

**tableBASE**

**Release Notes**

**Release 6.0.3**

Copyright © 2009 Data Kinetics Ltd.

Document Number: TBM0011-R6.0.3v1.0

The guide is copyrighted and all rights are reserved. This document may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without the prior written consent of Data Kinetics Ltd.

Information in this guide is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this guide is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement.

tableBASE and tablesONLINE are registered trademarks of Data Kinetics Ltd. The names of other products or companies may be trademarks or registered trademarks of their respective companies.

#### 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 1.2—April 2006

Service Pack 6 Release 6.0.2 Version 1.2—August 2006

tableBASE Release 6.0.3—Version 1.0b—March 2009

Data Kinetics Technical Support at 1-613-523-5588

Data Kinetics 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](mailto:tableBASE@dkl.com)

<http://www.dkl.com>

# Table of Contents

<b>Preface</b>	<b>9</b>
Conventions used in this guide .....	9
Glossary .....	10
<b>1 - Introduction</b>	<b>11</b>
Release 6.0.3 .....	11
Moving to Release 6.0.3 from Release 6.0.2 or earlier Version 6 releases .....	11
<b>2 - Release 6.0.3 (6.0.2, Level 8)</b>	<b>13</b>
Operation, Control and Usage Change .....	13
New tableBASE Process Manager optional component.....	13
New tableBASE Process Manager compatibility .....	15
tablesONLINE, tableBASE libraries .....	15
tablesONLINE/ISPF exit programming .....	16
Command-area COUNT field.....	17
GD command.....	17
Message DK100210E .....	17
ML,VTS:, VTSFIRST or VTSLAST .....	17
F command .....	17
DD command.....	17
DV command.....	17
PC Server .....	18
tableBASE option updates .....	18
Enhancements .....	19
tableBASE run-time options: new parameters and changes.....	19
ML command.....	21
LV command enhancements.....	21
LT command enhancements .....	21
TBEXEC enhancements .....	24
VE command .....	24
>S command .....	24
tablesONLINE/CICS Application Control Table change.....	24

Batch and CICS LT command displays.....	24
Batch Interface: RW Local TSR with LDS .....	25
New VTS management features .....	25
Error diagnosis .....	25
Abnormal terminations .....	25
tableBASE internal errors .....	25
Notes .....	26
Known issues in Release 6.0.3.....	27
Enhancements and Fixes Affecting all Interfaces.....	27
Batch Interface Enhancements and Fixes .....	27
CICS interface Enhancements and Fixes.....	27
tablesONLINE/CICS Fixes.....	27
tablesONLINE/ISPF Fixes .....	28
VTS Fixes .....	28
Installation information.....	28
Changed Datasets.....	28
Batch, CICS, IMS, DB2 SPAS, VTS: .....	28
tablesONLINE/CICS: .....	28
tablesONLINE/ISPF: .....	28
Future releases .....	29

### 3 - Level 7

**31**

Operation, Control and Usage Change .....	31
PC Server .....	31
CICS module DK1TCIN .....	31
tableBASE under CICS .....	31
CICS Threadsafe.....	32
TBLBASE.....	32
TBDRIVER CL,* .....	32
EXITISPF correction .....	32
Enhancements .....	32
CICS Threadsafe.....	32
TBST transaction .....	32
LD command .....	33
sample CICS DFHSRTTB table .....	33
TBPRINT.....	33
Remove a TBOL/CICS User-developed application.....	33
Known Issues .....	34
Empty directory block removal .....	34
Invalid table name notification .....	34
Documentation.....	34
Enhancements and Fixes by Interface.....	35
Enhancements and Fixes affecting all Interfaces.....	35

Batch Interface Enhancements and Fixes .....	35
CICS interface Enhancements and Fixes.....	35
tablesONLINE/CICS Fixes.....	36
tablesONLINE/ISPF Fixes .....	36
Installation information.....	36
Changed datasets.....	36
General Notes .....	36
Batch, CICS, IMS, DB2 SPAS, VTS .....	37
CICS.....	37
TBOL/CICS .....	37
TBOL/ISPF.....	37
<b>4 - Level 6</b>	<b>39</b>
Operation, Control and Usage Change .....	39
Command consistency .....	39
Jobstream control statement consistency .....	40
TBDRIVER .....	40
TBDRIVER CLIST .....	40
Table designation terminology standardization .....	40
Non-standard calls to tableBASE .....	40
Enhancements .....	41
LV command .....	41
User exits .....	41
Library directory caching.....	41
TBEXEC new parameter .....	41
New tableBASE option parameters .....	41
Known Issues.....	42
Thread-safe CICS transactions .....	42
Empty directory blocks.....	42
tableBASE abend.....	42
Enhancements and Fixes by Interface .....	43
Enhancements and Fixes Affecting all Interfaces.....	43
Batch Interface Enhancements and Fixes .....	45
tablesONLINE/CICS Fixes.....	46
tablesONLINE/ISPF Fixes .....	46
Installation information.....	46
General Notes .....	47
Batch, CICS, IMS, DB2 SPAS, VTS: .....	47
CICS.....	47
TBOL/CICS .....	48
TBOL/ISPF.....	48

**5 - Level 5****49**

Operation, Control and Usage Change .....	49
Implicit open .....	49
OA command .....	49
Enhancements .....	49
TBEXEC .....	49
DK1TCNV .....	49
Known Issues .....	50
Enhancements and Fixes by Interface.....	50
Enhancements and Fixes Affecting all Interfaces.....	50
Batch Interface Enhancements and Fixes .....	50
CICS Interface Fixes.....	51
tablesONLINE/CICS Fixes.....	51
VTS Fixes .....	51
Installation information.....	51
General Notes .....	52
Batch, CICS, IMS, DB2 SPAS, VTS: .....	52
CICS.....	52
TBOL/CICS .....	52

## **6 - Level 4 55**

Operation, Control and Usage Change .....	55
TSR usage .....	55
TBTSLIB tableBASE library.....	55
Paged tables .....	55
PC Server .....	55
tableBASE system exits.....	56
MULTOPNX .....	56
ISPF applications and DB/2 Stored Procedures .....	56
XX command.....	56
R/W VTS-TSRs .....	56
VTS-TSR initial load and refresh processes no longer supported.....	56
Enhancements .....	57
Known Issues .....	58

## **7 - Release 6.0.2 (Service Pack 6) 59**

Introduction.....	59
Moving to 6.0.2 Service Pack 6 from 6.0.2 or from 6.0.1 .....	59
New in tableBASE 6.0.2 Service Pack 6.....	59
New in tableBASE 6.0.2.....	60
Highlights of tableBASE Version 6, all release levels .....	60
Additional enhancements in Version 6, all release levels .....	61
Exceptions to compatibility with previous versions .....	63

Changes of note .....	64
Maintenance methodology.....	64
Base product changes.....	65
Naming protocol .....	65
Multitasking .....	65
Multiuser (LOCK-LATCH).....	66
Above-the-line operation .....	66
Unified tableBASE stub.....	66
Memory model modifications.....	67
Dataspaces .....	68
Paged tables .....	68
Indexes .....	68
Controlling the number of tables in a TSR.....	68
Error subcodes .....	69
User exits .....	69
TBCALLC .....	69
Alternate Indexes .....	70
Generation number .....	70
Changes to tableBASE API commands.....	70
Libraries .....	74
VTS-TSRs and DISP=OLD for tableBASE Libraries.....	74
z/OS 1.5 with Enhanced Data Integrity .....	74
Specify library as read-only .....	74
Library expansion .....	75
Library directory caching.....	75
User-selectable blocksize for tableBASE libraries.....	75
Description of tableBASE library versions .....	76
Converting a tableBASE library .....	78
CICS.....	79
tableBASE restart .....	79
TBASEV/TBCALLV and CICS storage protection.....	79
CICS userkey/CICSkey .....	79
Waiting CICS tasks.....	80
VTS.LOAD.....	80
tableBASE VTS.....	81
Read/write VTS-TSR.....	81
Linked tables.....	82
tablesONLINE .....	82
tablesONLINE/CICS .....	82
tablesONLINE/ISPF .....	84
Notes and cautions .....	84
Notes .....	84
Cautions .....	88
Documentation updates.....	91
Access to VTS-TSRs .....	91

Determining TSR size.....	91
The CD command.....	92
TBEXEC: the UPDATE command description has changed.....	92



# Preface

This Release Note provides a high level view of the latest tableBASE offering which includes tableBASE Release 6.0.3. It also includes information on the optional products, tableBASE VTS, tableBASE Process Manager, tablesONLINE/CICS and tablesONLINE/ISPF.

Note that this Release Note provides information that applies specifically to the Beta program of tableBASE Release 6.0.3. For the beta program, tableBASE Release 6.0.3 can only be packaged to run with tableBASE Process Manager and tableBASE VTS.

## Conventions used in this guide

This guide uses conventions to differentiate code and typed commands, and to display the names of parameters.

Convention	Description
code examples and commands	Appear in a fixed font.
<b>MAXNMTAB</b>	Names of parameters appear in upper case simply for ease of reading; actual case used is upper or lower or a mixture.
Version	Following the IBM standards, version refers to a program that has significant new code or new functionality. Version is a more general term than release. For example, Version 5 includes Release 5.1 and 5.2, and is equivalent to Release 5.x.
Release	Following the IBM standards, release refers to a specific program. For example, Release 5.0 is a term that is used to identify the first release of Version 5, and does not necessarily include other software releases published under Version 5, such as Release 5.1.

## Glossary

The terms defined in the glossary will help you better understand the content of this document.

<b>Alias name</b>	An alternate name for a VTS-TSR that can be created, assigned, and used in lieu of the name assigned at VTS-TSR definition.
<b>Catalog</b>	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.
<b>Cataloged VTS</b>	A VTS-TSR that is managed by a user-defined VTS Manager, as opposed to the <i>compat</i> VTS Manager.
<b>LDS</b>	Linear DataSet.
<b>TPM</b>	tableBASE Process Manager—the tableBASE Process Manager top tier component.
<b>TPVM</b>	tableBASE Process Manager VTS Manager—the middle tier component.
<b>VTS</b>	Virtual Table Share. A VTS-TSR is a shared Data Space that provides a public location to share tables among application regions.
<b>VTS Manager</b>	The middle tableBASE Process Manager tier component. Manages VTS-TSRs.
<b>VTS switch</b>	The action of switching an alias name from one VTS-TSR to another.

## 1

# Introduction

The purpose of this document is primarily to describe the features and functionality provided in the latest maintenance release of tableBASE, Release 6.0.3. This information is detailed in [Chapter 2 “Release 6.0.3 \(6.0.2, Level 8\)”](#) on page 13.

A secondary purpose, is to consolidate the information provided in all of the maintenance release notes from Release 6.0.2, Service Pack 6, until Release 6.0.3. This information is captured in the succeeding chapters.

Note that the information provided for older releases is superceded by the information provided for more recent releases. Information provided for Release 6.0.3 supercedes information provided for all other releases.

## Release 6.0.3

Release 6.0.3 offers significant enhancements and improvements, for optional components that may require additional consideration during implementation.

tableBASE Release 6.0.3 operates with z/OS and OS/390, and is compatible with CICS TS, IMS TM, DB2 and batch.

## Moving to Release 6.0.3 from Release 6.0.2 or earlier Version 6 releases

- Release 6.0.3 is a complete replacement for all executable modules from Release 6.0.2 or earlier releases.
- Release 6.0.3 tableBASE libraries are compatible with all Version 6 libraries.

**Note:** Release 6.0.3 libraries are not compatible with Version 4 and 5 (v4 and v5) libraries. We do not recommend updating from v5 or earlier directly to Release 6.0.3. If this is your situation, please contact Data Kinetics.



# 2

## Release 6.0.3 (6.0.2, Level 8)

This section covers the most recent release, Release 6.0.3 (6.0.2, Level 8)— the changes to tableBASE Version 6 between maintenance level 7 and maintenance level 8. The major components of this release are:

- a new optional component: tableBASE Process Manager
- changes to tablesONLINE (including tablesONLINE/ISPF exit programming)
- changes to tableBASE libraries
- TBOPT: new parameters and changes
- ML and LT command enhancements
- Also included, are fixes of known issues from Release 6.0.2, Level 7.

### Operation, Control and Usage Change

The following base product changes are applied to tableBASE Release 6.0.3.

#### **New tableBASE Process Manager optional component**

tableBASE Process Manager is a management product that can improve the performance of mainframe systems employing tableBASE VTS TSRs, and enhance the manageability and data security of those systems.

#### **tableBASE Process Manager inter-operability**

tableBASE Process Manager is compatible with tableBASE Release 6.0.3. It is not compatible with versions of tableBASE prior to Release 6.0.3.

To run tableBASE Process Manager, you must install tableBASE Release 6.0.3, and the tableBASE VTS component.

## **tableBASE Process Manager features**

The following is a brief description of the features provided with the tableBASE Process Manager application:

### ***Improved VTS initialization***

tableBASE Process Manager, being fully compatible with standard MVS scheduling systems, introduces a new level of flexibility to the VTS startup and shutdown processes. VTS-TSRs can be scheduled for startup and shutdown in groups, and at different times. This eliminates the need to startup and shutdown VTS-TSRs individually.

### ***Pre-loaded VTS-TSRs***

Using tableBASE Process Manager, VTS-TSR tables can be mapped to Linear Data Sets (LDSs). On VTS-TSR initialization, the link to the LDS is set up, and data is available immediately to calling applications. The use of LDSs in this way improves performance by eliminating the time previously required at VTS-TSR startup to load table data row-by-row from the DASD-based library, and to rebuild all indexes.

### ***Read-only VTS-TSRs***

It is now possible to create read-only VTS-TSRs. Read/write VTS-TSRs, useful when updating tables is a requirement, are less desirable when that data is intended for table-driven decision processing. Access time can be burdened by needless queuing and table locking—read-only VTS-TSRs eliminate the need for this type of overhead processing. Using read-only VTS access, such costly processing can be completely eliminated.

### ***Update tables in the background***

Using tableBASE Process Manager, VTS-TSRs can be updated in the background without affecting the performance of calling applications, and without the delays associated with VTS startup. In effect, you can synchronize live updates to your active tables.

### ***Switching VTS-TSRs***

Using tableBASE Process Manager, VTS-TSRs can be switched in real time. Initiated by a single command, the switch effectively switches old data (in the original VTS-TSR) for updated data (in a new VTS-TSR). Calling applications (batch programs, transactions, etc.) still accessing the original VTS-TSR will be allowed to complete their tasks. New tasks will access the updated data in the new VTS-TSR.

### ***Alias names for VTS-TSRs***

Calling applications can access VTS-TSRs using an alias name; the actual name of a VTS-TSR is unknown to the calling application. This feature supports the switching and background updating features.

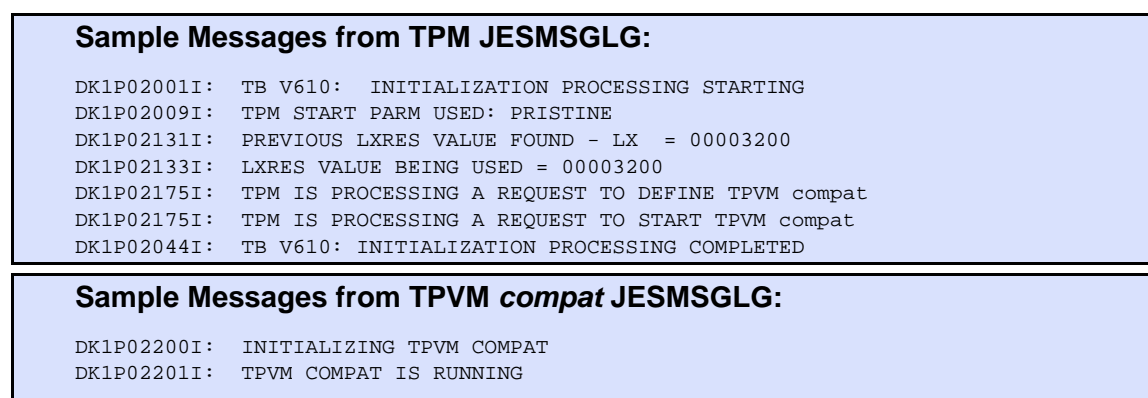
## New tableBASE Process Manager compatibility

tableBASE Release 6.0.3 is compatible with the new tableBASE Process Manager product. For further information on tableBASE Process Manager, see the tableBASE Process Manager User Guide.

### TPM, TPVM *compat* always running

tableBASE Process Manager (TPM) and the VTS Manager (TPVM) *compat* (the tableBASE PC Server equivalent) must be up before any tableBASE interfaces will work. For further information, see the tableBASE Process Manager User Guide.

The following messages indicate TPM and *compat* are initialized and ready to use:



**Figure 2-1: Sample JESMSGLG messages**

If the tableBASE PC Server equivalent, i.e., TPVM *compat* not is running, batch, DB2 & IMS interfaces will fail with message:

```
DK100201I tableBASE PC Server unavailable
```

CICS startup will fail with message:

```
DK100567E The tableBASE PC Server must be running for tableBASE to initialize
```

DK1VAGNT startup will fail with message:

```
DK100605E V610 Environment not ready; VTS stopping
```

**Note:** Prior to Release 6.0.2, the PC server was only required for VTS access and for multitasking.

## tablesONLINE, tableBASE libraries

tablesONLINE/ISPF, tablesONLINE/CICS, and all tableBASE libraries are unchanged from tableBASE Release 6.0.2.

