```


7

Generating data definitions: TBCOBFD and DK1TCSTR

This chapter describes two programs; TBCOBFD(DK1TPTBL), that generates COBOL copybooks and DK1TCSTR that generates C/C++ structures.

The View is multipurpose. It is used in both the ISPF and CICS versions of the optional tablesONLINE application.

Generating COBOL copybooks

The utility program, TBCOBFD (DK1TCOBF), generates COBOL copybooks and, optionally, the tableBASE COMMAND-AREA from Views that describe table row layouts.

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.

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

DATE fields are divided into subfields (year, month, day). These subfields are defined at level 10; a suffix is appended to each subfield to uniquely identify it. For example, an A format DATE field (YYYMMDD) is redefined with three level-10 fields suffixed with: -YY, -MM and -DD.

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 subfields. (COBOL field names are limited to 30 characters in length.)

Using TBCOBFD

Select the View for which a COBOL copybook is to be generated using 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.

FIELD PREFIX

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. Y 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 item 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. The View is read from MAINLIB. The View is written to the DSN assigned to CBLOUT. TBCOBFD always returns a condition code of zero.

JCL

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

Generating C/C++ structures

The utility program, DK1TCSTR, generates C/C++ structures from Views that describe table row layouts.

Note: DK1TCSTR is not shipped on tableBASE tapes or CDs. Please contact DataKinetics Technical Support at 1-613-523-5588 if you want this program.

Using DK1TCSTR

Select the View for which a C/C++ structure is to be generated using the PARM field of the JCL used to invoke the program. The PARM field is also used to define a structure name. An example of the specification of these parameters is shown below.

```
//STEP1 EXEC PGM=DK1TCSTR,PARM='view,struct_name'
```

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 C/C++ structure is to be generated.

STRUCT_NAME

This second parameter becomes the name of the generated structure. This parameter has a maximum length of 25 characters.

JCL

```
//STEP1   EXEC PGM=DK1TCSTR,PARM='PRIKEY,EXAMPLE_STRUCT'
//STEPLIB  DD DISP=SHR,DSN=*YOUR.PREFIX*.TBASE.LOAD
//TBDUMP   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD *
//CBLOUT   DD DSN=*YOUR.PREFIX*.CSTRUCT.LIBRARY(MEMBER),DISP=SHR
//MAINLIB  DD DSN=*YOUR.PREFIX*.TBASE.LIBRARY,DISP=SHR
/*
```


8 Table comparison utility: TBCOMP (DK1TCOMP)

Comparing tables with TBCOMP

The tableBASE Comparison (TBCOMP) program compares tables and identifies any differences for review. Tables are considered unequal if their definitions are significantly different or if any of the rows do not match.

Note: TBCOMP is used extensively to upgrade to a new release of tablesONLINE.

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

SEARCH-METHOD
ORGANIZATION
INDEX
STORAGE-MODE
ROW-SIZE
KEY-LOCATION
KEY-SIZE
EXPANSION-FACTOR
LOW-DENSITY
HIGH-DENSITY
table contents

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

The TBCOMP program 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, as listed in Table 8-1. The fields on each input statement must be given in a specific order with each field separated by one or more blanks or a comma.

Table 8-1: Input statement

Field	Description/comments
Command	Command may be COMPARE or CM
Comments	Comments follow an asterisk (*) in column 1
Table	New or old table to be compared
Library DDNAME	DDNAME of library where table resides
Read password	Read password of table, if any
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

The Command must start in column 1. An asterisk (*) in column 1 indicates a comment. TBCOMP ignores comment lines. The mandatory fields in each statement are:

- Command
- 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. It generates the differences required for making the changes to the second table, to be identical to the first.

TBL01
A
B
D

TBL02
B
C
G

DIFFERENCES

UPDATE
A
D

DELETE
C
G

From the example above, you can see that when you first apply the DELETE set to TBL02, followed by the UPDATE set also to TBL02, you will have the identical contents as TBL01.

TBCOMP can report the differences in any of three formats:

- two printed reports
- two flat files
- two difference tables

TBCOMP does not change any of the tables that it analyzes. Tables are changed by:

- inspecting the printed report and editing tables manually
- using the unload files as input to TBEXEC
- writing your own program to read the difference tables or the unload files

Using the printed reports allows for the inspection of each change to ensure that it is correct for the system.

Using the unload files is the simplest method to apply changes to a table. This is described in the next section.

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

A report is always printed containing the results of each comparison, which is summarized on one line showing table and library names, generation numbers, result code for the comparison, and a brief description, as listed in Table 8-2.

Table 8-2: Result codes

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 error codes from tableBASE—for example, invalid password

TBCOMP will issue one of the return codes listed in Table 8-3.

Table 8-3: 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 tables comparison job with JCL follows:

```
//*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.TBASE.LIBRARY1.NEW*     <====
//OLDLIB1  DD DISP=SHR,DSN=*YOUR.TBASE.LIBRARY1.OLD*     <====
//NEWLIB2  DD DISP=SHR,DSN=*YOUR.TBASE.LIBRARY2.NEW*     <====
//OLDLIB2  DD DISP=SHR,DSN=*YOUR.TBASE.LIBRARY2.OLD*     <====
//TBSYSLB  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 *
*CMD,TBL1,LIB1,PSWD1,GEN1,VIEW1,UPDTDD
*CMD,TBL2,LIB2,PSWD2,GEN2,VIEW2,DELEDD
* EXAMPLE 1
COMPARE,TABLE1,NEWLIB1,RPSWD1,0,VIEW1,UPDTTBL1
CM,TABLE1,OLDLIB1,RPSWD2,-1,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 (leave empty). This is optional. The default is to not print differences.
- S is a flag indicating whether to create difference tables (value S or Y) or not (leave empty). This is optional. The default is to not create the difference tables.

- UPDT and DELE are prefixes for difference table names when difference tables are to be created. 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). This is optional. The defaults are UPDT and DELE. For example, if you coded U2 and D3 for UPDT and DELE respectively and the name of both tables being compared is TABLE123, the DELETE difference table name would be named D3BLE123 and the UPDATE table would be named U2BLE123.
- nnnnnn is the master password. This is optional.There is no default.
- N is a flag indicating that the data content comparison is not case sensitive. Uppercase 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.

The COMPARE command provides for the optional parameter to specify 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 unload 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 unload files.

Using difference files with TBEXEC

TBCOMP analyzes the differences between tables by treating the first table as the new table and the second table as the old table. The two difference files that are generated when applied to the old table cause it to be identical to the new table. This is not the only way to apply the differences files, however the example below demonstrates how to update the old to bring it up to the new.

The first unload file contains rows which should be deleted from the second target table. The second unload file contains rows which should be inserted/replaced in the second 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 unload file. Therefore, deletes must be applied before updates. Please see the following example:

```
//UPDTE  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;
//
```


9
Library conversion utilities

Two new utilities DK1TCNV and DK1TLCHK are delivered with tableBASE 6.0.2 and Library Bridge (tableBASE 5.B).

The Library Conversion Utility, DK1TCNV, is used to convert tableBASE libraries between Version 5 (V5), Library Bridge (Bridge), Version 6 Transition (V6TRANS), and Version 6 (V6).

The Library Version Identification utility, DK1TLCHK, is used to identify the version of a tableBASE library.

For more information please consult the "Converting tableBASE libraries" Appendix in the *tableBASE Installation Guide* or the Library Bridge documentation.

Appendix A

tableBASE Messages

This appendix contains all of the DataKinetics tableBASE error codes and messages that can be encountered during the normal installation, administration and operation of the product. The error codes and messages fall into the following categories:

TBEXEC error messages—conditions that may be encountered while running tableBASE in conjunction with the TBEXEC batch utility program (see "TBEXEC error messages" on page 100).

tableBASE Process Manager messages—conditions that may be encountered while running tableBASE Process Manager (see "tableBASE Process Manager messages" on page 105).

tableBASE messages—conditions that may be encountered with running tableBASE in conjunction with the tableBASE VTS agent, CICS, batch, IMS, and other programs/utilities (see "tableBASE messages" on page 119).

Other messages—conditions that may be encountered while tableBASE performs user initiated activities (see "TBEXEC and library conversion messages" on page 126).

TBEXEC error messages

Error and audit messages generated by the batch utility program TBEXEC are identified and described in alphabetical order in the following table.

Table A-1: tableBASE batch utility messages

Message	Text	Meaning/Instructions
Data table index must be defined as P	TBEXEC attempted to invoke an Alternate Index of a table that is not defined to be type P (indexed or pointer).	
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 Index	The CHANGE ALT= command specified a table that is not an Alternate Index. 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.	Information only.
CMD requires specified table to be open	This command requires the specified table to be open. This error should not occur when using TBEXEC.	
Command is invalid for an Alternate Index	The command cannot be performed on an Alternate Index.	
Command rejected for reason above	The preceding TBEXEC Command was rejected for the reasons noted above the message.	
Command requires more parms than given	TBEXEC called tableBASE incorrectly.	
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. Diagnostic messages are issued to give the reason(s) for the failure.	
Command is invalid	The specified command is not a TBEXEC command.	
Copy of table (all generations) complete	The requested copy operation of all generations of a table was successfully completed.	
Create Alternate Index definition failed	The attempt to create an Alternate Index definition failed for the reasons noted in preceding messages or to the right of this message.	
Create Alternate Index definition successful	The Alternate Index definition has been created successfully.	Information only.
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 must be from 1-999	The density must be numeric and in the range 1-999.	
Destination library too small	The destination library is too small to hold the table(s) to be copied.	

