

---

TurboIMAGE/XL Database Management System  
**DBChange Plus User's Guide**



HP Part No. 36386-90001  
Printed in U.S.A. 19900101

First Edition  
E1290  
DRAFT 11/6/97 20:05

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company.

**Copyright © 1987, 1988, 1989, 1990 by Hewlett-Packard Company**

---

## Print History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

<b>Edition</b>	<b>Date</b>	<b>Software Version</b>
First Edition	December 1990	36386- A.00.00

---

## Preface

Because DBChange Plus is designed to be used primarily by database creators, this guide is written for a user with both a thorough knowledge of the HP 3000 and the TurboIMAGE/XL database management system.

---

## How to Use This Guide . . .

This guide has two purposes:

- If you are a new DBChange Plus user, this guide is designed to acquaint you with DBChange Plus and to help you learn to use DBChange Plus successfully.
- Once you are an experienced DBChange Plus user, this guide provides quick access to information about DBChange Plus commands.

The format of this guide reflects this dual purpose. Whether you are a new or experienced DBChange Plus user, you need to understand this two-part format to make effective use of the guide. The guide contains the following information:

- **Task Reference**—For new users, chapters 1 through 9 explain how to perform database restructuring tasks. Each chapter begins with an overview of the commands used for each task, followed by step-by-step instructions for each task. Each chapter uses examples to enhance the discussion of each task.
- **Command Reference**—For experienced users, chapter 10 provides quick access to the DBChange Plus commands.

You can use either the table of contents or the index to locate information.

Here's what you'll find in this guide:

### **Chapter 1. Introduction to DBChange Plus**

This chapter introduces you to DBChange Plus. Here you'll become familiar with some of the basic features of DBChange Plus, and you'll learn to do a few simple tasks.

### **Chapter 2. Copying a Database**

This chapter describes how to copy a database.

### **Chapter 3. Displaying Information about a Database**

This chapter describes how to print the schema for a TurboIMAGE/XL database. It also explains how to display information about the database structure.

### **Chapter 4. Changing Data Items**

This chapter describes how to change data items in a TurboIMAGE/XL database.

## **Chapter 5. Changing Data Sets**

This chapter describes how to change data sets in a TurboIMAGE/XL database. It describes the steps involved in changing data set fields and data set characteristics, such as capacity and blocking factor. This chapter also explains how to recover a data set.

## **Chapter 6. Changing Paths in a Database**

This chapter describes how to make changes to paths in a TurboIMAGE/XL database.

## **Chapter 7. Changing Database Security**

This chapter describes how to change user access to a TurboIMAGE/XL database.

## **Chapter 8. Maintaining the Database**

This chapter describes how to maintain a TurboIMAGE/XL database. Checking for structural integrity, fixing structural problems, improving database performance, and managing database capacity are discussed.

## **Chapter 9. Applying Changes to the Database Structure**

This chapter describes how to use the DBAPLUS program to actually restructure the database. Note that this chapter signals the end of the task reference portion of this guide.

## **Chapter 10. DBChange Plus Commands**

This chapter is a quick reference for experienced DBChange Plus users. All the DBChange Plus commands are presented here in alphabetical order. The syntax and parameters are defined for each command, followed by a description and examples.

## **Appendix A. DBChange Plus Messages**

This appendix lists the messages you may receive when using DBChange Plus. The meaning of each error and the action to take to correct problem are provided.

## **Appendix B. ORDERS Schema**

This appendix shows the ORDERS database schema listing. The ORDERS database is the sample database used for examples throughout this guide.

## **Index**

---

## Other Information Sources

### Manuals

In addition to this guide, you may need to consult the following manuals:

<b>Title</b>	<b>Part Number</b>
MPE XL Intrinsic Reference Manual	32650-90028
TurboIMAGE/XL Database Management System Reference Manual	30391-90001

### Training

The following training courses are available:

<b>Title</b>	<b>Course Number</b>
TurboIMAGE DBMS/3000	35053B
Advanced TurboIMAGE	51480A

---

## Conventions

### UPPERCASE

In a syntax statement, commands and keywords are shown in uppercase characters. The characters must be entered in the order shown; however, you can enter the characters in either uppercase or lowercase. For example:

COMMAND

can be entered as any of the following:

command          Command          COMMAND

### *italics*

In a syntax statement or an example, a word in italics represents a parameter or argument that you must replace with the actual value. In the following example, you must replace *FileName* with the name of the file:

COMMAND *FileName*

### punctuation

In a syntax statement, punctuation characters (other than brackets, braces, vertical bars, and ellipses) must be entered exactly as shown. In the following example, the parentheses and colon must be entered:

( *FileName* ) : ( *FileName* )

### underlining

Within an example that contains interactive dialog, user input and user responses to prompts are indicated by underlining. In the following example, yes is the user's response to the prompt:

Do you want to continue? >> yes

### { }

In a syntax statement, braces enclose required elements. When several elements are stacked within braces, you must select one. In the following example, you must select either ON or OFF:

COMMAND { ON }  
          { OFF }

### [ ]

In a syntax statement, brackets enclose optional elements. In the following example, OPTION can be omitted:

COMMAND *FileName* [OPTION]

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can

select `OPTION` or *Parameter* or neither. The elements cannot be repeated.

`COMMAND` *FileName*  $\left[ \begin{array}{l} \text{OPTION} \\ \textit{Parameter} \end{array} \right]$

---

## Conventions (continued)

[ ... ]

In a syntax statement, horizontal ellipses enclosed in brackets indicate that you can repeatedly select the element(s) that appear within the immediately preceding pair of brackets or braces. In the example below, you can select *Parameter* zero or more times. Each instance of *Parameter* must be preceded by a comma:

[, *Parameter*] [...]

In the example below, you only use the comma as a delimiter if *Parameter* is repeated; no comma is used before the first occurrence of *Parameter*:

[ *Parameter*] [, ...]

| ... |

In a syntax statement, horizontal ellipses enclosed in vertical bars indicate that you can select more than one element within the immediately preceding pair of brackets or braces. However, each particular element can only be selected once. In the following example, you must select A, AB, BA, or B. The elements cannot be repeated.

{ A } | ... |  
{ B }

...

In an example, horizontal or vertical ellipses indicate where portions of an example have been omitted.

In a syntax statement, the space symbol shows a required blank. In the following example, *Parameter* and *Parameter* must be separated with a blank:

( *Parameter* ) ( *Parameter* )

The symbol  indicates a key on the keyboard. For example,  represents the carriage return key.

character

character indicates a control character. For example, Y means that you press the control key and the Y key simultaneously.

# Contents

---

<b>1. Introduction to DBChange Plus</b>	
What Is DBChange Plus? . . . . .	1-1
DBChange Plus Operation . . . . .	1-2
Getting Started . . . . .	1-2
Running the DBCPLUS Program . . . . .	1-2
Getting Help . . . . .	1-3
Specifying the Database to be Restructured . . . . .	1-4
How the Change File Works . . . . .	1-4
Creating a New Change File . . . . .	1-4
Updating an Existing Change File . . . . .	1-4
Purging an Existing Change File . . . . .	1-5
Using Some DBChange Plus Commands . . . . .	1-5
Adding a Data Item . . . . .	1-5
Changing a Password . . . . .	1-5
Deleting a Field . . . . .	1-6
Leaving the DBChange Program . . . . .	1-7
Getting to Know the DBChange Plus Command	
Interface . . . . .	1-7
Command Format . . . . .	1-7
Difference Between Deferred and Immediate	
Commands . . . . .	1-8
Command Sequence . . . . .	1-8
Terminating a Command . . . . .	1-9
Executing Commands from a File . . . . .	1-9
Including Comments in a File . . . . .	1-9
Using a Command File . . . . .	1-9
Using a Job File . . . . .	1-11
Issuing MPE XL Commands from within the	
DBCPLUS Program . . . . .	1-12
Redirecting Input, Output, and the Schema Listing . . . . .	1-13
Printing the Schema to an MPE XL File . . . . .	1-13
Printing the Schema to an MPE XL File in	
Interactive Mode . . . . .	1-14
Printing the Schema to an MPE XL File in Batch	
Mode . . . . .	1-14
How DBChange Plus Handles Errors . . . . .	1-14
How DBChange Plus Helps You Avoid and Recover	
from Errors . . . . .	1-15
Using Job Control Words to Check the Status of	
DBChange Plus . . . . .	1-15