**Note:** Release 6.0.3 is based on Release 6.0.2 tableBASE; some optional products, utilities and tableBASE libraries are unchanged. For this reason, you may see V601 or V602 version indications in various reports and messages. This should not be a cause for concern.

## tablesONLINE/ISPF exit programming

The tablesONLINE/ISPF interface has changed; if you're using tablesONLINE/ISPF exits, you'll have to change them. The following shows the data areas that are passed between tablesONLINE and the user exit program. This information is also in the copybook, EXITISPF, in the distribution source library.

```

*-----*
* THIS IS A COBOL COPY BOOK FOR USE IN TABLESONLINE/ISPF EXIT
* PROGRAMMING. IT MAPS THE LINKAGE SECTION FOR THESE EXITS
*-----*
01  HOOK-PARAMETER-AREA-1.
    05  HOOK-FIELD                                PIC  X(50).
    05  HOOK-FLD-DEFS                             PIC  X(100).
    05  TABLE-COMMAND-AREA.
        10  TB-COMMAND                            PIC  X(02).
        10  TB-TABLE                              PIC  X(08).
        10  TB-FOUND                              PIC  X.
        10  TB-INDIRECT                           PIC  X.
        10  FILLER                                PIC  X.
        10  TB-ABEND-OVERRIDE                     PIC  X.
        10  TB-ERROR                              PIC  S9(04) COMP.
        10  TB-COUNT                              PIC  S9(08) COMP.
        10  TB-LOCKLATCH                          PIC  X(08).
        10  TB-ROW-OVERRIDE-LENGTH                PIC  S9(08) COMP.
        10  TB-ROW-ACTUAL-LENGTH                  PIC  S9(08) COMP.
        10  TB-FG-KEY-LENGTH                      PIC  S9(04) COMP.
        10  TB-FUNCTION-ID                        PIC  S9(04) COMP.
        10  TB-FUNCTION-AREA                      PIC  X(08).
        10  TB-DATE-AREA  REDEFINES  TB-FUNCTION-AREA.
            15  TB-DATE                            PIC  X(08).
        10  FILLER                                PIC  X(20).
        05  TB-RETURNED-ABS-GEN-NO                PIC  S9(04) COMP.
        05  TB-ERROR-SUBCODE                      PIC  S9(04) COMP.
    05  I-ZUSER                                    PIC  X(08).
    05  TABLE-OPEN-SW                            PIC  X(01).
    05  ACTION-BYPASS-IND                         PIC  X(01).
    05  FIELD-COMMAND-AREA.
        10  FT-COMMAND                            PIC  X(02).
        10  FT-TABLE.
            15  FILLER                            PIC  X(07).
            15  FT-TABLE-SUFFIX                   PIC  X.
        10  FT-FOUND                              PIC  X.
        10  FT-INDIRECT                           PIC  X.
        10  FILLER                                PIC  X.
        10  FT-ABEND-OVERRIDE                     PIC  X.
        10  FT-ERROR                              PIC  S9(04) COMP.
        10  FT-COUNT                              PIC  S9(08) COMP.
        10  FT-LOCKLATCH                          PIC  X(08).
        10  FT-ROW-OVERRIDE-LENGTH                PIC  S9(08) COMP.
        10  FT-ROW-ACTUAL-LENGTH                  PIC  S9(08) COMP.
        10  FT-FG-KEY-LENGTH                      PIC  S9(04) COMP.
        10  FT-FUNCTION-ID                        PIC  S9(04) COMP.
        10  FT-FUNCTION-AREA                      PIC  X(08).
        10  FT-DATE-AREA  REDEFINES  FT-FUNCTION-AREA.
            15  FT-DATE                            PIC  X(08).
        10  FILLER                                PIC  X(20).
        05  FT-RETURNED-ABS-GEN-NO                PIC  S9(04) COMP.
        05  FT-ERROR-SUBCODE                      PIC  S9(04) COMP.
    05  HOOK-FILLER                                PIC  X(171).
    05  I-TYPECHG                                  PIC  X(01).
    05  HOOK-POINTER                              PIC  S9(04) COMP SYNC.
    05  DATA-ARRAY-NEW.
        10  DATA-ENTRY-NEW                       OCCURS 1000
            PIC  X(50).
01  HOOK-PARAMETER-AREA-2.
    05  WS-EDIT-ERR                               PIC  S9(04) COMP SYNC.
    05  HOOK-USER-MESSAGE                         PIC  X(60).

```



## Command-area COUNT field

In tableBASE Release 6.0.3, tableBASE will zero the command-area COUNT field when it returns 17 in the command-area TB-ERROR field. tableBASE V510 left the count untouched in this case.

## GD command

The TBDRIVER/DK1TDRV GD command display of the DDNAME field is increased from eight bytes to 12 bytes. This allows display of VTS:xxxxxxx for tables accessed by a linked VTS-TSR.

**Note:** This may affect applications which post-process the output from TBDRIVER steps.

The tableBASE GD command now shows the time and data of creation for a temporary table (created by DT or ALT). Previous releases showed these values only for tables that were stored on a library.

## Message DK100210E

The existing message:

DK100210E Unable to locate tableBASE GCA

has its message number changed to DK100605E in tableBASE Release 6.0.3.

## ML,VTS:, VTSFIRST or VTSLAST

Use of ML,VTS:, VTSFIRST or VTSLAST to access tables is unsupported when using a Local TSR with an LDS.

## F command

As of Release 6.0.3, the F command can no longer be used to stop VTS-TSRs.

## DD command

If applications use the obsolete DD command is used to return a table definition, there will be differences between Version 5 operation and Version 6 operation. If a table has a read password, Version 5 DD returns code 30 (invalid password), and nothing in the definition block, while Version 6 DD returns code 0 and a complete definition block. Also, in Version 6, DD will return blanks for a table name, and low values for a VTS name.

## DV command

In prior releases, the DV command did not check whether the target DDNAME was allocated. tableBASE put the target DDNAME in the table definition even if the target DDNAME was not allocated. Subsequent Store commands also failed with return code 40. In Version 6, tableBASE checks to see that the target DDNAME is

allocated and returns an error code 40-1 if it is not. The table is then treated like a defined (DT) table.

## **PC Server**

The PC Server in Release 6.0.3 can be introduced without the need of an IPL on the LPAR. CICS regions that are not recycled with the new PC Server is started will continue to use the prior level PC Server.

## **tableBASE option updates**

Updates to tableBASE default options must be reapplied to the new version of the souces modules, reassembled and relinked:

DK1T1134 changes to: DK1BBASE

DK1T1334 changes to: DK1IBASE

DK1T1434 changes to: DK1DBASE

DK1T2734 changes to: DK1CBASE

DK1V1134 changes to: DK1VBASE

## Enhancements

The following changes apply to tableBASE Release 6.0.3:

### tableBASE run-time options: new parameters and changes

There are new parameters and changed tableBASE run-time option parameters (handled by TBOPT and TBOPTGEN), including:

- **HASH\_HI\_DEN\_LIM** and **HASH\_LOW\_DEN\_LIM** are new parameters. limit the density of the index for a hash table opened in Version 6. These values are designed to prevent performance problems which can occur when inappropriately high values are used when defining hash tables (can result in numerous new indexes being created). Other problems occur if the difference between low and high density values is too small. A ratio of 2/3 is now enforced: Low density may not be greater than 2/3 of high density. **HASH\_HI\_DEN\_LIM=nnn** must be between 100 and 900 (10% and 90%); the default is 900. **HASH\_LOW\_DEN\_LIM=nnn** must be between 10 and 600; the default is 600.
- **LDS** is a new parameter. If **LDS=Y** is used, the Local TSR or VTS-TSR will be mapped to an LDS (Linear Data Set), when the TSR is brought down. If this option is used, a corresponding **LDSTSR DD** statement, with the LDS Dataset Name, must be included in the job.  
If **LDS=Y** is used in conjunction with **TSRACCESS=RW**, the **DISP** of the LDS should be set to **OLD** i.e., **//LDSTSR DD DISP=OLD,DSN='Your.LDS.DSName'**  
If **LDS=Y** is used in conjunction with **TSRACCESS=RO**, then the **DISP** of the LDS can be set to **SHR** i.e., **//LDSTSR DD DISP=SHR,DSN='Your.LDS.DSName'**  
If **LDS=N** is used, no mapping of TSRs will occur during TSR shutdown.
- **LISTOPTIONS=X** is a special setting for TBOPTGEN (DK1T1134). It is the equivalent of **LISTOPTIONS=N** if the TBOPT DD was not present in the jobstream, and the equivalent of **LISTOPTIONS=Y** if the TBOPT DD is present. **ListOptions=X** applies to TBOPTGEN only; it does not apply to TBOPT.
- **LOCKTIMERC** and **LOCKTIMEWTO** are new parameters.  
**LOCKTIMERC=nnnnnn** specifies the number of seconds (default 0) that tableBASE should wait for a lock. When the **LOCKTIMERC** interval has passed, **RC=71** is returned. A value of **LOCKTIMERC=0** specifies that the process will never time out.  
**LOCKTIMEWTO=nnnnnn** specifies the number of seconds (default 30) to wait before issuing a message (DK100227W) that the process is waiting for a lock. A value of **LOCKTIMEWTO=0** specifies that no warning message will be issued. They are used internally by tableBASE to maintain table integrity in the TSR. It is unrelated to the table **ENQUEUE** that occurs when a table is opened for write (**OW**).
- **MTRETAIN** is a new parameter. It determines whether the allocated rows and index areas are to be retained when an **MT** command is issued or whether they are to be reduced to the original allocation before table expansion. (It may be necessary to save the original allocations). The default is "N", which is current processing—not retained.

- The maximum **STROBE** value has been increased to 2G – 1 (2,147,483,647), up from 10,000,000 in previous releases.
- **TABLEWAITWTO** and **TABLEWAITWRC** are new parameters. Use **TABLEWAITWTO** to specify a value to indicate the number of seconds that a user will wait to receive the MVS enqueue to open a VTS-TSR based table for read or write before timeout. The delivered default is 0. **TABLEWAITWRC** indicates the elapsed time before a message will be generated to report that the enqueue has not yet been received. The Wait for Enqueued Table switch must be set to Y for these parameter to have any effect.
- **TPVM** is a new parameter. It specifies the TPVM (VTS Manager) that manages the VTS-TSRs that are to be accessed during the life of a region. The TPVM name can be up to eight characters long.
- **TSR\_ALGORITHM** is a new parameter. **TSRAlgorithm=P|B|C|D** allows the process creating the TSR to specify whether the Performance, Balance, Capacity or Default operation is desired. This parameter allows for flexibility for the address space manager:
  - P (Performance) indicates that the TSR will be optimized for performance. Space will be assigned to tables within the TSR so as to minimize CPU usage, which may result in a less than optimum use of space.
  - C (Capacity) indicates that the TSR will be optimized for capacity. Space will be assigned to tables within the TSR so that so that they can contain as much data as possible, which may consume more CPU resources.
  - B (Balance) indicates that the TSR will be optimized for a balance between performance and capacity.
  - D (Default) indicates that there will be no optimization, and there will be no messages regarding optimization. However, messages will be provided regarding current TSR capacity percentage.
- **TSR\_WARNING\_PCT** and **TSR\_WARNING\_FREQ** are new parameters. Use **TSR\_WARNING\_PCT** to to indicate the percentage of TSR allocation allowed before a message will be generated to report on allocation. **TSR\_WARNING\_FREQ** specifies the frequency of these reports.
- **TSRACCESS** is a new parameter. The two values that can be used are 'RW' or 'RO'. If not specified, the default **TSRACCESS** is RW. If **TSRACCESS=RW** is used, applications accessing the Local TSR or VTS-TSR under which this option is specified will be allowed Read-Write access to the TSR. If **TSRACCESS=RO** is used, applications accessing the VTS-TSR under which this option is specified will be allowed Read-Only access to the TSR. This parameter applies only if you have licensed the optional tableBASE Process Manager.
- **VTSFIRST**, **VTSLAST**: These two parameters, which specify the VTS-TSR names to be searched prior to or after the ML search list, can now be up to eight characters long.
- The **VTSNAME** parameter, used only when the VTS Agent initiates a VTS-TSR, can now be up to eight characters long.

**Note:** All VTS names must eight characters long. A smaller name must be blank filled to eight characters.

- **VTSPREFIX** is a new parameter. You can use it to specify the VTS prefix that is to be used with the ML command. The default VTS prefix is "VTS:". The VTSPREFIX specified must end with a colon (":" - no quotes) and can be 1-4 characters long, including the colon.
- **ZEROROWS** is a new parameter. Related to the MTRETAIN parameter, and applies only when MTRETAIN = N. It determines whether the data rows area should be zeroed when a TSR is deallocated. Note that the index area is never zeroed, except for hash indexes.

## ML command

The VTS-TSR name in an ML search list can now be up to 7 characters long.

In order to include a VTS-TSR which has a name greater than 4 characters in the search list, first change VTSPREFIX in TBOPT to the desired length, then use the new prefix in the ML command.

For example, if VTSPREFIX=V:, then the ML command would look like ML V:xxxxxx where xxxxxx is the VTS-TSR name of up to 6 characters.

## LV command enhancements

The LV command displays the VTS prefix. Also, it is now supported in both TBDRIVER and CICS TBDR (DK1TDRV and DK1TDRVC).

## LT command enhancements

There are several new fields for the LIST-BLOCK parameter, and the TABLE-STATS optional parameter.

### New LIST-BLOCK parameter fields

The LT command will now return an additional 44 bytes of information in the LIST-BLOCK parameter if it is called with a FUNCTION-ID of 16 in the COMMAND-AREA. This parameter now consists of 21 fields of which the first three are input to the LT command and next 18 are output from the command. The inputs specify the portion of the open tables statistics to be placed in the companion TABLE-STATS parameter. The output fields describe usage statistics about the TSR.

If the LIST-BLOCK parameter is used in an LT command with no function-id in the command-area, only the first eleven fields will be used; i.e., after the first 3 input fields, only the next 8 fields will be returned. If the LIST-BLOCK parameter is used in an LT command with a function-id of 16 in the command-area, then all 21 fields will be used.

New fields are in **bold:**

```
01 LIST-BLOCK.
   05 LIST-FROM                PIC S9(8) COMP VALUE +1.
```

05 LIST-REQD	PIC S9(8) COMP VALUE +96.
05 LIST-SIZE	PIC S9(8) COMP VALUE +36.
05 LIST-TOTAL	PIC S9(8) COMP.
05 LIST-RETURNED	PIC S9(8) COMP.
05 LIST-TSR-HW	PIC S9(8) COMP.
05 LIST-OPEN-HW	PIC S9(8) COMP.
05 LIST-OPEN-NOW	PIC S9(8) COMP.
05 LIST-TSR-NOW	PIC S9(8) COMP.
05 LIST-TSR-SZ	PIC S9(8) COMP.
05 LIST-STROBE	PIC S9(8) COMP.
05 LIST-TOTAL-HWM	PIC S9(9) COMP.
05 LIST-TOTAL-CALL	PIC S9(18) COMP.
05 LIST-MAX-TBLS	PIC S9(9) COMP.
05 LIST-SIZE-SPCMAP	PIC S9(9) COMP.
05 LIST-SIZE-HASHINDX	PIC S9(9) COMP.
05 LIST-SIZE-INDX	PIC S9(9) COMP.
05 LIST-SIZE-DEFNS	PIC S9(9) COMP.
05 LIST-TSR-AVAIL	PIC S9(9) COMP.
05 LIST-SPCMGR-STATUS	PIC X.
05 LIST-RESERVED	PIC X(7).

For a description of the pre-existing fields, see the tableBASE Programmer's Guide.

#### New field descriptions:

- a. **LIST-TOTAL-HWM** (fullword binary)  
The High Water Mark for the number of open tables.
- b. **LIST-TOTAL-CALL** (doubleword binary)  
The total call count for all tables in the TSR.
- c. **LIST-MAX-TBLS** (fullword binary)  
The maximum number of tables in the TSR.
- d. **LIST-SIZE-SPCMAP** (fullword binary)  
Size of Spacemap + CME size in TSR
- e. **LIST-SIZE-HASHINDX** (fullword binary)  
Size of Hashed Index to the table directory in the TSR
- f. **LIST-SIZE-INDX** (fullword binary)  
Size of the Index to the table directory in the TSR
- g. **LIST-SIZE-DEFNS** (fullword binary)  
Size of the table directory in the TSR
- h. **LIST-TSR-AVAIL** (fullword binary)  
Number of blocks available in TSR
- i. **LIST-SPCMGR-STATUS** (1 byte)  
TSR Space Manager Status
- j. **LIST-RESERVED** (7 bytes)  
FILLER.