Message	Text	Meaning/Instructions
Directory is empty	The directory on the specified library is empty.	
DT command (maxgen parm) must be 1-9	The maximum generations to be kept must be numeric in the range 1-9 inclusive.	
Duplicate keyword for this command	The same keyword was specified twice for a command. Fix the command sequence and try again.	Fix the command sequence and try again.
End of data - tableBASE utility ended	End of data has been reached on CNTLCARD, the TBEXEC input file.	Information only.
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.	
Estimated number-of-rows is out of range	The estimated number of rows is non-numeric, or is too large.	
Expand library failed	The requested operation failed. Diagnostic messages are issued to give the reason(s) for the failure.	
Expand library successful	The requested operation completed successfully.	Information only.
Expansion factor must be from 1 To 999	The expansion factor must be numeric and in the range 1-999.	
Field is greater than 8 characters	All of the keywords and keyword values have a maximum length of eight characters.	Ensure field is 8 characters or less.
Format of library incompatible with Version 6	The library format needs to be converted to be compatible with Version 6. Please see the tableBASE Installation Guide, Appendix B.	
Generation number specified is invalid	The generation must be numeric and in the range 0-255.	
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.	Information only.
Index parameter must be P, T or blank	The Index field must be T for true tables, P for indexed (Pointer) tables or blank. In previous releases, tableBASE allowed two types of tables: Pointer and True. In Version 6, the concept of True tables still exists, however they are treated within tableBASE as Pointer tables as all memory is now in segmented memory that requires Indexes. The True table Indexes will be transparent to the application program.	
Initialization successful	The new tableBASE library has been initialized successfully.	Information only.
Insufficient local tableSPACE region	The TSR 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.	
Insufficient space on new library	There is not enough space on the target library for the tables being copied.	
INV storage-mode-code: Must be R, Blank	The storage mode code (SMC) must be R or blank.	
Key size invalid: must be1-256	The key size must be a number from 1-256, inclusive.	
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.	
Keyword invalid for this command	The keyword specified was not recognized.	
Keyword is incomplete	The keyword has not been followed by an equal sign (=)and a value.	
Keyword is invalid	The keyword specified is not used with this command.	
KLOC invalid: must be 1 to row-size	The key location must be numeric in the range 1 to the row size, inclusive.	

Message	Text	Meaning/Instructions
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. Diagnostic messages are issued to give the reason(s) for the failure.	

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.	Information only.
Newlib DDNAME not Assigned -- check JCL	The requested initiation of the DDNAME specified by the NEWLIB parameter could not be performed since the DDNAME is not defined in the JCL.	Alter JCL as required.
nnnnn/mmmmm tables exported successfully	nnnnn tables were exported from the tableBASE library successfully. mmmmm tables were requested to be exported.	Information only.
nnnnn/mmmmm tables imported successfully	nnnnn tables were imported to the tableBASE library successfully. mmmmm tables were requested to be imported.	Information only.
No changes specified	A change command did not identify any fields to be changed.	Identify fields as required.
Not copied, new name same as old name	A COPY table request specified the same value for NEWNAME as for TBL.	
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 Index definition deleted	During a copy of an Alternate Index 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.	
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.	
Org parm invalid: must be R, U, S, D or H	The Organization must be R, U, S, D, or H.	
Paged tables are not supported	Version 6 no longer supports paged tables.	
Print request completed successfully	The requested print operation has finished with no errors. The listing can be found in TBRPT.	Information only.
Rename failed	The requested operation failed. Diagnostic messages are issued to give the reason(s) for the failure.	
Rename successful	The requested rename operation was successfully performed.	Information only.
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.	Information only.
Requested generation has been deleted	The generation of the table specified has been deleted.	Information only.
Requested generation(s) deleted	The requested operation completed successfully.	Information only.
Required keyword missing for this command	One of the required keywords for this command was not contained in the command sequence.	Correct command sequence as required.

Message	Text	Meaning/Instructions
RN fails. new name Already Exists	The table was not renamed because a table with the new name already exists on the library.	
Row size invalid: must be 1-32767	The row size must be numeric in the range 1-32767 inclusive.	

Search-method incompatible with organization	The search method must be S, Q, B, C, or H and must match the organization. Valid combinations are:
	`Organization  Method`
	`------------  ------`
	`R, U  S`
	`S, D  S, B, C`
	`H  H`	

Semi-colon is missing from command	The command and its keywords must be followed by a semicolon (;) to indicate the end of the keyword list for the command sequence.	
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.	
tableBASE error detected	The requested operation failed. Diagnostic messages are issued to give the reason(s) for the failure.	
tableBASE internal program error - xxxx	The requested operation failed. Diagnostic 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 is not closed	This error should not occur when using TBEXEC.	Contact Technical Support.
Table is not found	The table was not found on the given library.	
Table name specified is invalid	A valid tableBASE table name is a string of 8 bytes that are not all blanks, all low values, all high values, or :TMPNAME.	
Table opened for read cannot be stored	A table has been opened for read-only access and a store or write command has been issued.	
Table type is invalid	The table type must be F, V, A, or X.	
Table unavailable. No wait in effect	TBEXEC could not store the table because it is locked by another application. The NO WAIT parameter is in effect.	
Table unloaded successfully	The specified generation of the specified table has been unloaded to the TO dataset.	Information only.
Table updated as requested	The requested update operations on the specified table were performed successfully.	Information only.
Table updated successfully	The requested operation completed successfully.	Information only.
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.	
tablebase error detected	A TBLBASE error was detected in the processing of this command. The explanation of the error is found to the right of this message.	

Message	Text	Meaning/Instructions
The count specified is invalid	The count is less than one, or greater than the number of rows in the table.	
The key will not fit within the row	The end of the key cannot exceed the end of the row.	
The library DDNAME does not exist	The LIB=DDNAME is not in the JCL.	Alter JCL as required.
The library status is invalid	tableBASE library disposition must be NEW for new libraries, and SHR or OLD for existing libraries.	
The password supplied is invalid	An incorrect password has been specified.	
The specified command is invalid	The specified command is not a valid TBEXEC command.	
The specified library is not suitable	There are a number of conditions that can create this error. See tableBASE error codes.	
Write password is missing or incorrect	The operation requires access to a write-protected table, but the write password has not been specified or specified incorrectly.	
xxxxxxxx field is greater than 8 characters	All keywords and keyword values must be eight characters or fewer.	

tableBASE Process Manager messages

The following table contains information on the tableBASE Process Manager error messages that can be encountered. Note that the *Error Code* is, in all cases, in the format DK1PnnnnnX, where X is a letter code suffix, which indicates the type and severity of the error message:

E—Error (usually indicates user error)
I—Information
W—Warning
A—Action (user action is required)
S—Severe (no further processing is possible)

Table A-2: tableBASE Process Manager messages and error codes

Error Code	Text	Meaning / Instructions
DK1P02000E	RECOVERY PARMS NOT AVAILABLE - RECOVERY STOPPING	
DK1P02001I	TB 6.1 INITIALIZATION PROCESSING STARTING	Information only.
DK1P02002S	TPM START ENQ NOT AVAILABLE - TERMINATING	
DK1P02003I	P1063 TRMEXIT: TPM CB NOT FOUND OR NOT RUNNING	
DK1P02004S	INITIAL SET RECOVERY-ON FAILED - TERMINATING	
DK1P02006S	INITIAL SNAPX OPEN FAILED - TERMINATING	
DK1P02007S	INPUT PARM(S) NOT CORRECT - TERMINATING	
DK1P02008S	MESSAGE INIT REQUEST FAILED - TERMINATING	
DK1P02009I	TPM START PARM USED: <parm>	Information only.
DK1P02010S	DK1PMGR NOT AUTHORIZED - TERMINATING	Load library which contains DK1PMGR is not an authorized library. Make sure you are running from an authorized library.
DK1P02011I		Not used.
DK1P02012S	MODESET macro failed - terminating	
DK1P02014S	RESMGR macro failed - terminating	
DK1P02015I	Recovery detected /C (cancel) command	Information only.
DK1P02016S	Failure setting recovery on - terminating, routine = <routine name>, RC= <return code #>	
DK1P02017I	P1063 TRMEXIT: PROC ERROR - POSSIBLE JCL ERROR	
DK1P02018S	Failure setting recovery off - terminating, routine = <routine name>, RC = <return code #>	
DK1P02019I	P1063 TRMEXIT: POST COMPLETED	Information only.
DK1P02020S	P1061 RC not zero - terminating, RC= <return code #>	
DK1P02022S	TPM Request Mgr ATTACH failed - terminating, RC= <return code #>	
DK1P02024S	QEDIT1 macro failed - terminating, RC= <return code #>	
DK1P02026S	QEDIT2 macro failed - terminating, RC= <return code #>	
DK1P02027S	Product already running: Start terminating	Another copy of the tableBASE Process Manager is already running. It needs to be shut down before you can start it up again.