<b>3. Displaying Information about the Database</b>	
Summary of Commands to Display Information about the Database . . . . .	3-1
Printing the Schema . . . . .	3-2
Aborting the PRINT SCHEMA Command . . . . .	3-4
Using REVIEW Commands to Display Information about the Database . . . . .	3-5
Displaying User Classes and Passwords . . . . .	3-5
Displaying Data Items . . . . .	3-6
Displaying Information about Data Sets . . . . .	3-7
Displaying Data Sets . . . . .	3-7
Displaying Fields . . . . .	3-9
Displaying Paths . . . . .	3-10
<b>4. Changing Data Items</b>	
Summary of Commands to Change Data Items . . . . .	4-1
Helpful TurboIMAGE/XL Information . . . . .	4-2
Item Type Conversions . . . . .	4-3
Adding a Data Item . . . . .	4-4
Deleting a Data Item . . . . .	4-6
Recovering a Data Item . . . . .	4-9
Reordering a Data Item . . . . .	4-12
Renaming a Data Item . . . . .	4-14
Changing a Data Item . . . . .	4-15
Changing Data Item Attributes . . . . .	4-18
<b>5. Changing Data Sets</b>	
Summary of Commands to Change Data Sets . . . . .	5-1
Helpful TurboIMAGE Information . . . . .	5-3
Adding a Master Data Set . . . . .	5-4
Adding a Detail Data Set . . . . .	5-5
Deleting a Data Set . . . . .	5-6
Recovering a Data Set . . . . .	5-7
Reordering a Data Set . . . . .	5-9
Renaming a Data Set . . . . .	5-12
Changing the Capacity of a Data Set . . . . .	5-13
Changing the Blocking Factor of a Data Set . . . . .	5-15
Changing the Maximum Block Length of a Data Set . . . . .	5-17
Changing the Device Class of a Data Set . . . . .	5-20
Adding a Field to a Data Set . . . . .	5-21
Deleting a Field from a Data Set . . . . .	5-22
Reordering a Field in a Data Set . . . . .	5-24
Adding a Sort Item to a Detail Data Set . . . . .	5-25
Deleting a Sort Item from a Detail Data Set . . . . .	5-27

<b>6. Changing Paths in a Database</b>	
Summary of Commands to Change Paths . . . . .	6-1
Helpful TurboIMAGE/XL Information . . . . .	6-2
Adding a Path to a Detail Data Set . . . . .	6-2
Redefining the Primary Path in a Detail Data Set . . . . .	6-4
Deleting a Path from a Detail Data Set . . . . .	6-5
<b>7. Changing Database Security</b>	
Summary of Commands to Change Database Security . . . . .	7-1
Helpful TurboIMAGE/XL Information . . . . .	7-3
Adding a Password to the Database . . . . .	7-3
Changing a Password in the Database . . . . .	7-4
Deleting a Password from the Database . . . . .	7-6
Changing Data Item Security . . . . .	7-8
Deleting Data Item Security . . . . .	7-9
Changing Data Set Security . . . . .	7-10
Deleting Data Set Security . . . . .	7-11
<b>8. Maintaining the Database</b>	
Summary of Commands to Maintain the Database . . . . .	8-2
Guidelines for Maintaining the Database . . . . .	8-3
How the FIX Commands Operate . . . . .	8-3
Repacking Data Sets . . . . .	8-4
Checking and Fixing Paths . . . . .	8-6
Checking Paths . . . . .	8-6
Analyzing the CHECK PATH Statistics . . . . .	8-7
Fixing Paths . . . . .	8-8
Checking and Fixing Data Sets . . . . .	8-10
Checking a Master Data Set . . . . .	8-10
Analyzing the CHECK SET Statistics for a Master Data Set . . . . .	8-11
Fixing a Master Data Set . . . . .	8-12
Checking a Detail Data Set . . . . .	8-13
Analyzing the CHECK SET Statistics for a Detail Data Set . . . . .	8-14
Fixing a Detail Data Set . . . . .	8-15
Checking and Fixing the Entire Database . . . . .	8-16
Using the QUICK Option with CHECK Commands . . . . .	8-17
<b>9. Applying Changes to the Database Structure</b>	
What is DBAPLUS? . . . . .	9-1
Before Restructuring the Database . . . . .	9-1
Saving the New Schema . . . . .	9-3
Saving the Schema Processor Output . . . . .	9-3
Methods of Using the DBAPLUS Program . . . . .	9-3
Initiating the DBAPLUS Program Interactively . . . . .	9-4
Creating a DBAPLUS Job File from within the DBChange Program . . . . .	9-5
Modifying the Job File . . . . .	9-7
Running the DBAPLUS Program Independently . . . . .	9-9
Using PARM Values . . . . .	9-9