#### New TABLE-STATS parameter fields

The LT command can now also return up to 80 bytes of table statistics in the TABLE-STATS parameter. As in Release 6.0.2, this is done by specifying the length of

information to be returned in the LIST-SIZE field under the LIST-BLOCK parameter.

New fields are in **bold**:

```

01 TABLE-STATS.
   05 LIST-TABLE-ENTRY OCCURS 36 TIMES.
      10 TABLE-NAME PIC X(8).
         10 TABLE-OPEN-STATUS PIC X.
         10 TABLE-LOCAL-VTS PIC X.
         10 TABLE-ALT-INVOKED PIC X.
         10 FILLER PIC X.
         10 TABLE-CALLS-TRUNC PIC S9(9) COMP.
         10 TABLE-SIZE PIC S9(9) COMP.
         10 TABLE-ROWS PIC S9(9) COMP.
         10 TABLE-RWS-BF-EXP PIC S9(9) COMP.
         10 TABLE-DATATBL-VTSNAME PIC X(8).
         10 TABLE-UPDATE-CALLS-TRUNC PIC S9(9) COMP.
         10 TABLE-DATE-TIME PIC 9(12).
         10 FILLER PIC S9(4) COMP.
         10 TABLE-CALLS PIC S9(18) COMP.
         10 TABLE-UPDATE-CALLS PIC S9(18) COMP.
         10 TABLE-VTSNAME PIC X(8).

```

For a description of the pre-existing fields, see the tableBASE Programmer's Guide.

New field descriptions:

- a. **TABLE-CALLS-TRUNC** (fullword binary)  
Only a name change was TABLE-CALLS, now TABLE-CALLS-TRUNC. The total number of calls made against the table since it was last opened (truncated to a fullword).
- b. **TABLE-UPDATE-CALLS-TRUNC** (fullword binary)  
Number of updates to this table (truncated to a fullword).
- c. **TABLE-DATE-TIME** (12 bytes)  
The date and time that the table was last stored to a library.
- d. **FILLER** (4 bytes)  
Filler.
- e. **TABLE-CALLS** (doubleword binary)  
The total number of calls made to this table since it was last opened.
- f. **TABLE-UPDATE-CALLS** (doubleword binary)  
The total number of updates to the table since it was last opened.
- g. **TABLE-VTSNAME** (8 bytes)  
The name of the VTS-TSR if this entry describes a Linked table.

## TBEXEC enhancements

TBEXEC has a new option for the PRINT command. `DETAIL=YES` provides additional information on each table. Also, TBEXEC error messages have been improved.

## VE command

A new command, VE is added to TBDRIVER and CICS TBDR (DK1TDRV and DK1TDRVC) and tablesONLINE/ISPF. It provides version information for the current application interface.

## >S command

A new command, >S, is added to CICS TBDR (DK1TDRVC). It provides a mechanism to transfer a table name containing hex characters from the LT command output to the table name field so another command can be invoked against it.

## tablesONLINE/CICS Application Control Table change

The tablesONLINE/CICS Application Control Table has a new field:

### Enable directory list

This field controls whether F1 (Help) produces a list of library tables, or just the standard help panel display. This indicator may be set to either Y or N. If this indicator is set to N, the help panel will be displayed; if set to Y, all tables in the library will be listed.

## Batch and CICS LT command displays

The LT commands in TBDRIVER and CICS transaction TBDR return additional information for tables:

- **UPDATES** - new in Release 6.0.3  
The number of updates made to the table in the TSR.
- **DATE-TIME** - new in Release 6.0.3  
The date and time the table was last stored to the library.

- **BASE/VTS**

If the table is a linked table from a VTS-TSR, this field will now take the format of VTS:xxxxxxx, where xxxxxxxx can now be up to 8 characters for the VTS name.

**Note:** All VTS names must eight characters long. A smaller name must be blank filled to eight characters.



## Batch Interface: RW Local TSR with LDS

With Release 6.0.3, batch jobs accessing tableBASE can now be associated with an LDS (Linear Data Set). Specifying an LDS in the batch job will allow the Local TSR to be mapped onto the LDS when the TSR is brought down at the end of the job.

This feature can be used to create the data for the LDS of a Read-Only VTS-TSR.

In order to use an LDS with your batch job, the following TBOPT parameters are required:

```
TSRACCESS=RW;  
LDS=Y;
```

and the following LDSTSR DD statement is required:

```
//LDSTSR DD DISP=OLD,DSN='Your.LDS.DSName'
```

## New VTS management features

The ability to define, startup, shutdown, delete, autostart & autoshut VTSs under TPVM *compat* using TPDRIIVER is now available – for further information, see the Release 6.0.3 documentation.

## Error diagnosis

New functionality is provided for programmers in the area of abnormal terminations and tableBASE internal errors.

## Abnormal terminations

Under CICS, tableBASE will produce a DKL1 CICS transaction dump for any abnormal termination.

## tableBASE internal errors

Under CICS, tableBASE will produce a CICS transaction dump with an abend code of LGIC when an internal error is trapped.

## Notes

The following items apply to tableBASE Release 6.0.3:

1. VTS Agent and PC Server

If a VTSAGENT initializes in Release 6.0.2 when the PC server is not running, the message DK100601E (Unable to locate tableBASE GCA) is returned.

In Release 6.0.3, if TPVM *compat* (replaces the PC Server) is not running the message DK100605E (Vxxx environment not ready, VTS stopping is returned.

The reason for the difference is that Release 6.0.3 detects the problem earlier than previous releases did.

2. Version 5 conversion abend

Certain structural errors in page tables on V5 tableBASE libraries may cause with the Library conversion process to fail with abend S209. This same error can occur when running DK1TLCHK.

3. tableBASE Return Code 38 is obsolete in tableBASE Version 6. The conditions that cause the return code apply to Version 5 only. For more information, refer to the Version 5 documentation.

## Known issues in Release 6.0.3.

### Enhancements and Fixes Affecting all Interfaces

ID	Description
2509	Set defaults for DT and CD as in V5
2527	Allow STROBE to be set to higher values
2686	Prevent TSR corruption when creating table index
2742	XT errors when deleting last table in library
2785	G014 abend during date sensitive processing
2798	Improve handling invalid call to tableBASE
2802	Performance access Sequential table with Queued processing
2810	Update date sensitive processing for compatibility with V5
2808	Handle out-of-space condition in TSR when opening alternate
2809	Correct TSR space allocation for alternate index
2815	Prevent access to base table during open of alternate

### Batch Interface Enhancements and Fixes

ID	Description
2717	Fix Multitasking Batch abend processing XX commnad

### CICS interface Enhancements and Fixes

ID	Description
2782	Table locked in VTS-TSR after CICS termination
2801	Prevent abends in CICS tableBASE termination logic

### tablesONLINE/CICS Fixes

ID	Description
2688	Fix multiple user support in TBOL/CICS
2774	Support for Read Only TSRs (V610)

## tablesONLINE/ISPF Fixes

ID	Description
2661	Upgrade TBOL/ISPF and its exit interface
2788	Allow Browse of very large tables

## VTS Fixes

ID	Description
2782	Table locked in VTS-TSR after CICS termination

## Installation information

### Changed Datasets

With this level the following datasets have changed:

- **LOAD:** Modules are a complete replacement for existing modules of the same name except for DK1TNAME and DK1xBASE. Modules specified in CNTL(AUTHLIB) and CNTL(AUTHLIBV) must be recopied to an authorized library.

### Batch, CICS, IMS, DB2 SPAS, VTS:

1. Replace your existing V6 loads with the new V601-08 loads except for DK1TNAME and all DK1xBASE modules. Use your current versions of those modules. Using CNTL(AUTHLIB) and CNTL(AUTHLIBV), copy the specified modules to the authorized libraries.
2. Copy changed members of the V601-08 CNTL, CLIST, PANELS, and MESSAGES PDSes to your corresponding V6 PDSes. If you have modified any of these members, reapply your changes.

### tablesONLINE/CICS:

- Eight load modules have been updated

### tablesONLINE/ISPF:

- Copy changed members of the V601-07 PANELS and MESSAGES PDSes to your corresponding V6 PDSes. If you have modified any of these members, reapply your changes. Remove members TBPAGE and TBUNPAGE from your V6 CLIST PDSes.

## **Future releases**

The next release of tableBASE is under development. If you have enhancement requests for future tableBASE releases please let us know.

Please contact the us at:

- [tablebase@dkl.com](mailto:tablebase@dkl.com) or
- 1-800-267-0730.



# 3

## Level 7

This section covers Release 6.0.2, Level 7. The major components of this release were:

- CICS Threadsafe
- PC Server changes
- CICS module DK1TCIN changes

### Operation, Control and Usage Change

#### PC Server

The PC Server must now be running for a CICS region to initialize tableBASE. It must also be running for the correct operation of CICS Threadsafe support. If it is not running, table BASE will issue the message “DK100567I The tableBASE PC Server must be running for tableBASE to initialize” and will not initialize. The PC Server shipped with this level is required for tableBASE to initialize correctly in a CICS region. If the PC Server is stopped while CICS is running, calls to tableBASE may abend.

#### CICS module DK1TCIN

CICS module DK1TCIN must now be defined as CONCURRENCY(QUASIRENT). If it is not, tableBASE will issue message “DK100568I Program DK1TCIN must be defined as CONCURRENCY(QUASIRENT)” and it will not initialize. It must also be defined as EXECKEY(CICS).

#### tableBASE under CICS

Under CICS, tableBASE is now initialized when DK1TCIN is invoked (either through the PLT or transaction TBST). In earlier releases, it was initialized by the first application call to tableBASE.

## CICS Threadsafe

With the introduction of CICS Threadsafe support in this release, customers will be able to run Threadsafe applications with tableBASE without experiencing performance degradation due to TCB-switching. Since the definition of Threadsafe CICS commands is not static, we recommend that customers contact us before migrating to a new release of CICS beyond CICS/TS 3.1. L9 OTE TCBs and OPENAPI applications have not been tested for any CICS release.

## TBLBASE

A new version of DK1TCALL(TBLBASE) is being supplied. It allows access to tableBASE to coexist in applications that access CICS through EXCI. If the linkedit characteristics of your installed version are different from the new version, relink this load module with the same characteristics as your current version.

## TBDRIVER CL,\*

The TBDRIVER CL,\* has been updated to work correctly when invoked with the TBTEST alias. Note that it will create and delete tables TBALIST and TBBLIST even if user tables exist with these names.

## EXITISPF correction

The supplied sample TBOL/ISPF exit data-area EXITISPF has been corrected. The previous version documented extended command areas for TABLE-COMMAND-AREA and FIELD-COMMAND-AREA. TBONLINE actually passes short (28-byte) command areas. The Programming Guide, Chapter 12 (tablesONLINE/ISPF Exit Programming), will be updated to include the corrected copy of EXITISPF under "Structure of an Exit Program".

## Enhancements

### CICS Threadsafe

CICS Threadsafe applications that access tableBASE are now supported. Several changes to operation are involved - see the installation instructions in this document.

### TBST transaction

The TBST transaction is supported for stopping and restarting tableBASE within a CICS region with restrictions. Stopping tableBASE results in the TSR being released.



Restarting tableBASE results in a new, empty TSR being created. This transaction is only supported if all transactions accessing tableBASE have completed or have been purged. If any transactions are still accessing tableBASE when this transaction executes, unpredictable results, including abends and blocked transactions, can result. Recovery could entail recycling CICS and any affected VTSSs.

## LD command

The LD command has an additional DIRTYE option. Option “L” returns only the library DDNAMEs and table names, one row for each table. It is significantly more efficient than other DIRTYE options.

## sample CICS DFHSRTTB table

A sample CICS DFHSRTTB table is supplied to avoid S067 abends when a VTS-TSR is not available.

## TBPRINT

Elapsed and CPU time usage by TBPRINT (DK1TPTBL) is significantly reduced.

## Remove a TBOL/CICS User-developed application

The following procedure will remove a TBOL/CICS User-developed application:

1. Delete the following application tables from TBAPPLB.
  - xxxxMENU
  - xxxxDESC
  - xxxxMSGs
  - xxxxHELP
  - xxxxCMDS
  - xxxxPFKS
  - xxxxLIBR

### Notes:

- Delete tables where xxxx is a 4 byte table prefix, used when the application was copied through TBOL admin.
- NEVER delete tables that have a prefix of TBOL. We assume that tables with prefix TBOL were never modified when tablesONLINE/CICS was delivered, and that the tables were copied over instead.
- To delete a particular application you can simply use TBDR to delete each table by:
  - doing an ML to TBAPPLB.

- doing a CL of the specified table to close it from the TSR
  - doing an XT to eliminate the table from the library.
  - Before deleting any of the tables, please ensure that no other application ID, in TBOLACT (in TBACTLB), except the application you are trying to delete, uses any of the same table names in any of its rows.
  - If the table name is used in any other application ID, then DO NOT delete the table.
2. Delete the application rows from TBOLACT in TBACTLB.  
Delete the respective application rows from the TBOLACT table in TBACTLB.
  3. Delete the application rows from TBOLM2M in TBDICLB.  
Delete the respective application rows from the TBOLM2M table in TBDICLB.

## Known Issues

### Empty directory block removal

Some earlier versions of the tableBASE library conversion process did not remove all empty directory blocks. The Library Directory Caching Feature does not tolerate empty directory blocks. To correct this problem, run the DK1TLCHK utility DIAGNOSE function against the library. If it indicates there are empty directory blocks, run the TBEXEC utility COPY LIB function to recreate the library. This process will remove empty directory blocks.

### Invalid table name notification

In Version 5, use of an invalid table name (e.g., spaces or lowvalues) on the Close command returned errcode 2 (table not found). Version 6 returns errcode 12 (invalid table name).

### Documentation

For an IMS/TM environment, documentation will be updated clarifying the requirement that DK1TNUCL, DK1TNAME, DK1TROT, DK1IBASE and DK1TCALL (and aliases) must be preloaded. This is currently documented in the Administration Guide. The load addresses of these modules are cached within tableBASE control blocks.

## Enhancements and Fixes by Interface

### Enhancements and Fixes affecting all Interfaces

ID	Description
1880	Improve module tracking for diagnostics
1899	Improve PC Server startup
2415	Add DIRTYPE L option for LD command
2462	Prevent empty Library block after abend
2547	Add internal tracing options
2556	Fix DK1TCALL abend S0C1 when using EXCI in batch/IMS
2568	Fix abend resulting from implicit open timing error
2569	Change abend in DT command with estimated rows < 0 to error code 43
2573	Improve TBPRINT (DK1TPTBL) performance
2586	Upgrade System level diagnostic Exits
2612	Refresh library directory cache refresh after incomplete update

### Batch Interface Enhancements and Fixes

ID	Description
2041	Prevent Abend in TBTEST entry point to TBDRIVER on CL,*

### CICS interface Enhancements and Fixes

ID	Description
2003	Provide sample DFHSRTTB table
2443	Implement CICS Threadsafe application support
2537	Process ABCODE=binzeroes the same as spaces
2557	Remove dangling enqueue if CICS transaction cancelled while waiting for enqueue
2591	Ensure tableBASE is initialized when DK1TCIN is invoked
2592	Add Threadsafe support for Enqueue Wait queue
2601	Ensure PC Server level is appropriate for Threadsafe support

## tablesONLINE/CICS Fixes

ID	Description
2558	Fixes storage violation if TBOL is exited abnormally and TSQUEUE is deleted
2584	Document procedure for removing User-developed TBOL Application

## tablesONLINE/ISPF Fixes

ID	Description
2525	Remove all references to Page tables
2581	Correct command area layout in EXITISPF
2653	Allow pointer tables to be defined

## Installation information

### Changed datasets

With this level the following datasets have changed:

- **LOAD:** Modules are a complete replacement for existing modules of the same name except for DK1TNAME and DK1xBASE. Modules specified in CNTL(AUTHLIB) and CNTL(AUTHLIBV) must be recopied to an authorized library.
- **CNTL:** Source for CICS table DFHSRTTB has been added.
- **CLIST:** The TBPAGE and TBUNPAGE CLISTS have been removed.
- **PANELS:** TBOL/ISPF Panels have been changed. Members TBDEFH, TBDEFINE, TBIDTABH, TBID1H, TBID2H and TBSEARCH are affected.
- **MESSAGES:** TBOL/ISPF Messages have been changed. Members MSG00, MSG02, MTB02 are affected.
- **SRC:** TBOL/ISPF EXITISPF sample program has been updated.

### General Notes

- The PC Server must be restarted using the updated authorized library.
- All CICS regions must be restarted after updating the DK1TCIN CICS definition.
- All VTSs must be restarted using the updated authorized library.

## Batch, CICS, IMS, DB2 SPAS, VTS

1. Replace your existing Version 6 loads with the new Release 6.0.2 Level 7 loads except for DK1TNAME and all DK1xBASE modules. Use your current versions of those modules. Using CNTL(AUTHLIB) and CNTL(AUTHLIBV), copy the specified modules to the authorized libraries.
2. Copy changed members of the Release 6.0.2 Level 7 CNTL, CLIST, PANELS, and MESSAGES PDSs to your corresponding Version 6 PDSs. If you have modified any of these members, reapply your changes.