Error Code	Text	Meaning / Instructions
DK1P02028S	#FINDTPM macro failed - terminating, RC= <return code #>	
DK1P02029S	#FINDTPM macro failed - terminating, RC= <return code #>	
DK1P02030S	#FINDTPM macro defining TPVM compat failed, terminating, RC= <return code #>	
DK1P02032S	TPM CATALOG INVALID - TERMINATING	
DK1P02034S	Define for TPVM compat failed - terminating, RC= <return code #>	
DK1P02036S	#FINDTPM macro starting TPVM failed - terminating, RC= <return code #>	
DK1P02037S	Request Mgr failed - terminating	
DK1P02038S	TPM Request Mgr not in WAIT state - TPM terminating	
DK1P02039W	Unexpected POST to Request Mgr - terminating	
DK1P02040W	AUTOSTART for TPVM ,<tpvm name> failed	
DK1P02041I	TPVM <tpvm name> RE-LINKED TO TPM	The TPVM identified has been reconnected to the TPM.
DK1P02042S	#FINDTVC macro failed - terminating, RC = <return code #>TPVM = <tpvm name>	
DK1P02044I	TB <release name>: INITIALIZATION PROCESSING COMPLETED	Information only.
DK1P02046I	TPM detected Request Mgr completion- terminating	Information only.
DK1P02048I	TPM detected operator STOP command - terminating	Information only.
DK1P02050S	TPM detected bad POST - terminating, RM ECB = <control block #>, COMM ECB = <control block #>	
DK1P02052E	TPM unable to obtain service ENQ - AUTOSHUT not performed	
DK1P02054S	#FINDTPM macro failed - AUTOSHUT not performed, RC= <return code #>	
DK1P02056S	TPM Request Mgr not in WAIT state: AUTOSHUT cancelled	
DK1P02058E	AUTOSHUT for TPVM <tpvm name> failed	
DK1P02060S	#FINDTVC macro failed - terminating, RC = <return code #> TPVM = <tpvm name>	
DK1P02062S	#FINDTPM macro failed - TPM terminating; PVTE flags invalid, RC=<return code #>	
DK1P02063S	TPM "PM" ATTACH failed - RC= <return code #>	
DK1P02064S	VTS PROC can not be used to start a TPVM	
DK1P02065S	DETACH PM failed; RC= <return code #>	
DK1P02066E	DETACH macro failed - TPM termination continuing, RC= <return code #>	
DK1P02067S	License error - TPM terminating	
DK1P02068E	MODESET macro failed - TPM termination continuing, RC= <return code #>	
DK1P02069S	Corrupted navigation (#FINDTPM macro failed) - terminating	
DK1P02070I	TB <release name>: Processing completing	Information only.
DK1P02071S	TPM failed - unrecoverable error; terminating	
DK1P02072S	P1061: INITIAL ESTAEX ON FAILED - TERMINATING	
DK1P02074S	P1061: INPUT PARM ERROR - TERMINATING	
DK1P02076S	Program not authorized - terminating	Your load module is not in an authorized library.

Error Code	Text	Meaning / Instructions
DK1P02078S	MODESET macro failed - terminating	
DK1P02080S	SERVICE ENQ ERROR - TERMINATING	
DK1P02082W	WARNING: CURRENTLY ACTIVE DKL PRODUCTS: <product names>	
DK1P02084S	STORAGE macro OBTAIN failed - terminating, RC = <return code #>	
DK1P02086S	DIV macro IDENTIFY failed - terminating, RC =<return code #>	
DK1P02088S	DIV macro ACCESS failed - terminating, RC = <return code #>	
DK1P02090S	TPM LDS NOT INITIALIZED - TERMINATING	Your LDS for starting up the TPM has not been configured. See the *tableBASE Installation Guide, Release 6.0.3* on how to configure it.
DK1P02092S	DSPSERV macro CREATE failed, RC = <return code #>	
DK1P02094S	ALESERV macro CREATE failed, RC = <return code #>	
DK1P02098S	DIV macro MAP failed, RC = <return code #>	
DK1P02100S	#FINDTPM macro failed - terminating, RC = <return code #>	
DK1P02101S	TPM pointer found with PRISTINE start - terminating	
DK1P02110S	IAZXJSAB macro failed - terminating, RC = <return code #>	
DK1P02112S	STORAGE macro OBTAIN failed - terminating, RC =<return code #>	
DK1P02114S	STORAGE macro OBTAIN failed - terminating, RC = <return code #>	
DK1P02115S	Invalid TPMDSECT address - terminating	
DK1P02116S	Failure setting recovery <state> IN <routine name> -TPM terminating, RC= <return code #>	<state> = ON or OFF
DK1P02117S	Corrupted Navigation (<target control block>)	
DK1P02118E	DIV macro, <macro name> failed - termination continuing, RC=<return code #>	
DK1P02120E	ALESERV macro DELETE failed - termination continuing, RC=<return code #>	
DK1P02121I	Recovery detected /C (cancel) command	Information only.
DK1P02122E	DSPSERV macro DELETE failed - termination continuing, RC=<return code #>	
DK1P02123S	Unexpected unrecoverable error	
DK1P02124E	STORAGE macro RELEASE for PVT failed - termination continuing, RC=<return code #>	
DK1P02125S	PC INIT failed - terminating, RC= <return code #>	
DK1P02126S	P1062: INITIAL ESTAEX ON FAILED - TERMINATING	
DK1P02128S	P1062: INITIAL FINDTPM FAILED - TERMINATING	
DK1P02130S	STORAGE macro OBTAIN failed for PC table - terminating, RC = <return code #>	
DK1P02131I	PREVIOUS LXRES VALUE FOUND - LX = <LX value>	Information only.
DK1P02132S	LXRES macro failed - terminating, RC = <return code #>	
DK1P02133I	LXRES VALUE BEING USED = <LX value>	Information only.
DK1P02134S	ETCRE macro failed - terminating, RC = <return code #>	

Error Code	Text	Meaning / Instructions
DK1P02136S	ETCON macro failed - terminating, RC = <return code #>	
DK1P02138S	LOAD macro failed - terminating, RC = <return code #>	
DK1P02140S	STORAGE macro OBTAIN for PC module failed - terminating, RC = <return code #>	
DK1P02142S	#FINDTPM macro failed - terminating, RC = <return code #>	
DK1P02144S	LOAD macro failed - terminating, RC = <return code #>	
DK1P02146S	STORAGE macro OBTAIN for PC module failed - terminating, RC = <return code #>	
DK1P02147S	#FINDTPM macro failed - terminating, RC = <return code #>	
DK1P02148S	Unexpected unrecoverable error	
DK1P02149I	Recovery detected /C (cancel) command	Information only.
DK1P02150	TPVM Start terminated or timed out; Review PROC, JCL or systems resources	The TPVM start could not be completed either due to errors in your user PROC or JCL or because there are insufficient system's resources to start another task. Review the TPVM job for JCL errors and check with your systems administrator for systems resource problems.
DK1P02151W	Initial recovery set on failed	
DK1P02152W	Failure setting recovery on, ROUTINE = <routine name #> RC = <return code #>	
DK1P02153W	Failure setting recovery off, ROUTINE = <routine name>, RC = <return code #>	
DK1P02154S	Initial #FINDTPM macro failed - terminating	
DK1P02155S	#FINDTPM macro failed, ROUTINE = <routine name>, RC = <return code #>	
DK1P02156S	TPM catalog not found or invalid	
DK1P02157E	Caller request code invalid, CODE = <code name>	
DK1P02158E	Request block data length incorrect	
DK1P02159E	Invalid TPVM name: all zeroes	The TPVM name cannot be all zeroes.
DK1P02160W	DEFINE requested for existing TPVM , <tpvm name>	
DK1P02161W	TPVM max entries exceeded in TPM catalog; DEFINE failed for TPVM <tpvm name>	There are a maximum of 14 entries allowed in the TPM catalog for named TPVMs and you have reached this limit.
DK1P02162S	DIV macro <macro name> failed, SERVICE = <service name>, RC = <return code #>, RS = <reason code #>	
DK1P02163W	START requested for running TPVM <tpvm name>	The TPVM that was requested to be started is already running.
DK1P02164E	START requested for undefined TPVM <tpvm name>	The TPVM that was requested to be started has not been defined.
DK1P02165E	START TPVM failed for TPVM <tpvm name>, RC=<return code #>	The TPVM start failed with the specified return code. Contact tableBASE technical support for further information.
DK1P02166E	Start TPVM ASCRE macro failed, TPVM =<tpvm name>, RC = <return code #>, RS = <reason code #>	
DK1P02167E	START TPVM terminated: TPVM=<tpvm name>, RC=<return code #>	
DK1P02168E	TPVM compat cannot be shut down using TPDRIVER SHUTDOWN command; use /P	The *compat* TPVM cannot be shutdown using TPDRIVER.