How DBAPLUS Restructures a Database . . . . .	9-10
Possible Restructuring Problems . . . . .	9-11
Restructuring the ORDERS Database . . . . .	9-12
<b>10. DBChange Plus Commands</b>	
DBChange Plus Command Formats . . . . .	10-1
Summary of DBChange Plus Commands . . . . .	10-2
ADD FIELD . . . . .	10-6
ADD ITEM . . . . .	10-8
ADD PASSWORD . . . . .	10-12
ADD PATH . . . . .	10-14
ADD SET . . . . .	10-16
ADD SORT . . . . .	10-23
BASE . . . . .	10-25
CANCEL COMMANDS . . . . .	10-27
CANCEL ERASE . . . . .	10-31
CANCEL REPACK . . . . .	10-32
CHANGE ATTRIBUTES . . . . .	10-33
CHANGE BLOCKFACTOR . . . . .	10-36
CHANGE BLOCKMAX . . . . .	10-39
CHANGE CAPACITY . . . . .	10-41
CHANGE DEVICECLASS . . . . .	10-43
CHANGE ITEM . . . . .	10-44
CHANGE ITEMSECURITY . . . . .	10-49
CHANGE PASSWORD . . . . .	10-51
CHANGE PRIMARYPATH . . . . .	10-52
CHANGE SETSECURITY . . . . .	10-53
CHECK BASE . . . . .	10-55
CHECK PATH . . . . .	10-63
CHECK SET . . . . .	10-69
CONTROL . . . . .	10-73
COPY . . . . .	10-76
DELETE FIELD . . . . .	10-78
DELETE ITEM . . . . .	10-79
DELETE ITEMSECURITY . . . . .	10-82
DELETE PASSWORD . . . . .	10-83
DELETE PATH . . . . .	10-84
DELETE SET . . . . .	10-86
DELETE SETSECURITY . . . . .	10-89
DELETE SORT . . . . .	10-90
ERASE SET . . . . .	10-91
EXIT . . . . .	10-93
FIX BASE . . . . .	10-94
FIX PATH . . . . .	10-97
FIX SET . . . . .	10-100
HELP . . . . .	10-103
OUTPUT . . . . .	10-105
PERFORM COMMANDS . . . . .	10-106
PRINT SCHEMA . . . . .	10-109
RECOVER ITEM . . . . .	10-111
RECOVER SET . . . . .	10-112

REDO . . . . .	10-113
RENAME BASE . . . . .	10-115
RENAME ITEM . . . . .	10-116
RENAME SET . . . . .	10-118
REORDER FIELD . . . . .	10-119
REORDER ITEM . . . . .	10-120
REORDER SET . . . . .	10-122
REPACK SET . . . . .	10-126
REVIEW BLOCKS . . . . .	10-132
REVIEW FIELDS . . . . .	10-134
REVIEW ITEMS . . . . .	10-136
REVIEW PASSWORDS . . . . .	10-137
REVIEW PATHS . . . . .	10-138
REVIEW SETS . . . . .	10-141
XEQ . . . . .	10-143

**A. DBChange Plus Messages**

DBChange Plus Messages (continued) . . . . .	A-19
DBChange Plus Messages (continued) . . . . .	A-35
DBChange Plus Messages (continued) . . . . .	A-57