## CICS

1. Ensure that the Program definition for DK1TCIN has CONCURRENCY=QUASIRENT in all CICS regions.
2. If CICS transactions access tableBASE VTS TSRs, evaluate the sample SRC(DFHSRTTB) file for applicability in your environment.

## TBOL/CICS

Eight load modules have been updated.

## TBOL/ISPF

Copy changed members of the Release 6.0.2, Level 7 PANELS and MESSAGES PDSs to your corresponding Version 6 PDSs. If you have modified any of these members, reapply your changes. Remove members TBPAGE and TBUNPAGE from your Version 6 CLIST PDSs.



# 4

## Level 6

These notes cover the changes to tableBASE Version 6 between maintenance level 5 and maintenance level 6. The major components of this release were:

- Command & jobstream consistency
- New LV command
- New tableBASE option parameters

### Operation, Control and Usage Change

An extra parameter in CClose command is better tolerated than in prior levels. In V5 CClose command processing ignored parameters following the command area. In Version 6, staging for future enhancements introduced support for an additional parameter. This created problems for applications which had erroneously coded a parameter. With this fix level, the parameter is scanned for an eye-catcher before being processed.

### Command consistency

The LT command is now consistent with Version 5 in the return of base table name for index tables.

The IK, RK, IC, RC commands are now consistent with Version 5 in the use of override length value.

The FC, IC, RC, DC, DU commands are now consistent with Version 5 in the setting of the count field after an implicit open.

The DU command is now consistent with Version 5 in the TBACC-DEF-Block contents and the format of the returned rows.

## Jobstream control statement consistency

The sample CNVB DIAG jobstream control statements are now consistent with the DDNAME statements

## TBDRIVER

TBDRIVER utility OA, AR commands are now consistent with the V5 version of these commands. Messages for these commands have been updated to be more accurate. Performance of both commands has been improved, particularly if multiple wild-carded commands are issued to the same library in succession.

## TBDRIVER CLIST

TBDRIVER CLIST specifies a secondary allocation for the output dataset.

## Table designation terminology standardization

Table designation terminology has been standardized:

- Data table: contains rows of user data (previously this was sometimes called a “base” table)
- Index table: contains an alternate index to a Data table (created by CA or IA command)
- View table: contains a description of the columns of a data table (created and used in TBOL)

## Non-standard calls to tableBASE

Non-standard calls to tableBASE in existing applications (e.g., assembler calls without the VL parameter) can cause incorrect output or abends with Version 6. Enhancements to tableBASE commands to accept additional parameters or implement reserved fields in TB-PARM and COMMAND-AREA rely on standard MVS CALLs as implemented in High Level Languages (HLLs).



## Enhancements

### LV command

A new command, LV, has been added. It returns current VTS name settings for the region. See the “Programming Guide: tableBASE Commands” for additional information.

### User exits

Two new user exits for thread start and thread end have been added. In CICS they are invoked at transaction start and end. See the “Administration Guide: tableBASE User Exits” for additional information.

### Library directory caching

Library directory caching has been extended to Bridge Libraries. See “[Known Issues](#)” on page 42 for limitations on all directory caching. See “Administration Guide: Best Practices” for additional information.

### TBEXEC new parameter

TBEXEC supports a new parameter on the COPY command, NEWID=NO, which copies tables to another library without changing the user-id and last-changed date. See the “Batch Utilities Guide: TBEXEC Commands” for additional information.

### New tableBASE option parameters

tableBASE option parameters LOCKTIMEWTO and LOCKTIMERC have been added. With these parameters an application that issues a tableBASE command that waits an excessive amount of time for a table in a TSR to be available will generate message “DK100227W JOB jjjjjjj IN vvvvvvvv WAITING FOR nnnn SECONDS FOR TABLE tttttt” and/or receive error code 71. See the “Installation Guide: Appendix: Parameters” for details.

## Known Issues

### Thread-safe CICS transactions

Thread-safe CICS transactions are not supported. They will be in a future maintenance level.

### Empty directory blocks

Some earlier versions of the tableBASE library conversion process did not clear all empty directory blocks. The Library Directory Caching Feature does not tolerate empty directory blocks. To correct this problem, run the DK1TLCHK utility DIAGNOSE function against the library. If it indicates there are empty directory blocks, run the TBEXEC utility COPY LIB function to recreate the library. This process will remove empty directory blocks.

### tableBASE abend

If tableBASE is accessed from a batch or IMS region that also accesses CICS with EXCI, tableBASE may abend. This can be circumvented by calling TBLBASE with a TBPARM before invoking EXCI and using the TBPARM on all subsequent calls to tableBASE.

## Enhancements and Fixes by Interface

### Enhancements and Fixes Affecting all Interfaces

ID	Description
2271	Implement new LV command to display VTS name accessed
2283	Implement new CICS transaction and Batch thread start and end exits
2294, 2467	Correct U301 on CL or RL after opening table from library with DISP=SHR and diverting it to library with DISP=OLD
1814	Correct missing table NQ when library DISP=OLD
2382	Prevent NQ Deadly Embrace on multiple updates to the same library
2482	Prevent potential library corruption problems due to library NQ synchronization errors
2492	Correct NQ order for RN, XT commands to prevent deadly embrace
2502	Correct NQ order for all library update commands
2507, 2518	Prevent potential tableBASE library corruption due to NQ/DQ logic with caching
2297	Enhance TBEXEC COPY to preserve update history for a table
2300, 2335,	
2339	Correct DU command processing
2301	Implement internal trace for NQ and DQ
2314	Implement new error code and WTO for stalled pre-empt
2315	Prevent S053 in IMS MPR after soft abends
2316	Correct clearing of pre-empt field after termination processing
2422	Disallow pre-empt for TCE LIST and SPACEMAP locks
2320	Correct S0C4 in Multitasking Batch accessing BDAM library
2324	Enhance space request calculation algorithm for RF using algorithm from OR/OW
2328	Correct FC processing with implicit open on
2329	Prevent U300 ABEND for ALESERV macro, error 12/C
2331	Prevent U301 while opening 'true' hash table with row size > 3120
2343	Enhance CL/RF commands to tolerate extra parameter
2356	Prevent memory leak during library access
2379	Enhancement to allow caching of Bridge Libraries for performance improvements
2412	Prevent S0C4 in CICS task termination after CL error 74
2417	Prevent CA, ST & DG ABENDS with MULTITASKING=Y
2418	Prevent U301 in DL on Undefined DD with MULTITASKING=Y
2426	Correct problem with missing tables on cached libraries on LD and NX commands
2434	Set User Exit ABEND indicator
2435	Prevent S0CA in strobe reporting when page number exceeds 999

<b>ID</b>	<b>Description</b>
2447	Correct LT command to always return BASE table name
2475	Disallow batch updates to table opened for read in a VTS-TSR
2478	Correct IK to be compatible with V5 when override length < key length or negative
2480	Correct RK to be compatible with V5 when override length < actual or negative
2483	Implement debug trace to capture library allocations
2485	Correct IC and RC to be consistent with IK and RK when override length negative
2495	Update messages table, TBASEMSG, to contain latest error codes and subcodes
2498	Prevent S878 batch ABENDS at termination
2508	Correct TSR space estimation too large for V6 tables converted from V5 table

## Batch Interface Enhancements and Fixes

ID	Description
2325	Correct sample distribution JCL for CNVB DIAG
2182	Prevent 1072 abends when connecting to a recycled VTS-TSR after IMS transactions stopped and restarted
2304	Correct conversion problem when middle generation of source table is paged
2307	Correct conversion problem when source library is full
2515	Correct conversion program to remove error for spurious FSDNEXT value
2281	Correct AR and OA problem of not returning any error messages with blank ML list
2391	Correct OA and AR behavior to be compatible with V5
2330	Enhance batch driver to display error sub-code for OA and AR functions
2427	Enhance OA and AR functions for performance and concurrent usage by multiple users
2318	Correct batch driver misleading message with wildcard function when TSR is full or almost full
2303	Correct TBEXEC UNLOAD function to initialize output record
2323	Correct misleading TBEXEC IMPORT error message carried over from previous failure
2359	Correct TBEXEC to remove erroneous line, 'XXXXXX', in banner
2410	Correct U333 in TBEXEC when NOWAIT is on
2458, 2460	Correct TBEXEC error handling to return non-zero return code and write error messages to TBMSG
2464	Correct TBEXEC LOAD to not create an empty generation of the table when input file does not exist
2381	Correct multi-tasking batch to handle more than 20 tasks

## tablesONLINE/CICS Fixes

ID	Description
2309	Correct error G012 trying to update TBOLCNST table
2322	Update TBOLMSGs to contain latest error codes and subcodes

## tablesONLINE/ISPF Fixes

ID	Description
2416	Correct sample TBDRIVER CLIST to allow secondary allocation for output DSN
2431	Revise table designation terminology in ISPF Panels (BASE vs DATA)

## Installation information

With this level the following datasets have changed:

- **LOAD:** Modules are a complete replacement for the existing modules. Modules specified in CNTL(AUTHLIB) and CNTL(AUTHLIBV) must be recopied to an authorized library.
- **CNTL:** Jobstream CNVB DIAG has been updated; source for CICS table DFHSRTTB has been added; Jobstream TBOL600V has been modified
- **CLIST:** the TBDRIVER and TBALLOCS CLISTs has been updated
- **SRC:** Sample exits DK1XU\* have been updated; the CICS definition of program DK1TCIN has been updated in member TBASE60; a new copy of macro TBOPTGEN is provided
- **PANELS:** TBOL/ISPF Panels have been changed. Members TBCREALT, TBDEFH, TBIDTABH, TBID1H, TBID2H and TBDEFH are affected.
- **MESSAGES:** TBOL/ISPF Messages have been changed. Members MSG00, MSG02, MTB02, MTB07, MTB09 are affected.
- **EDUCDATA:** Member TBPARMS has been updated.
- **TBSYSLB:** Table TBASEMSG used in the TBDR transaction has been changed and must be replaced to the production library.
- **TBAPPLB:** This tableBASE library is used in TBOL/CICS. TBOLMSGs has been updated and must be replaced in the production library.

## General Notes

- The PC Server must be restarted using the updated authorized library.
- All CICS regions must be restarted.
- All VTSeS must be restarted using the updated authorized library.

## Batch, CICS, IMS, DB2 SPAS, VTS:

1. Replace your existing Version 6 loads with the new Release 6.0.2, Level 6 loads. Using CNTL(AUTHLIB) and CNTL(AUTHLIBV), copy the specified modules to the authorized libraries.
2. If any changes have been made to DK1Txx34 (tableBASE parameters) source modules, they must be reassembled using the new copy of macro TBOPTGEN. The corresponding DK1xBASE modules must be relinked. See CNTL(ALT1134, ALT1334, ALT1434, ALT2734, ALV1134).
3. Copy changed members of the Release 6.0.2, Level 6 CNTL, CLIST, SRC and EDUCDATA PDSes to your corresponding Version 6 PDSes. If you have modified any of these members, reapply your changes.

## CICS

1. Ensure that the Program definition for DK1TCIN has concurrency 'quasi-reentrant' in all CICS regions.
2. The Release 6.0.2, Level 6 TBSYSLB has changes to table TBASEMSG. If you have no changes to TBSYSLB, replace TBSYSLB with the Level-06 version. If you have changes, the following sample JCL will invoke TBEXEC to copy just TBASEMSG:

```
//          EXEC PGM=TBEXEC
//TBMSG     DD SYSOUT=*
//TBRPT     DD SYSOUT=*
//DSYSLIB  DD DISP=SHR,DSN=distribution.TBSYSLB
//TSYSLIB  DD DISP=SHR,DSN=target.TBSYSLB
//CNTLCARD DD *
COPY TBL=TBASEMSG LIB=DSYSLIB TO=TSYSLIB REPLACE=YES;
//
```

## TBOL/CICS

Submit a DK1TEXEC utility job to copy table TBOLMSGs from the Release 6.0.2, Level 6 TBAPPLB to your Version 6 TBAPPLB. Sample JCL follows:

```
//          EXEC PGM=TBEXEC
//TBMSG     DD SYSOUT=*
//TBRTPT    DD SYSOUT=*
//DAPPLIB   DD DISP=SHR,DSN=distribution.TBAPPLB
//TAPPLIB   DD DISP=SHR,DSN=target.TBAPPLB
//CNTLCARD  DD *
COPY TBL=TBOLMSGs LIB=DAPPLIB TO=TAPPLIB REPLACE=YES;
//
```

## TBOL/ISPF

Copy changed members of the Release 6.0.2, Level 6 PANELS and MESSAGES PDSEs to your corresponding Version 6 PDSEs. If you have modified any of these members, reapply your changes.



# 5

## Level 5

These notes cover the changes to tableBASE Version 6 between maintenance level 4 and maintenance level 5.

### Operation, Control and Usage Change

#### Implicit open

Implicit open of a table is now compatible with V5 in setting of the count field to zero.

#### OA command

TBDRIVER utility OA command issues additional informational messages when no tableBASE library is accessed.

### Enhancements

#### TBEXEC

TBEXEC has been updated to display tableBASE error codes and subcodes on all tableBASE errors detected.

#### DK1TCNV

The library conversion utility DK1TCNV has been updated to support RECFM=FB as well as RECFM=F for BDAM table libraries.

## Known Issues

With this level an enhancement to the CL command is being staged. This enhancement uses a parameter with the CL command. If existing applications use a parameter on the CL command, they will receive error code 2-98 and possibly abend.

Multi-tasking applications which access tableBASE from multiple TCBs within a region may experience problems with I/O to tableBASE libraries. This can be circumvented if all library accesses are from the same TCB.

## Enhancements and Fixes by Interface

### Enhancements and Fixes Affecting all Interfaces

ID	Description
2246	Prevent memory leak when opening Alternate for which data table does not exist
2273	Reset count field on implicit open of table
2276	Extraneous table ENQ when table not opened
2282	Prevent G301 abend attempting to access table that has been deleted from library
2290	Correct S0C4 when storing hash table with very high density in a bridge library.

### Batch Interface Enhancements and Fixes

ID	Description
2042	Prevent abend 337 in Expand process when actual library and internal definition size do not match.
2278	Prevent S878 abend when running DK1TCNV to convert a large library
2281	Add messages for TBDRIVER AR command when no library accessed
2284	Enhance TBEXEC to add error code and sub-code displays for tableBASE errors
2287	Correct abend 301 in TBCOMP when tables being compared are on same library
2295	Enhance to allow RECFM=FB in source and target libraries for conversion job DK1TCNV

## CICS Interface Fixes

ID	Description
2277	Prevent G300 abend when tableBASE Library file is 'Disabled'

## tablesONLINE/CICS Fixes

ID	Description
2274	Prevent G001 when working with Version 5 exits in TBOL/CICS
2277	Revise messages in TBASEMSG, TBOLMSG and BKUPMSG for new subcodes

## VTS Fixes

ID	Description
2286	Correct AUTHLIBV sysin card.

## Installation information

With this level the following datasets have changed:

- **LOAD:** Modules are a complete replacement for the existing modules. Modules specified in CNTL(AUTHLIB) and CNTL(AUTHLIBV) must be recopied to an authorized library.
- **CNTL:** Jobstream BDMTOVSM has been updated. Jobstream AUTHLIBV has been updated.
- **TBSYSLB:** Table TBASEMSG used in the TBDR transaction has been changed and must be replaced in the production library. Tables tBOLCNST, TBOLCNST and TCNSHELP used in tbo/CICS have been changed and must be replaced in the production library
- **TBAPPLB:** This tableBASE library is used in tbo/CICS. TBOLMSG and BKUPMSG has been updated and must be replaced in the production library.

## General Notes

All CICS regions must be restarted.

## Batch, CICS, IMS, DB2 SPAS, VTS:

1. Replace your existing Version 6 loads with the new Release 6.0.2, Level 6 loads. Using CNTL(AUTHLIB) and CNTL(AUTHLIBV), copy the specified modules to the authorized libraries.
2. Copy changed members of the Release 6.0.2, Level 6 CNTL PDSes to your corresponding Version 6 PDSes. If you have modified any of these members, reapply your changes.