Error Code	Text	Meaning / Instructions
DK1P02169E	SHUTDOWN requested for TPVM not currently active, TPVM = <tpvm name>	The TPVM that was requested to be shutdown is not running.
DK1P02170E	SHUTDOWN request failed for TPVM <tpvm name>, TPVM not in WAIT state	
DK1P02171E	TPVM compat cannot be deleted	The *compat* TPVM definition cannot be deleted from the TPM catalog..
DK1P02172E	DELETE requested for undefined TPVM <tpvm name>	The TPVM that was requested to be deleted is not defined.
DK1P02173E	DELETE requested for running TPVM <tpvm name>	A TPVM that is running cannot be deleted.
DK1P02174E	TPVM <tpvm name> exceeds max TPVMs started in common memory	
DK1P02175I	TPM IS PROCESSING A REQUEST TO <action> TPVM <tpvm name>	Information only.
DK1P02176E	LDS in use; TPVM <tpvm name> not started	The LDS that is required to start the TPVM is already in use by another task.
DK1P02177I	Recovery detected /C (cancel) command	Information only.
DK1P02178I	LDS DSN=<LDS dataset name>	Information only.
DK1P02179S	Unexpected unrecoverable error	
DK1P02180S	Initial set recovery on failed - terminating	
DK1P02181S	P1070: INPUT PARM ERROR - TERMINATING	
DK1P02182S	Program not authorized - terminating	The program is not in an authorized library.
DK1P02183S	MODESET macro failed - terminating	
DK1P02184S	Corrupted navigation (<tpvm name>)	
DK1P02185S	TPM CONTROL BLOCK NOT FOUND	
DK1P02188I	Recovery detected /C (cancel) command	Information only.
DK1P02189S	TPM failed - unrecoverable error; terminating	
DK1P02190S	No parms received - terminating	
DK1P02200I	INITIALIZING TPVM <tpvm name>	Information only.
DK1P02201I	TPVM <tpvm name> IS RUNNING	Information only.
DK1P02202I	Operator STOP command detected for TPVM <tpvm name>	Information only.
DK1P02203I	TPVM <tpvm name> SHUTTING DOWN	Information only.
DK1P02204E	PROBLEM SAVING CATALOG FOR <tpvm name>	
DK1P02205E	PROBLEM UNMAPPING CATALOG FOR <tpvm name>, RC=<return code #>	
DK1P02206E	PROBLEM REMOVING CATALOG ACCESS FOR <tpvm name>, RC=<return code #>	
DK1P02207E	PROBLEM REMOVING CATALOG IDENTITY FOR <tpvm name>, RC=<return code #>	
DK1P02208E	Cannot delete ALET for <tpvm name>, RC=<return code #>	
DK1P02209E	Cannot delete dataspace for <tpvm name>, RC=<return code #>	
DK1P02210E	Cannot detach Request Manager for <tpvm name>, RC=<return code #>	
DK1P02211E	AUTOSTART for TPVM <tpvm name> ERROR STARTING VTS <vts name>	

Error Code	Text	Meaning / Instructions
DK1P02212E	AUTOSTART for TPVM <tpvm name>, VTS <vts name> FAILED, RC=<return code #>	
DK1P02213E	AUTOSHUT for TPVM <tpvm name>, VTS <vts name> FAILED, RC=<return code #>	
DK1P02214S	TPVM INIT failed - cannot open DDN DK1SNAP	
DK1P02215S	TPVM INIT failed - Program not authorized	
DK1P02216S	TPVM INIT failed - cannot change program mode	
DK1P02217S	TPVM INIT failed - TPM control block not found	
DK1P02218S	TPVM INIT failed - TPM not running	
DK1P02219S	TPVM INIT failed - error in QEDIT macro	
DK1P02220S	TPVM <tpvm name> INIT failed - cannot attach Request Manager	
DK1P02221S	TPVM <tpvm name> INIT failed - cannot determine shutdown type	
DK1P02222W	Failure setting recovery off	
DK1P02223S	TPVM failed - unrecoverable error	
DK1P02224I	TPVM <tpvm name> stopped running	Information only.
DK1P02225E	Error during shutdown - VTS not defined	
DK1P02226S	Messaging init request failed	
DK1P02227I	VTS <vts name> re-linked to TPVM <tpvm name>	Information only.
DK1P02231E	Problem saving catalog for <tpvm name>, RC=<return code #>	
DK1P02232S	DIV macro error, TPVM=<tpvm name>, RC=<return code #>	
DK1P02233S	Cannot allocate catalog dataspace for <tpvm name>, RC=<return code #>	
DK1P02234S	Cannot allocate ALET for <tpvm name> catalog, RC=<return code #>	
DK1P02235S	Cannot find TPM control block for <tpvm name>	
DK1P02236S	TPVM <tpvm name> exceeds max TPVMs started in common memory	
DK1P02237S	Cannot allocate space for <tpvm name> REQUEST BLOCK	
DK1P02238S	Cannot allocate space for <tpvm name> GCA	
DK1P02239S	TPVM <tpvm name> cannot use catalog for TPVM <tpvm name>	
DK1P02240E	Problem unmapping catalog for <tpvm name>, RC=<return code #>	
DK1P02241E	Problem removing catalog access for <tpvm name>, RC=<return code #>	
DK1P02242E	Problem removing catalog identity for <tpvm name>, RC=<return code #>	
DK1P02243E	Cannot delete dataspace for <tpvm name>, RC=<return code #>	
DK1P02244E	Cannot delete ALET for <tpvm name>, RC=<return code #>	
DK1P02245S	TPVM failed - unrecoverable error'	
DK1P02246S	Error loading PC for <tpvm name>, RC=<return code #>	

Error Code	Text	Meaning / Instructions
DK1P02247S	Error obtaining PC space for <tpvm name>, RC=<return code #>	
DK1P02248S	Error deleting PC for <tpvm name>, RC=<return code #>	
DK1P02249W	Failure setting recovery on, RC=<return code #>	
DK1P02250W	Failure setting recovery off, RC=<return code #>	
DK1P02251S	SHOWCB failed for LDS, RC = <return code #>	
DK1P02252S	TPVM LDS size allocation error; LDS size is: <LDS size>	
DK1P02253I	Recovery detected /C (cancel) command	Information only.
DK1P02261S	LDS for <tpvm name> was not empty as expected	
DK1P02262S	Definition not found for TPVM <tpvm name> in TPM catalog	The TPVM definition cannot be found in the TPM catalog.
DK1P02263S	Cannot find TPM control block for <tpvm name>	
DK1P02264S	TPVM <tpvm name> failed - unrecoverable error	
DK1P02265W	Failure setting recovery on, RC=<return code #>	
DK1P02266W	Failure setting recovery off, RC=<return code #>	
DK1P02267I	Recovery detected /C (cancel) command	
DK1P02301S	Cannot find VTS control block (GCA) for <tpvm name>	
DK1P02302S	Cannot find the catalog for <tpvm name>	
DK1P02303S	(1,C,8),' can not provide service requested opcode=<op code #>	
DK1P02304E	Invalid VTS definition under <tpvm name>, length=<length #>	
DK1P02305E	Invalid ALIAS definition under <tpvm name>, LENGTH=<length #>	
DK1P02306E	Maximum VTS definitions exceeded in <tpvm name> catalog	The maximum of 64 VTS definitions in the TPVM catalog has been reached,
DK1P02307E	Maximum ALIAS definitions exceeded in TPVM catalog <tpvm name>'	The maximum of 64 alias definitions in the TPVM catalog has been reached,
DK1P02308W	DEFINE requested for existing VTS <vts name> under <tpvm name>	The VTS definition already exists in the TPVM catalog.
DK1P02309E	DEFINE requested for existing <vts name> under <tpvm name>	The VTS definition already exists in the TPVM catalog.
DK1P02310E	An ALIAS cannot be used under TPVM compat	Information only.
DK1P02311E	No VTS associated with ALIAS <alias name> under <tpvm name>	
DK1P02312E	<vts name> IS RUNNING AS <alias name> under <tpvm name>	The VTS is running as the alias name specified.
DK1P02313E	Cannot find VTS definition <vts name> under <tpvm name>	The VTS definition cannot be found in the TPVM catalog.
DK1P02314E	Cannot find ALIAS definition <alias name> under <tpvm name>	The alias definition cannot be found in the TPVM catalog.
DK1P02315E	VTS <vts name> under <tpvm name> is not running	The VTS is not running under this TPVM.
DK1P02316E	ALIAS <alias name> under <tpvm name> is not running'	The alias name is not running under this TPVM.
DK1P02317E	VTS <vts name> under <tpvm name> is not Read-Only	The VTS specified is not in READ-ONLY mode.
DK1P02318W	VTS <vts name> under <tpvm name> is currently running	The VTS is already running under this TPVM.
DK1P02319E	ALIAS <alias name> under <tpvm name> is currently running	The alias name is already running under this TPVM.
DK1P02320S	VTS <vts name> under <tpvm name> is not in a wait state	

Error Code	Text	Meaning / Instructions
DK1P02321E	ALIAS <alias name> under <tpvm name> cannot be switched to ALIAS <alias name>	An alias name cannot be switched to another alias name.
DK1P02322E	VTS <vts name> under <tpvm name> is already associated with this ALIAS	The alias name is already associated with the VTS specified.
DK1P02323E	VTS <vts name> under <tpvm name> is associated with another ALIAS	The VTS specified is already associated with another alias name.
DK1P02324S	DIV macro error, TPVM=<tpvm name>, RC=<return code #>	
DK1P02325E	LDS in use, VTS <vts name> under TPVM <tpvm name> not started	The LDS that is required to start the TPVM is already in use by another task. See message 2326 for the name of the LDS.
DK1P02326E	LDS DSN=<LDS dataset name>	See message 2325 for the related message.
DK1P02327	VTS <vts name> under <tpvm name> was started by the VTS Agent	This VTS was started using VTSAgent.
DK1P02329I	Recovery detected /C (Cancel) command	Information only.
DK1P02330S	TPVM control block not found	
DK1P02331S	Service failed - unexpected unrecoverable error	
DK1P02332W	Failure setting recovery on, RC=<return code #>	
DK1P02333W	Failure setting recovery off, RC=<return code #>	
DK1P02335I	VTS <vts name> under <tpvm name> is now known as <alias name>	Information only.
DK1P02336I	VTS <vts name> under <tpvm name> no longer known as <alias name>	Information only.
DK1P02340I	<tpvm name> processing request to <action> <vts name>	Information only.
DK1P02341I	VTS Start terminated or timed out; Review PROC, JCL or systems resources	Problem with either PROC JCL or system resources: **PROC or JCL error**: look for a failed VTS job which would specify the reason for the failure. **Insufficient systems resources**: the start operation likely timed out, there would be no failed job to indicate this. Retry the start operation and if still unsuccessful, speak to your systems programmer regarding allocating more resources to your system.
DK1P02343I	P2063 TRMEXIT: Error in VTS Prolog	
DK1P02344I	P2063 TRMEXIT: PROC error - possible JCL error	
DK1P02345I	TPVM PROC cannot be used to start a VTS	
DK1P02400I	Initializing <xxxxxx> VTS <vts name> under <tpvm name>	Information only.
DK1P02401I	<xxxxxx> VTS <vts name> is running under <tpvm name>	Information only.
DK1P02402I	Operator cancelled VTS <vts name> under <tpvm name>	Information only.
DK1P02403S	Forced shutdown requested by recovery routine	
DK1P02404I	Shutdown request accepted for VTS <vts name> under <tpvm name>	Information only.
DK1P02405S	SNAP open failed	
DK1P02406S	MSG init failed	
DK1P02407S	TBLBASE init error	
DK1P02408S	TBLBASE term error	
DK1P02409S	Cancelled S222	
DK1P02410S	VTS start failed <vts name> - Program not authorized, RC=<return code #>	

Error Code	Text	Meaning / Instructions
DK1P02411S	VTS START failed <vts name> - Cannot change program mode, RC=<return code #>	
DK1P02412S	VTS START failed - TPVM name invalid name as supplied in hex <hex #>	
DK1P02413S	Cannot obtain ALET for <tpvm name>, RC=<return code #>	
DK1P02414S	Cannot find TPVM control block for <tpvm name>	
DK1P02415S	Cannot find catalog for <vts name> under <tpvm name>	
DK1P02416E	Cannot find VTS definition for <vts name>	
DK1P02417S	No LDS found for Read-Only VTS <vts name> under <tpvm name>	A READ-ONLY VTS must have an LDS associated with it.
DK1P02418S	LDSTSR DD statement not found or incorrect for <vts name> under <tpvm name>	The DD statement for the LDS is not found or incorrect. Check your VTS PROC.
DK1P02419S	VTS START failed under TPVM <tpvm name> - Error in QEDIT macro, RC=<return code #>	
DK1P02420S	Cannot find VTS control block for <vts name> under <tpvm name>	
DK1P02421S	Cannot load TBLBASE for VTS <vts name> under <tpvm name>	
DK1P02422S	RECOVERY ROUTINE DETECTED ERROR IN VTS PROLOG CODE	
DK1P02423S	VTS STARTUP FAILED - UNRECOVERABLE ERROR	
DK1P02424W	Failure setting recovery off, RC=<return code #>	
DK1P02425	ECB unknown	
DK1P02426	VTS <vts name> is running as ALIAS <alias name>	Information only.
DK1P02427	PVT Entry not found - VTS Startup terminated	
DK1P02500S	Messaging not available	
DK1P02590I	TPM Catalog configured with <count #> pages	Information only.
DK1P02591E	Configuration Module <module name> not loaded; RC=<return code #>	
DK1P02592E	Internal name <module name> does not match config module <module name>	
DK1P02593E	Version <version #> of config module <module name> does not match executing version <version #>	
DK1P02594E	<macro name> error for DDN TPMCAT, RC=<return code #>	
DK1P02595E	DISP for DDN TPMCAT must be OLD or NEW	
DK1P02596E	LDS DDN TPMCAT must be VSAM	The LDS for the TPM catalog must be a VSAM dataset.
DK1P02597E	Error in IBM macro <macro name>; RC=<return code #>, RS=<reason code #>	
DK1P02598E	LDS DDN TPMCAT too small at <# pages> pages; minimum is <# pages>	The LDS for the TPM catalog is too small.
DK1P02599S	TPM Catalog not configured due to preceding errors	
DK1P02601	<client name>	Information only.
DK1P02602	is not Licensed to use tableBASE Process Manager	You are not licensed to use tableBASE Process Manager.
DK1P02603	tableBASE Process Manager is Licensed to	Information only.
DK1P02604	Your "Process Manager" License expires on <date>	Information only.

Error Code	Text	Meaning / Instructions
DK1P02605	The PVT entry is not available	
DK1P02606	Your "Process Manager" License ends today	Information only.
DK1P02607	WARNING: Your "Process Manager" License will end in <number> days	Information only.
DK1P02608	Your "Process Manager" License expires in <number> days	Information only.
DK1P02609	NOTICE: Your "Process Manager" License has expired	Information only.
DK1P02610	is no longer Licensed to use tableBASE Process Manager	Information only.
DK1P02611	"Process Manager" must use an authorized LOAD library	Your library is not authorized.
DK1P02612	Licensing function found error in customer ID	
DK1P02613	Licensing function found error in product code	
DK1P02614	Licensing function found error; RC= <return code #>	
DK1P02615	Failure setting recovery on; RC= <return code #>	
DK1P02616	Failure setting recovery off; RC= <return code #>	
DK1P02617	Recovery detected /C (cancel) command	Information only.
DK1P02701S	TPVM <tpvm name> failed - Program not authorized	
DK1P02702S	TPVM <tpvm name> failed - cannot change program mode	
DK1P02703S	Cannot find control block for <tpvm name>	
DK1P02704S	Request Manager failed - unrecoverable error	
DK1P02705I	Recovery detected /C (cancel) command	Information only.
DK1P02706W	Failure setting recovery on, RC<return code #>	
DK1P02707W	Failure setting recovery off, RC=<return code #>	
DK1P02708S	Corrupted navigation path	
DK1P02709S	TPM control block not found	
DK1P02710S	TPM is not running	The tableBASE Process Manager PC Server is not running.
DK1P02711S	TPVM is not running	The TPVM is not running.
DK1P02800E	Error opening DDN TPDRCNTL, C error = <error code>	Error opening the input file for TPDRIVER.
DK1P02801E	Error reading input file, C error = <error code>	
DK1P02802E	Parameter value too long (<item name>)	The parameter value is too long.
DK1P02803E	Invalid command modifier (must be VTS or TPVM or ALIAS)	The valid TPDRIVER command modifiers are: VTS, TPVM or ALIAS.
DK1P02804E	Wrong input format - command incomplete and an EOF was reached	
DK1P02805E	Input command too long	The command line is too long.
DK1P02806E	Wildcard not allowed (<parameter name>)	Illegal wildcard use.
DK1P02807E	Invalid format for wildcard	The wildcard format is invalid.
DK1P02808E	Invalid format for default	
DK1P02809E	Invalid boolean value (<parameter name>)	Must be Y or N; nothing else is valid.
DK1P02810E	Invalid numeric value (<parameter name>)	Must be NUmeric value; i.e., 100K, 300000. Nothing else is valid.
DK1P02811E	Numeric value is out of bounds (<parameter name>)	Value must be within legal bounds. (For TSRSIZE, must be between 40K and 2G.)
DK1P02812E	Invalid value (RECOVERY)	

Error Code	Text	Meaning / Instructions
DK1P02813E	Missing parameter (<parameter name>)	The parameter name specified in parenthesis is required.
DK1P02814E	A parameter is invalid with this command (<parameter name>)	Command is not compatible with parameter used.
DK1P02815E	Wrong input format - cannot parse command	The input format is invalid.
DK1P02816E	Wrong input format	
DK1P02817E	Invalid command	The command is invalid.
DK1P02818E	Parameter is too long	The length of the parameter is too long.
DK1P02819E	Too many parameters in command, max is <count #>	Maximum is 20.
DK1P02820E	Invalid command modifier (must be VTS or TPVM or ALIAS)	The valid TPDRIVER command modifiers are: VTS, TPVM or ALIAS.
DK1P02821E	Command modifier invalid for this command	The command modifier specified for this command is invalid.
DK1P02822E	Invalid format for parameter <parameter name> (expected format: NAME1=Value1)	The format for the parameter should be as specified in paranthesis.
DK1P02823E	Invalid name for parameter <parameter name>	Parameter used is invalid.
DK1P02824E	Memory allocation failed	
DK1P02827W	TSRSIZE parameter will be ignored at VTS startup when an LDS is used	If an LDS is used for a VTS and the TSRSIZE parameter is also specified, it will be ignored and the actual TSRSIZE at startup will be that of the LDS.
DK1P02828W	TSRSIZE parameter will be ignored at VTS startup when TSRACCESS mode is Read-Only	If Read-Only mode is used for a VTS and the TSRSIZE parameter is also specified, it will be ignored and the actual TSRSIZE at startup will be that of the LDS.
DK1P02829E	LDS parameter required when TSRACCESS mode is Read-Only	If Read-Only mode is used for a VTS an LDS is required.
DK1P02830E	Invalid TSRACCESS value	The value specified for the TSRACCESS parameter is invalid. See *tableBASE Batch Utilities Guide* for details.
DK1P02831E	Invalid string format	
DK1P02832E	<command> command failed with error <error code> please review messages in <item> and <item> job	If there is no <item> job in which to search, see the specified error code in this table.
DK1P02900I	Process to remove and clean the product has started	Information only.
DK1P02901I	PVTE is full or pointer is corrupt	The PVT entry is full or the pointer to the PVTE has been corrupted. Contact tableBASE Technical Support for assistance.
DK1P02902I	Product vector table indicates product is not running	Information only
DK1P02903I	Product vector table indicates product is running	Information only.
DK1P02906I	Process to remove and clean has been run before	Information only
DK1P02907I	The product vector table pointer is corrupt	Contact tableBASE Technical Support for assistance.
DK1P02908W	Reply is incorrect. Halt is assumed	The response given to the WTO is incorrect, The process will be stopped.
DK1P02909I	Request to halt this process has been accepted	Information only.
DK1P02910I	Request to create a new PVT pointer has been accepted	Information only.
DK1P02911I	ENQ is not available. Process will terminate	
DK1P02912S	Messaging init request failed	
DK1P02913S	Program not authorized. Cannot proceed	
DK1P02914S	MODESET macro failed - terminating	