## CICS

The Release 6.0.2, Level 5 TBSYSLB has changes to table TBASEMSG. If you have no changes to TBSYSLB, replace TBSYSLB with the Level-05 version. If you have changes, the following sample JCL will invoke TBEXEC to copy just TBASEMSG:

```
//          EXEC PGM=TBEXEC
//TBMSG     DD SYSOUT=*
//TBRPT     DD SYSOUT=*
//DSYSLIB  DD DISP=SHR,DSN=distribution.TBSYSLB
//TSYSLIB  DD DISP=SHR,DSN=target.TBSYSLB
//CNTLCARD DD *
COPY TBL=TBASEMSG LIB=DSYSLIB TO=TSYSLIB REPLACE=YES;
//
```

## TBOL/CICS

1. The Release 6.0.2, Level 5 TBSYSLB has changes to tables tBOLCNST, TBOLCNST and TCNSHELP. If you have no changes to TBSYSLB, replace TBSYSLB with the Level-05 version. If you have changed any tables in this library, the following sample JCL will invoke TBEXEC to copy just these tables. If you have changed the TBOLCNST table (e.g., through the EDIT CONSTANTS option on the TBOL/CICS Administrator's Menu), you will have to reapply those changes.

```
//          EXEC PGM=TBEXEC
//TBMSG     DD SYSOUT=*
//TBRPT     DD SYSOUT=*
//DSYSLIB  DD DISP=SHR,DSN=distribution.TBSYSLB
//TSYSLIB  DD DISP=SHR,DSN=target.TBSYSLB
//CNTLCARD DD *
COPY TBL=tBOLCNST LIB=DSYSLIB TO=TSYSLIB REPLACE=YES;
COPY TBL=TBOLCNST LIB=DSYSLIB TO=TSYSLIB REPLACE=YES;
COPY TBL=TCNSHELP LIB=DSYSLIB TO=TSYSLIB REPLACE=YES;
//
```

2. Submit a DK1TEXEC utility job to copy tables TBOLMSGs and BKUPMSGs from the Release 6.0.2, Level 5 TBAPPLB to your Version 6 TBAPPLB. Sample JCL follows:

```
//      EXEC PGM=TBEXEC
//TBMSG  DD SYSOUT=*
//TBRPT  DD SYSOUT=*
//DAPPLIB DD DISP=SHR,DSN=distribution.TBAPPLB
//TAPPLIB DD DISP=SHR,DSN=target.TBAPPLB
//CNTLCARD DD *
COPY TBL=TBOLMSGs LIB=DAPPLIB TO=TAPPLIB REPLACE=YES;
COPY TBL=BKUPMSGs LIB=DAPPLIB TO=TAPPLIB REPLACE=YES;
//
```



# 6

## Level 4

These notes cover the changes to tableBASE Version 6 from tableBASE Version 5 through maintenance level 4.

### Operation, Control and Usage Change

#### TSR usage

All address spaces (regions) accessing tableBASE require TSRs. Previously batch jobs could optionally operate without TSRs (tables were individually loaded in getmained storage). An appropriate default TSR size for batch applications must be determined.

All TSRs are allocated in MVS dataspace; in previous versions non-VTS TSRs were allocated from getmained storage within the address space. Installation limits on dataspace usage may need to be reviewed.

#### TBTSLIB tableBASE library

The TBTSLIB tableBASE library, used for TSR overflow in previous versions, is no longer used. The DD can be removed from JCL (and CICS definitions).

#### Paged tables

Paged tables are no longer supported. They must be converted to another table format.

#### PC Server

Version 6 introduces the tableBASE PC Server. This task should be started at system initialization and never be terminated.

## **Re-initialization of the Program Call Server**

If the PC Server is brought down for any reason, all CICS regions that have already completed tableBASE initialization will continue to operate. Any regions which attempt to initialize while the PC Server is down will fail. When the PC Server is brought back up, all new regions will initialize at the new PC Server's level (if applicable). All regions still run using the previous PC Server's level (if different).

For CICS, the TBST TERM transaction can be used, followed by the TBST TBINIT transaction to re-attempt initialization.

## **tableBASE system exits**

The tableBASE system exits have been changed. If tableBASE exits were used in previous versions and are still needed, the new interface will be required. (See TBOL implementation in the installation manual for any changes to TBOL exits).

## **MULTOPNX**

Option MULTOPNX can only be set to Y. Any applications depending on this option being set to N must be updated.

## **ISPF applications and DB/2 Stored Procedures**

ISPF applications and DB/2 Stored Procedures accessing tableBASE must be linked RMODE 31, AMODE ANY.

## **XX command**

The XX command is effectively a NOP; in previous versions it closed paged tables.

## **R/W VTS-TSRs**

VTS-TSRs are now updatable. In previous version VTS-TSRs were read-only.

## **VTS-TSR initial load and refresh processes no longer supported**

The VTS-TSR initial load and refresh processes are no longer supported. Version 6 TBDRIVER supports the commands and functions of these processes.



## Enhancements

The following enhancements were added for Level 4:

1. Reentrant multitasking tableBASE engine.
2. DK1TCALL, the new API stub, can be used in all interfaces. Its ALIASEs include TBLBASE and API stubs from earlier releases. Since DK1TCALL dynamically determines its environment, it replaces the interface-unique TBLBASE modules distributed with previous versions. For compatibility, no relinking of tableBASE programs accessing previous APIs is required.
3. New options have been added to the interface defaults (DK1Txx34 modules) and to the TBOPT runtime overrides. Options can be listed when tableBASE initializes.
4. Under CICS, the TBOPT file may be QSAM (including DD \*). Previous versions required it to be VSAM.
5. Diagnostic functions have been enhanced. The TBDUMP dataset will contain a diagnostic dump of significant tableBASE control blocks.
6. TBEXEC EXPORT functionality has been improved; specific tables can be EXPORTed.
7. TBEXEC EXPAND is no longer limited to specific ranges of the library size.
8. An enhancement to the tableBASE API to support null parameters simplifies calls from C++ programs.
9. Library access performance can be improved significantly in some cases by using directory caching.
10. CICS transactions that wait for tableBASE table enqueues are supported with less CICS overhead.
11. To enhance the protection offered by CICS Storage Protection, tableBASE executes most of its code in the key in the transaction program definition.
12. tableBASE Library blocksize can now be set at definition to any value from 800 to 32760. In previous releases the library blocksize was 3120 bytes.
13. For applications that do not use the same tableBASE command area for successive calls to access the same table, table lookup performance has been significantly improved.

## Known Issues

The following issues are known to exist in Level 4:

1. At this level threadsafe applications under CICS are not supported. All applications accessing tableBASE must be defined as CONCURRENCY=QUASIRENT.
2. All tableBASE libraries must be converted to a Version 6-compatible format before being accessed by tableBASE Version 6. Previous releases cannot access the Version 6-compatible format. However, the Version 6-compatible format can be converted back to Version 5 format. A Library Bridge product is available to allow concurrent access to a common format library from both Version 5 and Version 6.
3. We strongly recommend that all users run the Version 6 tableBASE PC Server in every MVS Image in which tableBASE is used. If it is not, tableBASE module DK1TROTb will be updated at run-time with the address of a step termination routine. Since DK1TROTb is distributed as reentrant, it will be loaded in key 0 memory if it is loaded from LPA or an authorized load library and thus cannot be updated. This can be circumvented by relinking it as serially reusable.
4. If any reentrant applications do not use a TBPARM when calling tableBASE, there can be performance problems when DK1TROTb and/or DK1TCALL are loaded from LPA or an authorized load library. As above, they will be in loaded in key 0 memory. Without a TBPARM, our fast path pointer is not available so an MVS LOAD of DK1TVROT must be issued on every call to locate DK1TROTb. Then an MVS LOAD of DK1TNUCL must be issued. The LOAD of DK1TNUCL can be circumvented by relinking DK1TROTb as serially reusable. If the application module issuing the call to our API (TBLBASE or some alias) is reentrant or a dynamic call is used, DK1TCALL must be relinked as serially reusable. Then the addresses of the modules can be cached and the LOADs bypassed.

## 7

# Release 6.0.2 (Service Pack 6)

## Introduction

tableBASE Release 6.0.2 Service Pack 6 provides enhancements, fixes of known issues in Release 6.0.2 and support for the Library Bridge product (tableBASE 5.B).

Version 6 offers significant enhancements and improvements, such as multi-tasking capabilities, that also require additional consideration during implementation. A section is included in the Release Notes that covers considerations that should be reviewed carefully by those moving to this new release (see [“Cautions”](#) on page 34).

For users already on Release 6.0.1, the information in this guide includes sections indicating the changes in Release 6.0.2. For users moving from Version 4 or 5 of tableBASE, or who are new to tableBASE, please refer to all sections of the document to fully understand all the information that applies to all release levels of Version 6.

tableBASE Version 6 operates with z/OS and OS/390, and is compatible with CICS TS, IMS TM, DB2 and batch.

### Moving to 6.0.2 Service Pack 6 from 6.0.2 or from 6.0.1

- Release 6.0.2 Service Pack 6 is a complete replacement for all executable modules from Release 6.0.2 or Release 6.0.1.
- Release 6.0.2 Service Pack 6 tableBASE libraries are compatible with all Version 6 libraries (Version 6 libraries, however, are not compatible with Version 5 libraries).

### New in tableBASE 6.0.2 Service Pack 6

- In all prior releases, when copying tables or entire libraries using TBEXEC, tableBASE updates the USERID and the CREATION DATE/TIME fields in the definition of each copied table according to the copying task. With service Pack 6, the COPY command of TBEXEC uses the NEWID=NO parameter to prevent the update to the table definition with USERID and DATE/TIME of the copy task. The default value NEWID=YES, assures that TBEXEC COPY command works as prior releases.

- Two new TBOPT parameters are introduced to address both the handling and the reporting of unusual table lockout conditions. The parameter LOCKTIMERC specifies the maximum time tableBASE should wait for a lock and LOCKTIMEWTO specifies the time to wait before issuing a message that a process is waiting for a lock.
- A new TBLBASE API command, LV allows a program to discover the settings of the VTS parameters VTSONLY, VTSNAME, VTSFIRST and VTSLAST.
- A new pair of additional user exits that allows the trapping of thread start and end.

## New in tableBASE 6.0.2

- Most significant is the support for the Bridge library format. The Bridge format is created with the Library Bridge product (tableBASE 5.B) to facilitate the migration from Version 5 and older versions of tableBASE.
- Incorporates all fixes of known issues with earlier releases.
- The fix for the data integrity issue in 6.0.1 when the Status Switch WAIT is set to Y (see Chapter 3 of the *tableBASE Programming Guide, Release 6.0.3*) in your tableBASE environment. Release 6.0.1 causes the waiting application to access the previous generation of the table, thereby potentially losing the changes of the current generation.
- The library conversion and diagnostic tools of 6.0 and 6.0.1 (DK1TLBFX, DK15CNV1 and DK15CNV2) have been replaced by two new utilities: DK1TCNV and DK1TLCHK.
- Improved error detection in the table update and store functionality.

## Highlights of tableBASE Version 6, all release levels

- tableBASE Version 6 provides a fully re-entrant engine that extends tableBASE performance benefits to multi-tasking applications.
- Virtual Table Share (VTS-TSR) provides read and write access to shared tables from any region.
- tableBASE uses Data Spaces for local TSR and VTS-TSR, allowing for an increased size up to 2 G.
- Support for DB2 stored procedures managed either by DB2 or by Workload Manager (WLM) is introduced in this release.
- OTE functionality is supported in CICS Transaction Server 2.2 through 3.1. Support of OTE reduces the number of TCB switches and consequently the CPU usage.
- A single unified API (Application Programming Interface) that supports all prior-release stubs, for all environments, with a single link-edit include library.
- The internal locking of objects within tableBASE has been greatly improved. For the user this means enhanced performance.

- Many code paths have been shortened to provide faster, more efficient access to tables. This is particularly true of access to VTS-TSR tables where the code path is now the same as that for the local TSR.
- Enhanced performance for initial table access and for table access without the table's handle (also known as the pivot). A more efficient algorithm is used to significantly improve the time needed to locate a table. The table's handle (pivot) is preserved in the command area for subsequent accesses to the same table (see [“Table access optimization”](#) on page 29).
- Improved internal tableBASE locking ensures the fastest speed for table access and reduces contention.
- Improved API simplifies tableBASE access for C++ and Java applications — the Version 5 API required the user to specify a variable number of parameters for calls to tableBASE. The variable number of parameters depends on the command specified. For C++ and Java this syntax was cumbersome. With Version 6, four parameters may be specified on every call. The user enters null pointers for the parameters that are not required for the specified command.
- New capabilities have been added to the EXPORT function available in the TBEXEC utility to allow exporting by table rather than just by library. The IMPORT command is unchanged and continues to work on all tables specified in the input file.
- Tightened memory management supports 24x7 use of tableBASE.
- Improved library conversion process makes the move from Version 5 libraries to the new Version 6 format easier.
- Improved open table recovery in error situations is automated. This feature requires the PC Server to be operational.

## Additional enhancements in Version 6, all release levels

### Performance enhancements

- A new memory-model provides more efficient memory management of the TSR.
- MVS paging of TSR pages is reduced. This should result in reduced overhead in the tableBASE regions and less paging by the operating system.
- tableBASE libraries can have custom blocksizes. If the tables are large or have a big row size, a custom blocksize allows for fewer I/Os to read and write tables. Libraries with different blocksizes may be used in the same tableBASE Library List.
- Library directory caching improves time to load tables into memory or to store to disk, particularly when multiple tables are being opened, which is the case during the initial loading of a TSR.
- Two new methods are used to put CICS transactions into wait state if the transaction elects to wait while opening for write a table that is already open for write in another region. These methods do not affect the QR TCB and they do not use excessive CPU as did the technique used in prior releases.

- Almost all Version 6 modules are re-entrant, making them candidates for LPA. A list is provided in the *tableBASE Installation Guide, Release 6.0.3* Appendix "tableBASE LPA-eligible programs".
- Batch applications can be built using multitasking and tableBASE will handle each task as a separate entity with its own execution state.

### Operational enhancements

- The tableBASE Root and Nucleus modules are above the line, as is the tableBASE CICS Resource Manager.
- Additional TBOPT parameters have been added to provide runtime overrides for most of the installation options.
- Version 6 I/O routines now tolerate open failures and I/O errors that may have previously resulted in region failures.
- Improved diagnostics are provided with the use of error subcodes in the extended command area.
- Individual libraries may be set to read-only access.
- The Version 6 stub is re-entrant, allowing you to build fully re-entrant application load modules.
- A library of any size may be expanded.
- All API stubs are shown in CICS EDF tracing. In previous releases, TBCALLC, TBASEV, TBCALLV were not captured by the EDF trace.
- If a CICS task is put into a wait for a table exclusively held—opened for write—by another region, a CEMT INQUIRE task will show TBLBASE in the Suspend Value field.
- STGPROT may be used with TBASEV and TBCALLV.
- The re-entrant API stub allows for re-entrant applications that can be used with the CICS RENTPGM feature.
- Support for CICS storage protection keys is enhanced. Almost the entire tableBASE execution path is in the key set by the installation's CICS EXECKEY definitions, affording the protection of this CICS feature.
- The CICS TBOPT file may now be QSAM (including DD \*) or, as it was in previous releases, VSAM.
- More flexibility in specifying TBOPT input, more options, and the ability to list all options have been added.
- The TBOPTV functionality has been integrated into TBOPT, allowing for a single source of runtime parameter input. TBOPTV is still maintained for backwards compatibility.
- The allocation of VTS-TSRs is done in a way that does not effect your site's MVS MAXCAD parameter.
- The default data-sensitive processing date rolls over at midnight.

## Administration and maintenance enhancements

- A common naming structure for all load modules has been implemented for easy identification. It is DK1xxxxx. tablesONLINE, an optional component of tableBASE, does not yet follow this convention.
- Ongoing maintenance is reduced with the introduction of a single load library that contains all Version 6 tableBASE modules.
- The Version 6 install process is simplified. The process is similar for all interfaces (batch, CICS, IMS, and VTS-TSR) and there is a uniform treatment of the install options.
- A library diagnosis utility is provided that allows for detection of errors in tableBASE libraries.

## Exceptions to compatibility with previous versions

tableBASE 6.0 is compatible with Versions 4 and 5 with the following exceptions:

- All tableBASE libraries must be converted to Version 6 or Library Bridge format before use with tableBASE Version 6.
- With the TSR now resident in Data Spaces and up to 2 GB in size. There is no need for paged tables and ROLL-IN/ROLL-OUT/TBTSLIB. The utility program DK1TCNV integrates the conversion of paged tables to memory-resident tables in the library conversion process. Please refer to the *tableBASE Installation Guide, Release 6.0.3* for more information.
- The TBOPT parameter MULTOPNX=Y refers to the capability to open (OR) multiple alternate indexes. In Version 6, this parameter will only accept the value Y for backwards compatibility purposes. If it is set to N the parameter is ignored. See [Chapter 2 “Alternate Indexes”](#) on page 12.
- The VTS Agent no longer runs the CMD file during initialization. A VTS-TSR can be loaded by a separate DK1TDRV job, or by user-written applications.
- With Version 6, the interface to the user exits—formerly called system exits—has been formalized to provide a well-defined interface to user exit code which should be stable over subsequent releases. However, it is not compatible with that of prior releases. Please refer to the *tableBASE Administration Guide, Release 6.0.3* for more information.
- The Refresh functionality used with the optional VTS component has been modified and will require changes in the applications that use this functionality. However, with the upgrade of VTS to read/write functionality, the Refresh functionality may no longer be needed.
- In Version 6, it is possible to release a table (RL) that has just been defined (DT) or renamed in the TSR (CN). In Version 5, the sequence DT, RL generated error code 38.
- [Release 6.0.2, Level 6] The Count field in the Command Area (TB-COUNT) is zeroed for implicit opens, except for FC, IC, RC, DC, and DU commands. This is the same behaviour as Version 5. Previous Releases of Version 6 operated as:

- [Release 6.0.2, previous Levels] Count was zeroed for all opens, whether implicit or explicit
- [Release 6.0.1] Count was left unchanged for all implicit opens
- In Version 6, all tables are internally maintained as pointer tables.
- In Version 6, the DL command parameter LIB-SPACE is optional. In Version 5, it was not.
- The last parameter must be marked with a high-order bit, and must be set. This is automatic in COBOL, but must be done explicitly in Assembler with the VL parameter on the MVS CALL macro. It may also need to be explicitly set in C and other languages.
- In Version 6, an insert by count command (IC) is successful only if the specified row is empty (in Version 5, a row was added to the end of the table). In either case, the integrity of the hash table is undermined. Results are unpredictable and the hash table may not be usable after the command completes.  
**Note:** Use of the IC command with hash tables is NOT RECOMMENDED.
- In Version 6, invalid linkage conventions in calls to tableBASE from AMODE 24/RMODE 24 applications will cause abends. In Version 5, this was not always true.
- In Version 6, error code 85 is returned for all attempts to issue DV or DW commands against an alternate table. In Version 5, error code 0 was returned in some cases, even though the command was not successful.