Error Code	Text	Meaning / Instructions
DK1P02915S	Error found between IBM control blocks CVT and CAT. Cannot proceed	Contact tableBASE Technical Support for assistance.
DK1P02916S	DKL slot in IBM CAT (Customer Anchor Table) is corrupt - zeroing entry	Contact tableBASE Technical Support for assistance.
DK1P02917I	Pointer to TPM control block is corrupt	Contact tableBASE Technical Support for assistance.
DK1P02918I	The TPM control block is corrupt	Contact tableBASE Technical Support for assistance.
DK1P02919I	Pointer indicates TPM control block does not exist	Contact tableBASE Technical Support for assistance.
DK1P02920I	TPVM control block is corrupt	Contact tableBASE Technical Support for assistance.
DK1P02921I	Process to remove and clean up <tpvm name> has started	Information only.
DK1P02922I	Forcing down TPVM <tpvm name>	Information only.
DK1P02923I	Pointer to the VTS control blocks is corrupt	Contact tableBASE Technical Support for assistance.
DK1P02924I	Shutting down and cleaning up <vts name> under <tpvm name>	Information only
DK1P02925I	Stopping PGM named: <program name> ASIDX=<asidx name>	Information only
DK1P02926I	PGM named: <program name> did not stop	
DK1P02927I	Corrupt or invalid PGM name - HEX value: <hex value #>	
DK1P02928I	Forcing down the TPM	Information only
DK1P02929I	Service ENQ <major name>, <minor name> in effect	Information only.
DK1P02930I	Process will halt based on the reply	Information only.
DK1P02931I	Process will continue as requested	Information only.
DK1P02932W	Failure setting recovery off, RC=<return code #>	
DK1P02933I	Recovery detected /C (Cancel) command	
DK1P02934S	Unexpected unrecoverable error	
DK1P02999I	FORCE ended	Information only.
DK1P03000E	Invalid command modifier (must be VTS or TPVM or ALIAS)	
DK1P03001E	Unaddressable Function	
DK1P03002E	Invalid command	No such command.
DK1P03003E	Invalid parameter list (TOKEN)	
DK1P03004E	Invalid token	
DK1P03006E	Invalid parameter list (OBJECT)	
DK1P03007E	Invalid command modifier (must be VTS or TPVM or ALIAS)	
DK1P03008E	Invalid parameter list (OBJECT NAME)	
DK1P03009E	Invalid value in one of the parameters	
DK1P03010E	Invalid parameter list (STRUCTURE)	
DK1P03011E	Structure is too large	
DK1P03012E	TBOPT/PARM34 length mismatch	
DK1P03013E	STACK POST routine failed	
DK1P03014E	TPM is not accessible	
DK1P03015E	TPVM is not accessible	
DK1P03016E	TPVM name does not match name in Request Block	
DK1P03017E	VTS name does not match name in Request Block	
DK1P03018E	Missing or invalid PROC name	PROC name is required for ths command and is either missing or the name does not exist in your PROCLIB.

Error Code	Text	Meaning / Instructions
DK1P03019E	Invalid AUTOSTART value used (must be Y or N)	Value for the AUTOSTART parameter must be Y or N.
DK1P03020E	Invalid TSRACCESS value used (must be RO or RW)	Value for the TSRACCESS parameter must be RO or RW
DK1P03021E	Invalid AUTOSHUT value used (must be Y OR N)	Value for the AUTOSHUT parameter must be Y OR N
DK1P03022E	TSRSIZE is less than minimum of 40K	Minimum value for TSRSIZE is 40K.
DK1P03023E	TSRSIZE is larger than maximum of 2G	Maximum value for TSRSIZE is 2G.
DK1P03024E	Request Block data is too short	
DK1P03025E	TPVM NOT FOUND OR NOT RUNNING	The TPVM cannot be found or it is not running.
DK1P03026	Product has no License; TPDRIVER terminated	You are not licensed to run tableBASE Process Manager
DK1P03027	The TPM control block is corrupt	Contact tableBASE Technical Support for assistance.
DK1P03040	Unaddressable object	
DK1P03041E	Invalid object	
DK1P03042I	Object name not found	
DK1P03043I	TPM data layout not valid	
DK1P03044E	TPVM data layout not valid	
DK1P03045E	VTS data layout not valid	
DK1P03046E	TPM catalog dataspace error	
DK1P03047E	TPVM catalog dataspace error	
DK1P03081E	P0100: PVT not found	
DK1P03082E	P0100: TPM not running	
DK1P03083E	P0100: ENQ error	
DK1P03084E	P0100: No parms passed	
DK1P03085E	P0100: TPM not found	
DK1P03086E	P0100: TPVM not found or not running	
DK1P03087E	P0100: TPVM not running	
DK1P03088E	P0100: Request Manager not waiting	
DK1P03089E	P0100: STACK POST routine failed	
DK1P03200E	DK1T03200E: no parms passed to message routine	
DK1P03201E	DK1T03201E: message INIT STORAGE OBTAIN error	
DK1P03202E	DK1T03202E: Message INIT OPEN error	
DK1P03203E	DK1T03203E: Target message not found	
DK1P03204E	DK1T03204E: Repository message length error	
DK1P03205E	DK1T03205E: Max nmbr of sub elements exceeded	
DK1P03206E	DK1T03206E: Max message length exceeded (text)	
DK1P03207E	DK1T03207E: Input data not half or full word	
DK1P03208E	DK1T03208E: Input substitution data exceeds max size	
DK1P03209E	DK1T03209E: Max msg length exceeded (SUBSTITUTION)	
DK1P03210E	DK1T03210E: Type X substitution data length error	
DK1P03211E	DK1T03211E: Message PUT failed	
DK1P03212E	DK1T03212E: Message FILE CLOSE error	
DK1P03213E	DK1T03213E: STORAGE RELEASE error	

Error Code	Text	Meaning / Instructions
DK1P03214E	DK1T03214E: Max WTO message length exceeded	

Note: Abnormal terminations between 99 and 1000 or over 1099, are tableBASE errors and should be brought to the immediate attention of your tableBASE Administrator.

tableBASE messages

Messages issued from the tableBASE components, such as the VTS Agent, CICS, batch and others, vary from function to function, and are listed in the table below. These messages can usually be viewed on the JESMSGLG. Any Severe Errors (S) messages should be referred to tableBASE Customer Support. Messages are in the format DK1nnnnnA where DK1 is the prefix, followed by a five digit numeric code and an error code letter. The error code letter indicates the type and severity of the error message:

E—Error (usually indicates user error)
I—Information
W—Warning
A—Action (user action is required)
S—Severe (no further processing is possible)

Table A-3: Messages and error codes for VTS Agent, CICS, batch, etc.