## Changes of note

The following changes should be noted:

- The XX command no longer closes paged tables that have been updated because Version 6 does not use paged tables.
- TBACC and TBINDX programs are still available in Version 6, however they have not been modified. These programs continue to access main memory as they did in previous releases.
- With previous releases of tableBASE, the use of the TBCALLV was promoted as a way to access VTS with the shortest path. This is no longer the case with Version 6. A user of TBCALLV offers no advantages in Version 6 and may provide slower performance.
- All ISPF applications and DB2 stored procedures must be linked AMODE(31), and RMODE(ANY).

## Maintenance methodology

Version 6 is no longer maintained by ZAPs.



All levels of Version 6 track maintenance by load module level. A LISTLVL CLIST and job are provided that allow you to list the maintenance level of every tableBASE load module in your library. This information allows DKL support staff to determine whether specific fixes are included in your tableBASE software.

The internal level of Release 6.0.2, when displayed by the LISTLVL CLIST is V601.00006.

## Base product changes

### Naming protocol

Version 6 features the new tableBASE naming protocol.

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

All aliases of TBCALLx/TBLBASE calling stubs have been retained so that no changes are required to your existing applications.

### Multitasking

tableBASE Version 6 is a fully re-entrant engine that extends tableBASE performance benefits to multitasking applications. With a fully re-entrant tableBASE, multiple tasks can open, read, update, and store tables in all TSRs (local or shared).

**Note:** Multitasking applications require the use of the Version 6 stub TBLBASE and the Program Call Server.

The Program Call Server (PC Server) is new in Version 6. The PC Server provides the functions needed for tableBASE multitasking, accessing a VTS-TSR, and compatibility with DB2 stored procedures. For more information please see the *tableBASE Administration Guide, Release 6.0.3*.

tableBASE supports batch multitasking where a primary task attaches one or more subtasks. tableBASE is compatible with CICS multitasking through the use of quasi-re-entrant TCBs (QR-TCB) and CICS 2.2 OTE which uses both QR-TCB and the attaching of sub-tasks. tableBASE is also compatible with DB2 stored procedures managed either by DB2 or WLM.

tableBASE can be tailored to perform optimally in your environment by setting execution-time parameters and switches for both batch and online environments. In a batch environment only, the Multitasking parameter can be set to Y to allow multiple subtasks within a region to access tableBASE concurrently. The normal batch default is N. The default in a DB2 Stored Procedure Address Space is Y.

**Note:** For a full list of execution-time parameters see the *tableBASE Installation Guide, Release 6.0.3*.

## Multiuser (LOCK-LATCH)

The LOCK-LATCH feature has been present in previous releases of tableBASE but it was limited to the TSR opened within a CICS environment. Version 6 expands this feature into a read and write VTS-TSR and batch.

If required, an application may use a LOCK-LATCH to restrict table updates and closes to only those programs that provide the LOCK-LATCH. The scope of this restriction also extends to all Alternate Indexes associated with the locked table.

If Batch applications are moved to a multitasking environment they will require the long COMMAND-AREA to prevent miscellaneous characters from being interpreted as a LOCK-LATCH.

## Above-the-line operation

All Version 6 tableBASE modules can run above the 16M line. However, a small amount of below-the-line memory is required for each active task and for each open library. Those modules that are required to support any RMODE(24) applications will continue to run below the line.

**Note:** Release 4 and 5 stubs operate only in Rmode(24).

## Unified tableBASE stub

The tableBASE API in Version 6 can be used in CICS, batch, IMS, and VTS-TSR operating environments. tableBASE programs that use this stub can be ported from one environment to another without requiring any stub relinking. Of course, programs must continue to conform to the requirements of each execution environment.

The Version 6 stub contains all the old API entry names such as TBASEC, TBCALLI, TBASEV. Applications which have the older stub statically linked will continue to work with Version 6. However, we strongly recommend using the Version 6 stub itself to gain all the benefits of Version 6.

The Version 6 stub, TBLBASE, is distinct from the stub of the same name used in Version 5. The TBLBASE stubs of previous releases were held in physically separate libraries — one for each operating environment. With Version 6, there is only one TBLBASE for all operating environments. In Version 6, TBLBASE also has a shorter instruction path.

If your application dynamically calls the tableBASE stubs, it can immediately take advantage of the faster path by using the Version 6 tableBASE load library in your

STEPLIB concatenation (but subject to the restriction that you are running tableBASE Version 6 and the libraries are converted to the Version 6 format or to Library Bridge format).

The Version 6 stub is re-entrant, allowing you to build fully re-entrant application load modules.

Of course, your stubs for previous releases will continue to work. However, the Version 6 stub is needed to experience the full benefits of Version 6, and we strongly recommend using the Version 6 stub.

**Note:** The Version 6 stub is not backwards compatible with previous tableBASE releases. If the stubs of previous releases are replaced with the Version 6 stub, the application can no longer be used in Version 4 or 5.

During your initial testing of Version 6, it is possible to have both Version 5 and 6 application load modules where corresponding modules differ only by the tableBASE stub, Version 5 or 6, that is linked-in.

It is possible to run programs, such as various CICS transactions with different stubs from Version 4, 5 and/or 6, under Version 6 in a single CICS region.

## Memory model modifications

In Version 6, the memory model has been modified to utilize segmented memory for more efficient memory management. Segmented memory means the space allocated for data rows are no longer contiguous and therefore the rows no longer need to be moved to accommodate updates. Instead, an Index is used for each table to point to the rows. This allows for more efficient use of the local TSR and VTS-TSRs.

Previous to this release, memory for each table had to be contiguous. This presented a number of issues. For example, space was not released until a table was closed. In Version 6, when an entire segment becomes empty the space is freed for reuse.

The data will now be maintained as efficiently as possible as a result of using segmented memory. When there is insufficient memory available to load data together, tableBASE will use whatever space is available — even if it is not contiguous.

This change to the memory model also means that all tables are now stored internally as Pointer tables. In previous Releases, tableBASE allowed two type of tables: Pointer and True tables. True tables were characterized by not having an Index and by being stored in contiguous memory. In 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.

## Dataspaces

tableBASE now uses Dataspaces for the local TSR and VTS-TSR. This allows the size of a TSR to be increased up to 2 GB. The use of Dataspaces also means that TSRs are better-protected against accidental overwrites by faulty application programs, and that no space is taken up for the local TSR from the region's above-the-line virtual memory.

**Note:** Maximum table size is limited by the maximum 2 G size minus the system overhead.

Dataspaces used for the VTS-TSRs do not affect the MAXCAD parameter of the MVS operating system. This prevents a MAXCAD issue which could have potentially occurred in Version 4 and Version 5 in which the allocation of VTS Dataspaces competed with other jobs in the system causing the MAXCAD limit to be exceeded and, as a result, intermittent failures in VTS initialization. An IEFUSI MVS system exit may also limit dataspace usage.

## Paged tables

Version 6 loads the table into the Data Space and allows the operating system to handle paging, rather than having tableBASE paging individual blocks to and from a tableBASE library. The library conversion process converts existing paged tables to non-paged tables as it converts the libraries to Version 6 format.

## Indexes

All tableBASE tables have Indexes in Version 6. The designation of a table as True (Index = T) is only supported for compatibility with previous releases.

All Indexes are dynamically generated when the tables are opened.

## Controlling the number of tables in a TSR

tableBASE requires some space in a TSR to maintain the open table directory. In previous releases, expansion of this directory was disruptive to the tableBASE response time. In this release, the directory limit is set during the tableBASE initialization and is dependent on the TSR size. The size of the open table directory can be reduced by specifying the tableBASE execution-time parameter MAXNMTAB=N. For more information see the *tableBASE Installation Guide, Release 6.0.3*.

**Note:** For a VTS-TSR region, VTSNMTAB is retained for compatibility.

## Error subcodes

Better diagnostics are provided with the use of error subcodes in the extended command area. New applications can make use of this value to pinpoint a reason for failure. If a task has the Abend Switch set ON, a message with the error code and any related subcode is written to the job's JES log. The TBDRIVER utility and the TBDR CICS transaction both display the error subcode.

## User exits

In previous documentation, user exits were referred to as system exits. System exits are reserved for use by DKL and user exits refer to exits created by tableBASE users.

User exits that were available in previous versions of tableBASE have been enhanced and renamed. In addition, a new user exit has been added. This exit gets control after each tableBASE command is processed.

Exits are dynamically loaded at tableBASE start up if the tableBASE execution-time parameter USEREXITS=Y is specified in TBOPT.

**Note:** The interface to the user exits in tableBASE Version 6 is not compatible with that of prior releases. The interface has been formalized to provide a well-defined interface which should be stable over subsequent releases. To allow for future enhancements, user exits are required to set R15 to zero before returning to tableBASE.

For detailed information on user exits, please see the *tableBASE Administration Guide, Release 6.0.3*.

## TBCALLC

The operation of TBCALLC may change for some users.

It is now implemented consistently with TBASEC and TBLBASE. That means all calls to TBCALLC will be considered a single thread.

In prior releases, if the TBCALLC user zeroed the TMA Pointer after the first call (or presented a second TMA-AREA with a zeroed TMA Pointer), tableBASE would create a new thread within the transaction.

In Version 6, only the most recent LIB-LIST will be in effect, even if the TMA was zeroed or a second TMA-AREA was presented. TBCALLC calls are supported through CEDF in Version 6.

## Alternate Indexes

There have been two changes of note with respect to Alternate Indexes.

In previous releases, it was possible to create an Alternate Index with the same name as the Data Table (even though accessing such a table resulted in an endless loop). However, in Version 6 you are no longer able to create an Alternate Index with the same name as the Data Table. An attempt to do so now results in error code 19.

**Note:** If a Version 5 Alternate Index with such a naming problem is imported into Version 6, the library conversion program will delete the problematic Alternate Index and post it on the library conversion report.

If an application opened a Data Table using an Alternate Index in Release 4.x, any attempt to open the Data Table directly or open another Alternate Index resulted in error code 84. In Release 5.x this restriction was lifted but an indicator (MULTOPNX) could be set to N to enforce the restriction for compatibility with Release 4.x. However, in Version 6 this restriction is no longer enforceable. The MULTOPNX option is set to Y and cannot be changed.

## Generation number

The absolute generation number of the table being accessed is returned to the user's extended command area, and can be used to verify that the same generation of the table is being used from transaction to transaction.

## Changes to tableBASE API commands

### Release Table command (RL)

The command that changes a table's status from open for write to open for read (RL) has been changed. In previous releases, if RL was applied to an Alternate Index, the status of the Index and all Alternate Indexes were changed to open for read. An RL command applied to an Alternate Index now demotes the Alternate Index to open for read, but leaves the Data Table as open for write.

In addition, if an RL command was applied to a newly created table using either a Define Table command (DT) or a Change Name command (CN), in previous releases, this would result in a return code of 38. In version 6, this returns a zero. A subsequent attempt to change the in-memory table back to open for write (OW) will only work in single user batch where exclusive access is assured.

## List Open Tables command (LT)

There have been formatting and data changes to the LT command and the display drivers, such as TBDRIVER. Figure 7-1 and Figure 7-2 show the differences between the information displayed for Release 5.1, and Version 6.

- I/O has become L/V (local or VTS-TSR). This flag now indicates where the table resides.
- ROLL INS and ROLL OUT have been removed as there is no longer a roll-out library (TSTSLIB).
- ROWS and RWS-BF-EXP have been added. ROWS indicates the current number of data rows in the table. RWS-BF-EXP indicates the total number of data rows that can be added to this table before the data row space or Index space must be expanded.
- BASE TBL has changed to BASE/VTS. If this LT entry represents an Alternate Index the name of the Data Table appears in this field. If the entry represents a linked table, the name of the VTS-TSR appears in this field. For more information on linked tables, please see “[Linked tables](#)” on page 26.

**Note:** When in a VTS, the display driver shows the VTS name in the top left-hand corner.

If any errors or inconsistencies are found in the inputs to the LT command, -1 is returned in the LIST-BLOCK LIST-TOTAL field in Version 6. Prior releases returned with no indication and no output data.

		----- HIGH WATER MARKS -----				- CURRENT SPACE -			
TSR ENTRIES		OPEN TABLES	TSR USAGE	TSR SIZE	OPEN TABLES	TSR USAGE			
34		7,279,224	7,213,560	7,340,032	7,278,176	7,212,512			
REF	NAME	W/R	I/O	IA?	CALLS	SPACE	ROLL INS	ROLL OUTS	DATA TBL
16	TBSYSCP	R	I	N	59	696	0	0	
17	TBOLPROF	R	I	N	6	14,856	0	0	
18	DD.....	W	I	N	200	18,128	0	0	
19	TBOLLIBX	R	I	N	4	712	0	0	TBOLLIBR
20	DV.....	W	I	N	182	19,296	0	0	
21	BIG	W	I	N	1	6,658,240	0	0	
22	PRIKEY	W	I	Y	66	15,880	0	0	
23	SP.....	W	I	N	12	776	0	0	
24	TBASEMSG	R	I	N	10	11,664	0	0	
25	PRIKEY	R	I	N	96	2,304	0	0	
26	TBDRTUTR	R	I	N	66	67,368	1	1	
27	pRIKEY04	R	I	N	1,344	2,304	0	0	
28	EC.....	W	I	N	887	2,208	0	0	
29	TBSYSALI	W	I	N	1	776	0	0	
30	DATKEY	R	I	N	1	2,376	0	0	PRIKEY

**Figure 7-1: Release 5.1 information for the LT command**

----- HIGH WATER MARKS -----										- CURRENT SPACE -	
TSR		OPEN			TSR	TSR	OPEN		TSR		
ENTRIES		TABLES			USAGE	SIZE	TABLES		USAGE		
18		812K			844K	2,048K	640K		672K		
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+											
REF	NAME	W/R	L/V	IA?	CALLS	SPACE	ROWS	RWS-BF-EXP	BASE/VTS		
1	TBOLUSAP	R	L	N	25	8,192	3	252			
2	TBOLACT	R	L	N	18	8,192	7	14			
3	TBSYSTMP	W	L	Y	13	61,440	0	10			
4	TBSYSID	W	L	N	366	61,440	0	10	TBSYSTMP		
5	EXAMPLE	R	V		1	0	0	0	VTS:DK1S		
6	TBOLPFKS	R	L	N	81	8,192	24	161			
7	TBOLDESC	R	L	N	1299	98,304	416	511			
8	TBOLMENU	R	L	N	1003	20,480	69	90			
9	TBOLCMD5	R	L	N	286	12,288	65	81			
10	TBOLMSG5	R	L	N	176	102,400	703	840			
11	TBOLHELP	R	L	N	3	12,288	53	70			
12	TBOLLIBR	R	L	Y	37	8,192	21	77			
13	TBOLTUTR	R	L	N	8	188,416	1,190	1,443			
14	TBSYSCP	R	L	N	160	8,192	24	504			
15	TBOLPROF	R	L	N	48	12,288	9	21			

Figure 7-2: Version 6 information for the LT command

### Banner Retrieval command (BN)

The BN command returns the customer information, as before. In Version 6, when the command is called with an optional second data area parameter, BN returns the tableBASE version (16 bytes) 'tableBASE V601'.

**Note:** The internal level of Release 6.0.2, when displayed by the LISTLVL CLIST is V601000n, where n is the maintenance level.

### Change Table Definition command (CD)

When the CD command specifies an Alternate Index rather than a Data Table, the following restrictions are now enforced:

- The Table Index field cannot be changed. It must be P (Pointer table).
- The row size cannot be changed. The Data Table's definition determines the row size.
- The expansion factor and upper and lower densities cannot be changed.
- The read and write passwords cannot be changed. The Data Table's passwords must be used to open an Alternate Index.

In prior releases, attempts to change these fields with the CD command may have resulted in RC = 0 even though no change was made. Now, these attempts will result in RC = 85 (The command is invalid for an Alternate Index).