Msg Code	Text	Meaning / Instructions
DK100049S	NO CMDAREA IN CALL TO TBLBASE. ABENDING.	There was no command area supplied. Ensure that a command area is supplied for the tableBASE call.
DK100200S	Initialization failed	
DK100201E	PC Service unavailable; Required for VTS-TSR use	
DK100201I	tableBASE PC SERVER UNAVAILABLE	Information only.
DK100202I	tableBASE <version> in initializing for <system license info>	Information only.
DK100203I	tableBASE <type> license expires on <date>	Information only.
DK100207E	DK1TX071 not loaded; System Exits disabled	
DK100207E	DK1TX072 not loaded; User Exits disabled	
DK100210E	Module DK1TNUCL not loaded	
DK100212E	MULTITASKING requested or VTS Agent; see message DK100201E	
DK100213E	Error in tableBASE PC server	
DK100214E	CMA GETMAIN failed	
DK100221E	I/O ERROR:	
DK100222E	QSAM ERROR: FUNCTION=xxxxxxxx,FILE=ffffffff	
DK100224E	I/O PROBLEM: FILE=xxxxxxxx FUNC=ffffffff RESP=dddd RESP2=dddd	
DK100226I	Please change DD TBOPTV to TBOPT	Information only.
DK100227W	JOB <jobname> IN vvvvvvvv WAITING FOR nnnn SECONDS FOR TABLE <tablename>	
DK100228W	JOB <jobname> IN <VTSname> WAITING FOR <number> SECONDS FOR TABLE <tablename>	Job is waiting for VTS-TSR table enqueue--not received yet.
DK100230I	TBOPT Processing	Information only.
DK100231I	keyword=value.....	Information only.
DK100232E	Invalid value	
DK100233E	Missing keyword	

Msg Code	Text	Meaning / Instructions
DK100234E	Invalid keyword	
DK100235E	Value exceeds maximum allowed	
DK100236E	Keyword not allowed in this environment	
DK100237E	Missing "="	
DK100238E	Invalid quoted string	
DK100239E	Value field too wide	
DK100240E	Value less than minimum allowed	
DK100241E	Missing value	
DK100242E	Keyword (or alias) previously specified	
DK100243E	Option is not supported in this version	Your organization is not authorized to use the optional product; please contact Technical Support.
DK100244I	MAXNMTAB set to nnnnnn	Information only.
DK100245W	VTSONLY not valid with LIB entries	
DK100246W	Invalid VTS prefix for LIBnn=xxxxxxxx	
DK100247W	Empty value: LIBxx - LIByy	
DK100248W	LIBxx exceeds max allowed mm	
DK100249W	MULTOPNX=N is not valid	
DK100250I	TBOPT Parameters: (*= If default)	Information only.
DK100251I	keypwrd=value.....	Information only.
DK100252I	<<<<<<< TBOPT Done >>>>>>	Information only.
DK100253I	<message>	Information only.
DK100260I	DK1TX066 Exit Manager Initializing	Information only.
DK100262W	Exit xxxxxxxx not loaded	
DK100263I	Exit Manager Shutting down; Exits active	Information only.
DK100264I	DK1TX066 Exit Manager Shutdown complete	
DK100265W	DK1TC073 NOT LOADED	
DK100270E	Insufficient storage to process strobe data; strobe suppressed.	
DK100271E	Error on CICS WRITE JOURNALNUM; Strobe logging suppressed	
DK100272E	WRITE JOURNALNUM RESP=1234, RESP2=1234	
DK100273E	Dynalloc of strobe report failed. Strobe suppressed.	
DK100275E	Module DK1TRSTA not found.	
DK100277I	>> ESTAI -NO CMA	Information only.
DK100280E	QSAM ERROR: FUNCTION=xxxxxxxx,FILE=ffffffff	
DK100281E	STROBE OUTPUT SUPPRESSED	
DK100282E	I/O ERROR:	
DK100292E	DEBUG TRACE requested but not started	
DK100295A	Specify character for PC server Named Token	
DK100299	DK1T1153 PRE-XDC	
DK100299	DK1T1153 XDC ENABLED	

Msg Code	Text	Meaning / Instructions
DK100300S	SYSTEM FAILURE: xxxxxxxx Code=cccc/ hhhhhhhh, Reason=rrrr/hhhhhhhh	
DK100300S	Call from ???????? Offset ????	
DK100300S	R0-R7 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx ...	
DK100300S	R8-R15 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx ...	
DK100301E	LOGIC ERROR DUMP TAKEN. DUMPCODE IS LGIC. DUMPID IS 12345678	
DK100301S	LOGIC ERROR; DUMP TAKEN TO TBDUMP	
DK100302S	System ENQ limit exceeded. Task abended with tableBASE	
DK100310W	TABLEBASE TSR nn% ALLOCATED	The series of space management algorithms used in prior levels of V6 was changed in level8. It now only issues a message when the TSR usage rises above a threshold and does not indicate what algorithm is being used. Setting the TSR_WARNING_FREQ = 0 will cause all the messages to be displayed once the TSR space usage threshold is crossed from below the threshold to above it or exactly equal to it. The default FREQ setting is 1 (i.e. only display one message per second, at the most). Level 8 will issue this message anytime the TSR usage rises above the value in TSR_WARNING_PCT. (The Default TSR_WARNING_PCT is 85.) Level 8 warnings start when the usage is exactly the TSR_WARNING_PCT value or higher. For more details refer to the TBOPT parameter settings in the *tableBASE Installation Guide*
DK100330I	DDNAME xxxxxxxx IS NOT A TABLEBASE Version 6 LIBRARY	Information only.
DK100331W	LIBRARY FREE SPACE MAP INCONSISTENT WITH LEFT BLOCK COUNT	
DK100340W	Old form ML mixed with use of extended ML	
DK100341W	CK, CO, CP COMMANDS NOT SUPPORTED YET	
DK100400E	tableBASE User Error Abend G200, Reason=0999/03E7, tableBASE Initializing	tableBASE license has expired. Call DKL (613)523-5500 to renew..
DK100411E	TSR Create failed: Failed by site IEFUSI	
DK100412E	TSR Create failed: MVS resource shortage	
DK100413W	SNAP SUPPRESSED; GETMEM SHORTAGE	
DK100470E	BDAM ERROR: FUNCTION=xxxxxxxx,FILE=ffffffff	
DK100471E	VSAM ERROR: FUNCTION=xxxxxxxx,REG15=rrrrr,ERROR CODE=xxxxx,FILE=ffffffff	
DK100472E	I/O ERROR: followed by text of I/O error	
DK100475E	xxAM ERROR: FUNCTION=xxxxxxxx,FILE=ffffffff	
DK100476E	VSAM ERROR: FUNCTION=xxxxxxxx,REG15=rrrrr,ERROR CODE=xxxxx,FILE=ffffffff	

Msg Code	Text	Meaning / Instructions
DK100477E	I/O ERROR: followed by text of I/O error	
DK100490E	TBL BASE ERROR DEBUG WTD, REASON=xxxx, DEBUG DISABLED	
DK100491I	ROTB EOTX	Information only.
DK100492I	ROTBEOT: ABND=123/123	Information only.
DK100493I	ROTB NORMAL END	Information only.
DK100494E	PERSISTENT TASK ENDED BEFORE END OF STEP	
DK100495I	IN ROTBVTSD	Information only.
DK100500E	tableBASE ERROR ABEND xxxx, Command=xx,xxxxxxxx, Reason=dddd/hhhh	
DK100501S	EXEC CICS FAILURE: xxxxxxxx Resp=cccc/ hhhhhhhh, Resp2=cccc/hhhhhhhh	
DK100511E	TSR Create failed: Failed by site IEFUSI	
DK100512E	TSR Create failed: MVS resource shortage	
DK100513W	SNAP suppressed; GETMEM shortage	
DK100530I	No storage for QTCWA; QCT function not used.	Information only.
DK100531I	Module DK1TRSTA not found; QCT function not used.	Information only.
DK100550I	tableBASE 6.0 - Initializing	Information only.
DK100551I	tableBASE 6.0 - Initialized	Information only.
DK100552E	tableBASE 6.0 - Initialization Error	
DK100553E	tableBASE 6.0 - Initialization Failed	
DK100554I	tableBASE 6.0 - Starting tableBASE	Information only.
DK100555I	tableBASE 6.0 - tableBASE started	Information only.
DK100556I	tableBASE 6.0 - Deactivating	Information only.
DK100557I	tableBASE 6.0 - Deactivated	Information only.
DK100558E	tableBASE 6.0 - Deactivation Error	
DK100559I	tableBASE 6.0 - Deactivation Failed	
DK100560I	tableBASE 6.0 - Restarting	Information only.
DK100561I	tableBASE 6.0 - Verifying	Information only.
DK100562I	tableBASE 6.0 - Verification completed	Information only.
DK100563I	tableBASE 6.0 - Error Info: Func(function12345678/subfuncn)	
DK100564I	DK1TCIN used in PLT takes one of INIT (default) or TBINIT	Information only.
DK100565I	Successful xxxx INIT must be done before xxxx TBCALL	Information only.
DK100566I	Verification failure	
DK100570I	xxxx takes one of INIT, TERM, RESET, TBINIT, TBTERM,TBRESET, TBCALL, VER, HELP	Information only.
DK100571I	Phase in of xxxxxxxx not allowed by CICS; current in-storage copy used.	Information only.
DK100572I	12345678 LP=xxxxxxxx EP=xxxxxxxx AM=xx RC=xxxxxxxx UC=xxxxxxxx	Information only.
DK100573I	Module 12345678 not loaded	Information only.