### **Divert Table command (DV)**

In prior releases the DV command did not check whether the target DDNAME was allocated. In Version 6, tableBASE checks to see that the target DDNAME is allocated and return an error code 40-1 if it is not.

### **Fetch Next by key (FN)**

This is a new command that has been added in Version 6 that allows you to search for a row that is greater than or equal to the key if the table is in ascending order, or for a row that is less than or equal to the key if the table is in descending order.

### **Get Next Table Name command (NX)**

The *tableBASE Programming Guide* for previous tableBASE releases stated that the LIB-SPACE parameter is ignored unless the COMMAND-AREA table name is blank or contains low values. In fact, tableBASE checked only for low values. Version 6 uses the LIB-SPACE parameter if present, independent of the command-area table name.

Previous *tableBASE Programming Guides* also stated that if the DDNAME parameter is not provided, the first entry in the LIB-LIST is used. However, if the DDNAME parameter is blanks or is null, then the first LIB-LIST entry is also used. Version 6 also operates in this manner.

### **Open For Write (OW)**

In Version 6, a security hole was repaired that was introduced in Version 5. In previous releases, an open for write with a LOCK-LATCH used on an Alternate Index when the Data Table is already open for read, promotes the open table from read to write. This behavior is correct. However, previous releases also allowed you to update the table either directly or through the Alternate Index and then store it without the LOCK-LATCH. In Version 6, the LOCK-LATCH is required to update or store the table.

### **List Directory (LD)**

In Version 5, the LD command returned zero in the COUNT field. In Version 6, the COUNT field contains the total number of directory entries in the table.

If the TABLE containing the directory entries is already open and a subsequent LD command has the same format as the prior request, then the new information will be appended to the end of the table and the COUNT field will be updated to reflect the total number of directory entries in the table.

## Close Table (CL)

In Version 5, the CL command returned an error code of zero when the table name was invalid; indicating that the command was successful. In Version 6, this anomaly has been corrected and the CL command issued with an invalid table name returns error 0012.

## Libraries

This section provides information on the Libraries used in all release levels of Version 6.

### VTS-TSRs and DISP=OLD for tableBASE Libraries

Although there are no library DD statements required on the jobs that launch a VTS-TSR, these library DD statements are on the jobs that use the VTS-TSR. Normally, DISP=SHR is the appropriate setting for tableBASE libraries so that enqueues will properly protect open tables in the VTS-TSR.

You can use DISP=OLD with temporary datasets.

You can also use DISP=OLD when tables are only ever opened for read in the VTS-TSR and any newly defined tables will never be stored to libraries. In this case, DISP=OLD may enhance performance since we will not ENQ the library when reading tables. Also, DISP=OLD will prevent other applications from updating the library, which may be a desirable condition.

### z/OS 1.5 with Enhanced Data Integrity

With the release of z/OS 1.5, an option called "Enhanced Data Integrity" can be enabled in SYS1.PARMLIB. This option is designed to protect physical sequential (DSORG=PS) datasets allocated with DISP=SHR from being concurrently opened for update by multiple users.

This can affect tableBASE users, since tableBASE libraries can be allocated with DSORG=PS (even though we only document BDAM (DSORG=DA) and VSAM). tableBASE internally protects tableBASE libraries when allocated with DISP=SHR, so "Enhanced Data Integrity" should not be enabled for tableBASE libraries.

The system programmer can bypass this option for specific datasets. Ensure that tableBASE libraries allocated with DSORG=PS are listed in the exclude list in the IFGPSEDI member.

### Specify library as read-only

tableBASE supports customers who wish to prevent updates to a tableBASE library. In batch and IMS, customers can use LABEL=(,,IN) on the DD statement. In CICS,

customers can use the RDO statements READ(YES), ADD(NO), and/or UPDATE(NO). If an application attempts to update a read-only library, tableBASE will return error code 61 subcode 11 "Operation to write to a Library was terminated. Library read-only" or the calling task abends, depending on the abend switch setting.

## Library expansion

Any library may be expanded in Version 6. Previous tableBASE versions had limitations that prevented a library from being expanded beyond certain boundary points, depending on the original size of the library. After expansion in Version 6, libraries cannot be converted to previous releases.

**Note:** Library expansion must be done using the corresponding version of TBEXEC. For example if a Version 6 library must be expanded, use the TBEXEC provided with tableBASE 6.0.2.

## Library directory caching

The tableBASE library directory can now be cached in memory. This can dramatically reduce the time needed to open a large number of tables when a TSR is initially loaded.

**Note:** In an in-house test, load time for 40,000 tables from a tableBASE library into a local TSR was reduced from 11 hours to 7 minutes. This was an extreme stress test; your results may vary.

Caching for a library is controlled by coding OPTCD=C on the DD statement for a library. This feature is not supported in a CICS region.

The initial loading of a TSR may benefit from the use of the library directory caching feature, and subsequent refreshes may also benefit if more than one table from the library is being refreshed. However, if updates are being done to the library while the loading or refreshing is being performed, these benefits will be lost. It is also important to note that the library directory caching feature is enabled only at region initialization.

**Note:** Caching has significant benefit when a library is accessed heavily by a region and other regions are not frequently updating the library. However, caching can have negative performance consequences when multiple regions are frequently updating the library directory.

## User-selectable blocksize for tableBASE libraries

Each library can have an optimal blocksize selected at the time it is created.

tableBASE Version 6 features user-selectable blocksizes. Previously, blocksize was fixed at 3120 bytes. Now you can optimize the blocksize for your tables up to the size of a track

(or 32,760 bytes, whichever is smaller). This increased flexibility allows you to decrease the time to load tables from the library to a TSR since fewer I/Os are required.

**Note:** Consider putting all of your large tables together in one library and increasing the blocksize.

Changing the blocksize of a library that has been converted from Version 5 format to 6 prohibits it from being converted back to a Version 5 library.

Libraries created in Version 6 can have any blocksize from 800 to 32760.

## Description of tableBASE library versions

With the introduction of tableBASE Version 6 and Library Bridge, there are now a number of types of tableBASE libraries. The attributes of each are highlighted below:

### Version 5 libraries

- Created with tableBASE Version 5 (including Library Bridge tableBASE 5.B)
- Can be converted to Bridge or Version 6 libraries
- Can be accessed by tableBASE Version 5.x and Library Bridge
- 3120 blocksize, Version 5 directory structure, indexes are stored on the library, hash tables may be true or indexed, paged tables supported

### Bridge libraries

- Converted from Version 5 libraries via the Library Bridge conversion utility
- Can be converted back to Version 5 libraries
- Can be converted to Version 6 libraries
- Can be accessed by Library Bridge, and tableBASE Version 6.0.2 (or later)
- 3120 blocksize, Version 6 directory structure, indexes are stored on the library, no paged tables
- Can *NOT* be accessed by any tableBASE 5.x releases prior to Release 5.B
- Library expansion is possible but is limited by the same restrictions as Version 5

**Note:** All releases of tableBASE 6.0.x will continue to provide support for Bridge libraries. However, DKL cannot guarantee support in future releases of tableBASE beyond 6.0.x (e.g. tableBASE 6.1.x). Any modifications or improvements required to the libraries will only be made to the Version 6 libraries.

### Version 6 transition libraries

- Created originally by converting Version 5 tableBASE libraries using the job streams CVLB526B and CVLB526V delivered with tableBASE 6.0.1 that are now obsolete, see note below
- Converted from Version 5 libraries using the conversion utility DK1TCNV
- Can be converted back to Version 5 libraries if they have not been expanded
- 3120 Block Size, Version 6 directory structure, indexes are stored on the library, no paged tables
- Can only be accessed by tableBASE release 6.0.x (or later)

Version 6 Transition Libraries must be converted using the DK1TCNV utility to Version 6 libraries, or to BRIDGE libraries if so desired. The library and can be identified with the DK1TLCHK utility. For more information see the *tableBASE Installation Guide, Release 6.0.3* Appendix "Converting tableBASE libraries".

### Version 6 libraries

- Defined with tableBASE Version 6 or are converted from Version 5, Bridge, or Version 6 Transition libraries with the conversion utility DK1TCNV.
- Cannot be converted back to any other type of library
- Can only be accessed by tableBASE Version 6
- No fixed Block Size (This feature does not apply to libraries converted from previous library versions to the Version 6 format.)
- Version 6 directory structure, no paged tables
- Provide enhanced performance unique to Version 6 libraries:
  - Tables have dynamic indexes that are build at open time, instead of having to read indexes from and write indexes to the library.
  - Libraries that are dedicated to a region may cache their directory information, eliminating many library accesses.
  - Libraries can be defined with BLOCKSIZEs optimized for your table characteristics and DASD devices.

**Table 7-1: Compatibility of library versions with tableBASE code**

<b>tableBASE version/release</b>	<b>Libraries it can access</b>
tableBASE 5.x	V5
Library Bridge (tableBASE 5.B)	V5, V5 Bridge
tableBASE 6.0.2	V5 Bridge, V6Trans, V6

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

## Converting a tableBASE library

**Note:** tableBASE Version 6 libraries are not compatible with libraries from previous releases.

Converting libraries to the Version 6 format is an essential part of the migration to tableBASE 6.0.2. This can be done in one of two ways: use the utilities provided with tableBASE 6.0.2 to convert your libraries, or use the utilities provided along with the Library Bridge product.

**Note:** Please backup your libraries before proceeding with any conversion process.

### Conversion utilities

tableBASE 6.0.2 comes with utilities to allow you to convert your tableBASE libraries to the Version 6 format. Once your libraries are converted, and you've set up your environment (CICS, batch, IMS, etc.) to use tableBASE 6.0.2, you're ready to go.

### Library Bridge

The main advantage of using the Library Bridge is to allow an extended conversion period as both Library Bridge and tableBASE 6.0.2 (and later) are able to read Bridge libraries.

Install Library Bridge and convert your Version 5 libraries to Bridge libraries. Bridge libraries can be read by both Library Bridge (tableBASE 5.B) and tableBASE 6.x. Install and set up your environment (CICS, batch, IMS, etc.) to use tableBASE 6.0.2 (or later). Gradually convert Bridge libraries to the Version 6 format, and you're done.

The Library Bridge can only be used with tableBASE 6.0.2 (or later), and is a leased product. Data Kinetics will lease the Library Bridge free of charge to our maintenance customers for a period of 9 months. After this period expires you may continue to lease this product at a monthly rate. For more details visit our web site or contact Customer Support.

**Note:** The conversion utilities will also work for Version 4 libraries.

For information on converting your library from Version 5 to 6, see the *tableBASE Installation Guide, Release 6.0.3* Appendix "Converting tableBASE libraries".

For information on using the Library Bridge to migrate your libraries, please consult the *tableBASE Library Bridge Manual, Release 5.B*.

## Conversion restrictions

Once a library has been converted to the Version 6 format it cannot be converted back. However, it is possible to convert Library Bridge and Version 6 transition libraries back to the Version 5 format provided that the V6TRANS library has not been expanded.

## CICS

Major improvements to CICS processing are delivered by Version 6. For example, now that the local TSR and VTS-TSRs are in Data Spaces, CICS users will notice an increase in the amount of available memory in CICS regions.

tableBASE Version 6 is compatible with CICS/TS 1.3, 2.1, 2.2, 2.3 and 3.1.

### tableBASE restart

Version 6 CICS allows you to shut down tableBASE in a CICS region and subsequently restart it. When it restarts, it automatically forces new copies of the tableBASE modules to be loaded. This allows you to reconfigure or change tableBASE without restarting a production CICS region. It also allows you to back out load modules and go back to a previous level of tableBASE Version 6.

Use the transaction `TBST TERM` to terminate tableBASE.

Use the transaction `TBST INIT` to restart tableBASE.

### TBASEV/TBCALLV and CICS storage protection

Previous versions of tableBASE could be called via the TBASEV and TBCALLV stubs. These stubs were not re-entrant and could not be loaded into key-0 memory. They were loaded into key-8 memory, which caused problems for tasks running in UserKey (9) in a STGPROT environment. So any application calling them could not use STGPROT=YES. tableBASE Version 6 provides a re-entrant stub named TBLBASE. TBLBASE has aliases TBASEV and TBCALLV. You can re-link existing applications to use this new stub and then the applications can be marked STGPROT=YES. See note in [Chapter 2 “Unified tableBASE stub”](#) on page 8.

### CICS userkey/CICSkey

Version 6 of tableBASE honours the CICS Userkey/CICSkey conventions. Most of tableBASE executes in the designated EXEC key, thus providing the protection intended by the CICS multi-key feature.

## Waiting CICS tasks

Version 6 of tableBASE uses several strategies to handle CICS tasks that are allowed to wait while opening a table for write that is held exclusively by another region.

In previous releases, a simple delay-and-retry loop was used to handle this type of CICS task. This approach often resulted in degraded performance because the CICS region, running at a higher dispatch priority than batch, looped too fast to allow a batch job to complete its work on a held table so that the table could be released for the CICS users.

If the tableBASE PC Server is installed at the site, a very efficient algorithm is used that lets the CICS task wait for the availability of the held table without impacting the CICS QR (quasi re-entrant) task.

**Note:** The Program Call Server (PC Server) is new in Version 6. The PC Server provides the functions needed for tableBASE multitasking, accessing a VTS-TSR and is compatible with DB2 stored procedures.

If the tableBASE PC Server is not installed, then a second algorithm is used. This algorithm uses a proxy subtask to wait for the table to be released and the CICS task is suspended—again, with no effect on the QR task—until its proxy subtask notifies it that the table is available.

For more information on the PC Server see the *tableBASE Administration Guide, Release 6.0.3*.

## VTS.LOAD

In previous releases of tableBASE, TBCALLV and TBASEV (along with TBROOTV and TBNUCLV) were MVS-loaded into memory. For this reason, they had to be in a library that could be concatenated to the CICS region's STEPLIB. The special APF-authorized library VTS.LOAD was provided for this purpose.

In Version 6, all tableBASE modules required in a CICS region are CICS-loaded from the RPL concatenation. You no longer need the STEPLIB concatenation.



## tableBASE VTS

### Read/write VTS-TSR

Version 6 introduces update capabilities to the VTS-TSR. This allows the complete removal of any tableBASE transaction affinity restrictions.

Tables opened for write in a VTS-TSR can be accessed and modified at the same time by multiple applications running in all tableBASE supported environments, such as batch, CICS, and IMS. The rule that ensures that a table can be opened for write in only one TSR (local or shared) at a time remains unchanged in this release.

In prior releases, VTS commands that attempted to update the VTS-TSR would have received a non-zero error code. With read and write now available in the VTS, those commands, like DK, IK, RK, DC, IC, RC, OW, and MT, among others, will now complete successfully.

Installation of the PC Server is required for access to a VTS-TSR. The PC Server is new in Version 6. It provides the functions needed for tableBASE multitasking, accessing a VTS-TSR, and is compatible with DB2 stored procedures. For more information please see the *tableBASE Administration Guide, Release 6.0.3*.

**Note:** Version 6 of the VTS-TSR cannot be accessed by previous releases of tableBASE, nor can tableBASE Version 6 access previous versions of a VTS-TSR.

### Considerations for moving to read/write VTS-TSR

Significant reduction of virtual storage utilization is possible when accessing tables from a VTS-TSR. If your installation uses multiple IMS regions or multiple CICS regions that each require access to the same tables, without VTS the tables are replicated in the TSR local to each region. Since each IMS or CICS region can access tables from a common VTS-TSR, the storage required for local TSRs can be greatly reduced or eliminated altogether. If a local TSR is not required after moving all the tables into a VTS-TSR, the local TSR size can be reduced to the absolute minimum by setting the `TSRSIZE=0`. For more information, see the `TSRSIZE` parameter description in Appendix A of the *tableBASE Installation Guide, Release 6.0.3*.

If multiple users could be updating a table—whether online, in a VTS-TSR or other multitasking environment—be aware that a `LOCK-LATCH` is available to restrict updates to authorized users/applications. It is not required. See [Chapter 2 “Multiuser \(LOCK-LATCH\)”](#) on page 8.

In a read/write VTS environment, it is important to ensure all tableBASE jobs operate at the same WLM priority level. This is to insure that a low priority job does not lock out a high priority job. This can occur when the low priority job is updating a tableBASE table and is interrupted by MVS.

## Linked tables

A linked table is created when a user issues a command to open a table, and during the LIB-LIST search, the table is found to be already open in a VTS-TSR. Since the table is already open in a VTS-TSR that was part of the LIB-LIST, a link is created to that table. A linked table is also known as a remote table.

**Note:** The LIB-LIST provides the list and order of libraries to be searched when tables are opened. A VTS-TSR can be added to this list and treated as if it was a library using the ML command or the VTSFIRST/VTSLAST execution-time parameters of TBOPT.

In Version 6, a linked table is always treated as opened for read, even if the table in the VTS-TSR is opened for write. This will ensure backwards compatibility with Release 5.x when the VTS-TSR was read-only.

An attempt to open for write a linked table will fail with an error code 13 subcode 5.

A linked table is identified by the use of an LT or GD command.

By setting the VTSNAME in the TBPARM, you can avoid linking the table and access the table directly. For more information, see [“Linked tables and TB-PARM”](#) on page 32.

## tablesONLINE

### tablesONLINE/CICS

#### New in Version 6

New features were added to tablesONLINE/CICS Version 6 to:

1. track which user has created or last modified a row. This is accomplished by defining an eight byte field within the row to receive the logged-on USER ID. For more information, consult "Define Fields in a View" in the *tablesONLINE/CICS User's Guide, Release 6.0.3*. When a table is being edited, any row that is created or modified will have the USER ID placed into this field.
2. enforce unique primary keys with an Alternate View
3. define the initial value, upper bound, and lower bound of a field
4. control simultaneous updates by multiple users to a single table

While using Alternate Views, there has always been the risk of having duplicate primary keys created when the Data Table is edited with the Alternate View. Now, the primary keys of the Data Table can be incorporated into the design of the Alternate Index View and indicate that you wish to enforce these primary keys in your Alternate View. Taking these steps prevents duplicate primary keys from being created. The default is still to allow duplicate keys. For more information, consult "Enforcing Unique Primary Keys with an Alternate Index" in the *tablesONLINE/CICS User's Guide, Release 6.0.3*.

Before 6.0, only Action Codes could be used to restrict the range of the field input. This was often difficult and cumbersome if you only wanted to set the upper and lower bound. In Version 6 of tablesONLINE, you can set the initial value of a field as well as an upper and lower value. The initial value sets the default input of the field when the row is edited. The upper and lower values are used together to restrict the field input. For more information, consult "Define Fields in a View" in the *tablesONLINE/CICS User's Guide, Release 6.0.3*.

Multiple users can edit the same Data Table at the same time. Version 6 allows you to set options to allow multiple users to work in a View and/or and Alternate View on the same Data Table—warnings are provided to indicate to users that multiple user access is in effect.

Duplicate primary keys are not allowed when multiple users can edit a table. Enforcing unique primary keys in a multiple user environment ensures that all of the data can be properly referenced.

tablesONLINE/CICS ensures integrity in this new multiple-user environment. tablesONLINE tracks all of the users accessing a Data Table simultaneously and prevents users from editing the same row at the same time.

If you attempt to access a row that is being edited by another user, a message informs you that the row is already being edited, and also identifies the user and the application.

Although you won't be able to edit the row while another user is editing it, the row is always accessible to browse. For more information, consult "Multiple User Update" in the *tablesONLINE/CICS User's Guide, Release 6.0.3*.

There have also been improvements to the *tablesONLINE/CICS User's Guide, Release 6.0.3*. The section on Description Tables has been revised and expanded to provide detailed information on how to customize tablesONLINE.

## Exits

In Release 6.0, in order to implement the browse integrity enhancement, the offset of the variable T-LCMD-NO-OF-ITEMS has been changed by 4 bytes. If your exits use this variable, you will need to recompile. Otherwise, no changes are needed. The change is flagged by F00017 in the margin of the EXITPARM copybook.

**Note:** TBOL/CICS requires that working storage be initialized to binary zeroes. Under LE, this is accomplished by ensuring that the run-time storage option for the CICS region is STORAGE=(00,...). Refer to IBM's LE Customization manual for guidance in creating a CEEUOPT or CEEROPT module.

## tablesONLINE/ISPF

There have been no changes to the functionality of TBOL/ISPF with this release. The tableBASE driver (TBDRIVER) that you access through TBOL/ISPF is the Version 6 driver.

## Notes and cautions

### Notes

#### Table access optimization

Table access can be optimized by using a separate command area for each table. Instead of having to search a TSR directory each time to determine where the table resides in memory, tableBASE uses a reserved field in the command area (table handle) pointing to the position of the table in the TSR. By keeping a separate command area for each table accessed, this table handle is used to quickly locate a previously opened table.

#### High-speed processing without EDF

When using EDF to trace an application in CICS, all calls to any tableBASE entry point will be traced using the tableBASE CICS Resource Manager. If you stop using EDF with the transaction, tableBASE will bypass the Resource Manager and automatically revert to high-speed processing (TB-Turbo option in TB-PARM).

#### Date-Sensitive Processing

The default date used by Date-Sensitive Processing rolls over at midnight. This will prove useful for 24x7 applications that rely on date-sensitive processing.

#### Access Register mode

Version 6 has the following MVS restrictions applicable to AR mode (this applies only to applications that have been coded in Assembler):

1. tableBASE must be called in a primary mode. Failure to do this may result in unpredictable results.

2. tableBASE zeroes all access registers (A0-A15) on return to a caller. It is a responsibility of the caller to save and restore its own access registers.

### **VSAM LSR pools for tableBASE libraries**

tableBASE must be allowed to control its own I/O activity to ensure the integrity of library directories. If a VSAM library has been associated to an LSR pool, when the library is opened by tableBASE and found to be pooled, it is closed by tableBASE, removed from the pool, and reopened. The exception to this is if the library happens to have DISP=OLD, then it will be left in the LSR pool.

**Note:** You risk destroying your data if you are using a tool that dynamically modifies the VSAM LSR setting.

When a tableBASE library is set to DISP=SHR, there is an exposure when using third party software like MBO to dynamically turn on LSR processing (or mechanisms equivalent in effect). In these cases, tableBASE is not notified that LSR is turned on and cannot remove the library from the LSR pool. If more than one region can update the library, the library will be corrupted.

The library directory caching feature of Version 6 should provide equivalent or better results than LSR pooling.

### **Considerations for working with large tables**

With Version 6, there are new considerations for dealing with large tables in tableBASE. Both the local TSR and VTS-TSRs are now in Data Spaces. The maximum size of a Data Space is 2 G.

In addition you should consider the region size, the number of tasks running in the region, processing sequence, and the frequency and type of access activity. The choice of one search strategy over another can impact system performance.

What is large? For the purposes of the following recommendations, a table is considered very large when the total size is 50 M or larger, or the number of records is greater than one million. The maximum row size a table can have is 32 K, which is the maximum positive value in a half-word.

#### **Large table recommendations**

Binary or hash search methods are recommended for tables with more than 300-400 rows.

High frequency of inserts with low expansion factor are not recommended.

**Note:** With the introduction of Version 6, all tables are now treated as Pointer tables and have an Index, even if they are listed as being True tables. The change is

transparent to the user, but Pointer and True tables are now treated internally as Pointer tables.

### **Table Space Region (TSR)**

The TSR is in a data space in Release 6.0. Rollouts of tables using TBTSLIB are no longer supported with this release. The data space must be large enough to hold all tables that were paged tables in prior releases.

### **Binary searches**

With Version 6, all searches cache the first 4 bytes of the key. This greatly improves search performance. With a large table where these 4 bytes may be the same throughout all rows of the tables these benefits may not be realized. Better results can be achieved by using a key that varies within the first 4 bytes.

### **Open/store activity**

The size of the table determines the load on the I/O system, possibly causing slowdowns in an online environment.

### **Creation method**

For a keyed table, creation in Sequential order can be much more efficient than in Random order.

If the table is already in Random order, consider loading the Data Table and then changing the definition to Sequential.

### **Expansion factor**

This controls the amount of additional memory that will be obtained to expand non-Hash tables. It should be set sufficiently large so all insertions between opens of the table will fit. For example, if the table has 100,000 rows, and a maximum of 10,000 rows are inserted with no deletions when it is opened, an expansion factor of 100 (10%) is sufficient.

### **tableBASE libraries**

A tableBASE library will never use secondary allocated space because it can never increase in size dynamically. It will only use the initially allocated space. This initial space allocation does not need to be contiguous.

### **Linked tables and TB-PARM**

If every tableBASE call in your applications is to VTS-TSR, then using the VTSNAME in the TB-SUBSYSTEM field in the TB-PARM to access VTS-TSR is more efficient than using the VTSFIRST, VTSLAST, or the ML command. Use of the VTSNAME in the TBPARM completely bypasses the use of the local TSR.

If you are using VTSFIRST, VTSLAST, or ML to add a VTS-TSR to the LIB-LIST, the transaction process will first go to the local TSR (through CICS, IMS or batch) to find a handle (pivot entry) for the table requested, and if it is the first request for that table, a dummy entry will be entered in the local TSR. The process then goes to the VTS-TSR to locate the table entry.

The table entry address in the VTS-TSR would then be used to update the initial dummy entry in the local TSR.

On subsequent passes, the local TSR would have a valid address as a Pointer to the VTS-TSR for that table. This process describes a linked table. Also see “[Linked tables](#)” on page 26).

**Note:** Use of VTSFIRST, VTSLAST, or the ML-list to access a table in VTS-TSR (linked table) does not allow for update commands against the table. Updates of the table can only be achieved by the use of the VTSNAME in the TB-PARM to access the table.

## Using the TB-PARM Communication Area

Data Kinetics recommends using a TB-PARM Communication Area on every call to TBLBASE. If a TB-PARM is not used, performance can be significantly affected in some instances—for example, when DK1TCALL or DK1TROTb is in LPA or an authorized library. To circumvent this, relink these two modules as serially reusable. Contact Data Kinetics Technical Support at 1-613-523-5588.

## Application performance with TBASEV or TBCALLV

To improve the performance of applications using the V5 TBASEV or TBCALLV interface, relink the application modules with the Version 6 TBASEV or TBCALLV (now aliases for DK1TCALL).

## XML and tableBASE

The ability to export COBOL copybooks can be used to assist in creating XML from your tableBASE data and in moving tableBASE data out of tableBASE using third party tools. For more details, please see the *tableBASE Batch Utilities Guide, Release 6.0.3*.

## Unique DDNAMEs

Version 6 requires each tableBASE library to be identified by a unique DDNAME. You may not associate more than one DDNAME with the same tableBASE library.

Version 6 normally requires each tableBASE library to be identified by a unique DDNAME. You may not associate more than one DDNAME with the same tableBASE library. The exception is when using the AL and UL commands in an IMS or single-tasking batch environment and the different DDNAMEs are not allocated at the same

time. See the *tableBASE Programming Guide, Release 6.0.3*, “Allocate (AL)” and “Un-allocate (UL)” for more information.

### **tableBASE and CICS threadsafe**

The tableBASE CICS interface is delivered with most tableBASE programs defined as CONCURRENCY(THREADSAFE). At this release level of tableBASE, CICS threadsafe applications are restricted to accessing (but not updating or opening) tableBASE tables only in a VTS-TSR.

An update is being developed to fully support CICS threadsafe applications.

Currently, to avoid problems, ensure that all applications that access tableBASE are defined as CONCURRENCY(QUASIRENT). Because the tableBASE API is an MVS call (rather than an EXEC CICS LINK or XCTL), redefining the tableBASE programs as CONCURRENCY(QUASIRENT) is not necessary.

### **Installation of data dictionary library**

In prior releases the installation of the data dictionary library (\*YOUR.PREFIX\*.TBASE.TBDICLB) was described as optional in the members DFHFCTOL and TBOL60V in \*YOUR.PREFIX\*.TBASE.SRC.

This installation instruction was inconsistent with the rest of the product documentation. It is now recommended that it is installed and the option to use the data dictionary rests with the applications development teams.

## **Cautions**

### **tableBASE and BMC product 'ENERGIZER for CICS' versions 4.4**

When using the BMC product 'ENERGIZER for CICS' versions 4.4 in a CICS environment, tableBASE transactions appear to stop. Transactions going to tableBASE appear to enter tableBASE but do not appear to exit tableBASE. High CPU usage is also observed.

#### ***Solution***

You can either disable and not use the BMC product or contact BMC to obtain the software fix for versions 4.4 that are below PUT04041B.

### **Alternate Index problem in Version 5**

In previous releases, it was possible to create an Alternate Index with the same name as the Data Table (even though this table was consequently not usable). However, in



Version 6 you are no longer able to create an Alternate Index with the same name as the Data Table. An attempt to do so now results in error code 19.

This problem may occur in Version 6, if a Version 5 table with such a loop is imported into Release 6.0.1.

### **Solution**

Ensure that tableBASE libraries in Version 5 format that are to be converted to Version 6, do not contain Alternate Indexes that have the same name as the Data Table.

### **Error importing library**

An invalid true hash table in Version 5 (such as one that has more populated rows than the row count value in the table definition indicates) will cause errors during the import process into a Version 6 release of tableBASE.

### **Solution**

The TBEXEC EXPORT command is modified in Release 6.0.1 to return an error if a hash table has more rows than specified in the definition.

**Note:** Prior versions of TBEXEC will still incorrectly export an invalid hash table. The Version 5 library can be converted directly to Release 6 using the library conversion utility provided. A corrected version of TBEXEC for 5.1.0 is now available by request.

### **Enqueue missing for table opened for write**

tableBASE issues an MVS enqueue for any table open for write in a TSR if the tableBASE library is allocated as DISP=SHR. If a tableBASE library is allocated as DISP=OLD the enqueue is not obtained.

If the table is opened in a VTS-TSR the table will not be protected by the enqueue once the job that opened the table has terminated. This could result in a table being opened for write in multiple TSRs.

### **Solution**

If a job opens tables for write in a VTS-TSR ensure that the job allocates the tableBASE library as DISP=SHR.

### **Table remains locked in a TSR**

If a job that accesses tableBASE is cancelled or abnormally terminates during tableBASE processing, a table may remain locked in the TSR. The subsequent attempts to access the table may result in loops.

**Solution**

Recycle the TSR (local region for a local TSR or the VTSAGENT for a VST-TSR).

**Customer anchor table DKL slot invalid**

When the PCSERVER initializes, and attempts to create or use an internal anchor, the PCSERVER sometimes erroneously puts out a message:  
DK100844W Customer Anchor Table DKL slot invalid.

**Solution**

Please ignore message as it is for diagnosis only and, not for tableBASE operation.

**Lock out of batch processing**

The enabling of Implicit Open can cause CICS transactions to lock out batch processing if the transaction is accessing a table in the VTS that is not open and no tableBASE library for that region exists.

**Solution**

When no tableBASE library for the region exists, turn Implicit Open off.

**Batch/IMS Performance Degradation**

If tableBASE modules are loaded from an authorized library or LPA when a newly tested application is moved into production after being tested using tableBASE STEPLIBs, the application may experience a performance degradation.

The performance testing of an application should occur in the same environment as the production environment.

**Solution**

Use a TBPARM parameter with all calls to tableBASE if batch, IMS and DB2 stored procedures accessing tableBASE use excessive CPU time when tableBASE modules are loaded from an authorized library or LPA.

See Best practices, “Batch/IMS Performance Degradation”, in the *tableBASE Administration Guide, Release 6.0.3*.

## Documentation updates

This section lists additions to the DKL documentation that have recently been added. It is listed here, as well as in the appropriate guides, to assist you in identifying new information.

### Access to VTS-TSRs

In prior releases VTS-TSRs users were allowed only read access (updates were supported only through the refresh process run under the VTS address space). Four methods of access were supported:

1. using the TBOPT data set VTSNAME= parameter to specify the VTS-TSR and invoking tableBASE with the TBCALLV or TBASEV API.
2. using the VS command to specify the VTS-TSR and invoking tableBASE with any API.
3. specifying the VTS-TSR in the TBPARM data area passed as the first parameter on the call to tableBASE.
4. using the TBOPT data set VTSFIRST= or VTSLAST= parameters to specify a VTS-TSR and/or specifying a VTS name in the LIB-LIST of the ML command.

With Version 6 users are allowed read/write access to VTS-TSRs. Methods 1 through 3 (above) are supported as read/write interfaces—with Method 3 as the recommended update interface. Method 4 is supported as a read-only interface as in prior releases.

Commands which access the VTS-TSR and also access a tableBASE library will locate the library based on the LIB-LIST in effect in the local address space (not the VTSAGENT address space which created the VTS-TSR).

The refresh process is no longer run under the VTS address space. The refresh process can be emulated in Version 6 by using the TBDRIVER utility. The SELECT SUBSYSTEM (SS) command is used to specify the VTS-TSR for subsequent commands. The SS command is a TBDRIVER/TBDRIVC command; it is not a tableBASE command. The ReFresh (RF) command is now a tableBASE command and can be issued from any application, not just TBDRIVER.

### Determining TSR size

In prior versions of tableBASE the area for tables was acquired from local virtual memory. (Only in VTS-TSR was a dataspace acquired.) If a TBOPT parameter specified TSRSIZE= then the table space was acquired in a single GETMAIN request. If it was not then GETMAIN requests were made for every table as it was opened. If TSR size was

specified then a tableBASE rollout library allocated with DDNAME TBTSLIB was also required to provide overflow space for the TSR.

With Version 6 the area for tables is always provided by acquiring a data space. This means a TSRSIZE= is always used, either from the default setup at installation or the TBOPT data set for this job. For more information, see Appendix A in the *tableBASE Installation Guide, Release 6.0.3*.

## The CD command

<b>Return value</b>	Error Code 92—In the event of receiving an Error Code 92 create more space in the VTS-TSR or local TSR. Retry the CD command.
<b>Notes</b>	In the event of an ABEND when using the CD command, review the table to confirm data integrity.

## TBEXEC: the UPDATE command description has changed

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

In previous versions the following note was in error:

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

This has been changed as follows:

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