Msg Code	Text	Meaning / Instructions
DK100574I	Module 12345678 not found.	Information only.
DK100575I	EX=xxxxxxxx EN=xxxxxxxx STST=xxxx SHST=xxxx TAST=xxxx TALN=xxxxx GALN=xxxxx	Information only.
DK100576I	xxxxxxxxxxxxxxxx/xxxxxxxx OK	Information only.
DK100577I	xxxxxxxx LP=xxxxxxxx EP=xxxxxxxx LEN=xxxxxxxx AM=xxx KEY=x	Information only.
DK100590E	TBL BASE ERROR DEBUG WTD, REASON=xxxx, DEBUG DISABLED	
DK100595I	Trans error dump taken. Dumpcode is DKL1. Dumpid is xxxxxxxx	Information only.
DK100596E	CICS Assign failed (Resp/Resp2=nnnnn/nnnnn); dump suppressed	
DK100597E	CICS DUMP TRANSACTION failed (Resp/ Resp2=nnnnn/nnnnn). Dump for userid abcd1234 not taken.	
DK100600I	tableBASE VTS xxxxxx initialized	Information only.
DK100601E	GCA cannot be located; VTS stopping	The PC Server is not running. It must be running before any VTS-TSRs can be brought up (for systems NOT running the optional tableBASE Process Manager).
DK100602E	VTS Agent incompatible with GCA structure	
DK100603E	Duplicate VTS attempting to start; auto-stopping	
DK100604E	Program x2345678 is not APF authorized	
DK100605E	Unable to locate tableBASE GCA	The TPVM (VTS Manager) COMPAT is not running. It must be running before any VTS-TSRs can be brought up (for systems running the optional tableBASE Process Manager).
DK100610E	Error in xxxxxxxx macro; RC/REASON=xxx/yyyy	
DK100630I	Console interface is available	
DK100631E	Invalid command; command ignored	
DK100632W	'REFRESH not supported in tableBASE V6 VTS	
DK100633I	SHUTDOWN	Information only.
DK100634I	STOP command received	Information only.
DK100690A	VTS Agent waits, non-swappable; reply D to terminate	
DK100691A	Specify character for PC server Named Token	
DK100800I	tableBASE PC server V6 available	Information only.
DK100802E	PC server xxxxxxxx terminating; see message DK100803I for reason.	

Msg Code	Text	Meaning / Instructions
DK100803I	ERROR IN PARM FIELD ERROR IN GCA STRUCTURE ANOTHER tableBASE PC SERVER IS RUNNING LIBRARY xxxxxxxx IS NOT APF xxxxxxxx MODULE xxxxxxxx IS NOT APF BOTH STEPLIB AND SYSLIB ARE MISSING MODULE xxxxxxxx NOT FOUND IN ANY OF LPA, STEPLIB, OR SYSLIB LIBRARY xxxxxxxx MUST NOT BE CONCATENATED BLDL FOUND MODULE xxxxxxxx IN OTHER THAN STEPLIB/ SYSLIB OPEN xxxxxxxx FAILED CLOSE xxxxxxxx FAILED ERROR IN MACRO xxxxxxxx ; RC/REASON=xxx/xxxx PC SERVER NOT COMPATIBLE WITH GCA STRUCTURE	Information only.
DK100810A	PC server: Reply Q to terminate PC server	
DK100820A	PC server: Specify character for test Named Token	
DK100821I	Invalid command; ignored: xxxxxxxx	Information only.
DK100822I	STOP command received	Information only.
DK100830I	PC server: Utility functions -	Information only.
DK100830I	Purge GCA, Snap GCA, snap GCA Header, or Exit options.	Information only.
DK100831A	PC server: Specify p, s, h, or e.	
DK100832I	PC server: Exit options - U for Utilities, R to Resume PC server, or E to Exit PC server.	Information only.
DK100833A	PC server: Specify u, r, or e.	
DK100834I	PC server not compatible with GCA.	
DK100834I	Do you want to do a partial purge, i.e., NT, GCA, not GCAXs, PC code?	
DK100835A	PC server: Reply y for partial purge, n to bypass	
DK100840I	GCA does not exist.	
DK100841I	Negative GCA extension link.	
DK100842I	PC server not compatible with GCA; unformatted GCA dumped.	
DK100843I	GCA length value not valid; GCA header dumped.	
DK100844W	Customer Anchor Table DKL slot invalid.	
DK100845I	Allocate of DDname for snap dump failed.	
DK100846I	Open of DDname for snap dump failed.	
DK100847I	DDname specified for snap dump is in use.	Information only.
DK100848I	DDname specified for snap dump is invalid.	
DK100850I	DELETING NT	Information only.
DK100851I	NT DELETED	Information only.
DK100852I	FREEING PC CODE	Information only.

Msg Code	Text	Meaning / Instructions
DK100853I	PC CODE FREED	Information only.
DK100854I	FREEING GCA	Information only.
DK100855I	FREEING GCAX	Information only.
DK100856I	GCA PURGED	Information only.
DK100891I	RESXIT	Information only.
DK100990S	Incompatible ROOT and NUCLEUS versions	
DK100991E	Parms module not loaded	
DK100993E	Parms module corrupted	

TBEXEC and library conversion messages

Error and audit messages generated by the batch utility program TBEXEC, and other programs are identified and described in alphabetical order in the following table.

Messages issued from tableBASE in conjunction with the TBEXEC and other programs/ utilities vary from function to function, and are listed in the tables below. The error code is shown at the end of the message, while the originating program can be identified by the message prefix. The following prefixes may be encountered:

IOSRT-51— indicates TBEXEC in conjunction with subroutine DK1T0051
T1182— indicates utility DK1TCNV in conjunction with subroutine DK1T0051

Table A-4: Messages and error codes for TBEXEC: DK1T0051

Msg Code	Meaning / Instructions
IOSRT-51 <message> ERROR 22	DDNAME provided in command is not a valid DDNAME or it is not allocated in the jobstream.
IOSRT-51 <message> ERROR 27	The dataset for the DDname specified must be a DISK file; it is not.
IOSRT-51 <message> ERROR 30	The DSORG (dataset organization) for the DDNAME specified must be compatible for the usage: VSAM RRDS, BDAM or QSAM for a library, QSAM or a member of a PDS(E), SYSIN or SYSOUT for a sequential dataset.
IOSRT-51 <message> ERROR 31	The RECFM (Record Format) for the DDNAME specified must be F for a library, F or FB for sequential input. FA and FBA are allowed for sequential output of Report files.
IOSRT-51 <message> ERROR 32	The BLKSIZE (Block Size) parameter for the DDNAME specified must be a multiple of record length.
IOSRT-51 <message> ERROR 33	The output dataset for this DDname may not be allocated DISP=SHR.
IOSRT-51 <message> ERROR 34	An error was encountered processing a VSAM dataset. See the JESMSGLG for a description of the VSAM error.

Note: The message handling is used by many processes. Depending upon how the message is generated, it may contain only the IOSRT error message number and the associated message text (shown as <message> above), or just the message number.

Table A-5: Messages and error codes for library conversion

Msg Code	Meaning / Instructions
T1182: <message> ERROR Invalid DDname	DDNAME provided in command is not a valid DDNAME or it is not allocated in the jobstream.
T1182: <message> ERROR Not a disk file	The dataset for the DDname specified must be a DISK file; it is not.
T1182: <message> ERROR Non zero parameters	The DSORG (dataset organization) for the DDNAME specified must be compatible for the usage: VSAM RRDS, BDAM or QSAM for a library, QSAM or a member of a PDS(E), SYSIN or SYSOUT for a sequential dataset.
T1182: <message> ERROR RECFM	The RECFM (Record Format) for the DDNAME specified must be F for a library, F or FB for sequential input. FA and FBA are allowed for sequential output of Report files.
T1182: <message> ERROR BKLSIZE	The BLKSIZE (Block Size) parameter for the DDNAME specified must be a multiple of record length.
T1182: <message> ERROR SHR WRITE	The output dataset for this DDname may not be allocated DISP=SHR.
T1182: <message> ERROR VSAM	An error was encountered processing a VSAM dataset. See the JESMSGLG for a description of the VSAM error.
T1182: <message> INVALID BLKSIZE	The specified blocksize for the target library was invalid. tableBASE library blocksizes must be 3120.

Note: Certain structural errors in page tables on Version 5 tableBASE libraries may cause the Library conversion process to fail with abend S209. This same error can occur when running DK1TLCHK.

Appendix B

TBOPT dataset coding

The TBOPT dataset can be a sequential file, a member of a data set, or, for CICS, a VSAM dataset. The TBOPT dataset can be specified for all interfaces, including VTS. The dataset must contain fixed-length 80-byte records.

Note: The TBOPTV functionality has been integrated into TBOPT, allowing for a single source of run-time parameter input. TBOPTV is still maintained for backwards compatibility. If both TBOPT and TBOPTV are used, TBOPT is read first.

The data in TBOPT uses the same parameter names and values as are coded on the TBOPTGEN macro for the defaults, with the exception of LIB-LIST. TBOPT uses LIBNN to specify tableBASE libraries to update the tableBASE Library List.

Each parameter is entered on a single line in the dataset. The parameter may begin in any column. A line beginning with an asterisk (*) denotes a comment. Comments may also be added after the parameter value. A semicolon may be used to indicate line end. Comments may follow the semi-colon.

Although each region may have defined its own TBOPT dataset, all regions can share a sequential DASD dataset, and CICS regions can share a VSAM TBOPT dataset.

A sample TBOPT dataset for a batch region follows:

```
//TBOPT    DD *
* A leading asterisk denotes a comment
  ListOptions=Y
  TSRegion = 12M
  MAXNMTAB=500
  LIB01 = TESTLIB
  LIB02 = MAINLIB
/*
```

Note: ListOptions=Y is handy for diagnostic purposes; TESTLIB is first for batch testing.

Note: With the exception of the LISTOPTIONS parameter, parameters must appear only once in the TBOPT file. The form KEYWORD=* indicates that the site default is to be used. The form KEYWORD=0 indicates that the default value of a parameter that takes a character string be nullified.