**B. ORDERS Schema**

**Index**

## Figures

---

1-1. Typical DBChange Plus Command Format . . .	1-7
---	-----

## Tables

---

1-1. DBChange Plus Functions . . . . .	1-1
3-1. Summary of Commands to Display Information about the Database . . . . .	3-2
4-1. Summary of Commands to Change Data Items . .	4-2
4-2. DBChange Plus Supported Item Type Conversions	4-3
5-1. Summary of Commands to Change Data Sets . .	5-2
6-1. Summary of Commands to Change Paths . . . .	6-1
7-1. Summary of Commands to Change Database Security	7-2
8-1. Summary of Commands to Maintain the Database	8-2
9-1. DBAPLUS Parameters . . . . .	9-10
10-1. Summary of DBChange Plus Commands . . . .	10-3
10-2. DBChange Plus Supported Item Type Conversions	10-46
10-3. Required MPE XL Security Capabilities . . . .	10-76
10-4. DBAPLUS Parameters . . . . .	10-107

## Introduction to DBChange Plus

---

This chapter introduces DBChange Plus by explaining how it operates, by demonstrating a few tasks to get you started, and by discussing the features of the command interface. If you are a new DBChange Plus user, you should read this chapter before moving on to subsequent chapters. To learn about more DBChange Plus tasks, refer to chapters 2 through 9.

If you understand DBChange Plus tasks and only want to see the command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### What Is DBChange Plus?

DBChange Plus is a command-driven utility that allows you to perform many necessary TurboIMAGE/XL database management system tasks. DBChange Plus does a variety of changes and maintenance functions while preserving user data. DBChange Plus does not have to unload and reload the data.

Table 1-1 describes some functions of DBChange Plus.

**Table 1-1. DBChange Plus Functions**

Function	Description
Restructuring	Adding, deleting, reordering, renaming, and general management of data items, data sets, fields, sort items, and paths. Redefining primary paths.
Copying	Copying a database within a group, across groups, and across accounts.
Displaying structural information	Reviewing information about data items, data sets, fields, sort items, and paths. Printing the schema.
Security management	Adding, deleting, and changing passwords and user class access.
Maintenance and performance tuning	Capacity management. Detail data set repacking to clean up messy chains. Checking and fixing structural problems. Selective placement of data sets on specific disk classes. Smart reblocking.
Erasing	Bulk erasing of data set entries.

DBChange Plus contains two programs: DBCPLUS and DBAPLUS.

The DBCPLUS program accepts and stores your requests in a **change file**. The change file contains information about the database and also contains requests for changes as specified by the DBChange Plus commands.

The DBAPLUS program performs the actual restructuring or maintenance of the database. DBAPLUS can be invoked either from within the DBCPLUS program by issuing a PERFORM COMMANDS, or it can be invoked independently.

## DBChange Plus Operation

You can use DBChange Plus either interactively or in batch mode. When using DBChange Plus interactively, you are prompted for commands and other input, and you receive error and information messages on your terminal screen. DBChange Plus is usually used interactively for non-repetitive tasks that require immediate attention. To use DBChange Plus in batch mode, you send input to DBChange Plus from a text file called a **job file** that contains DBChange Plus and MPE XL commands. Batch mode is usually used for repetitive or time-consuming tasks.

You can also execute DBChange Plus commands from a text file called a **command file**. Command files can either be used interactively or in batch mode. Job files and command files are discussed later in this chapter under, "Executing DBChange Plus Commands from a File."

---

## Getting Started

This section gets you started with DBChange Plus by explaining the following procedures:

- Logging on to your system.
- Running the DBCPLUS program.
- Getting help.
- Specifying which database you want to restructure.
- Trying a few commonly used DBChange commands.
- Leaving the DBCPLUS program.

## Running the DBCPLUS Program

Before running DBChange Plus, first log on to the account and group where the database you want to restructure resides. You must log on to the system as the database creator. To log on, type the following:

```
:hello user.account,group
```

Once you are logged on, run the DBCPLUS program by typing the following:

```
:run dbcplus.pub.sys
```

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

TUE, MAY 22, 1990, 8:46 AM

>

The DBChange prompt (>) appears on your screen.

## Getting Help

DBChange Plus has an online help facility, which you can access anytime you are running the DBCPLUS program in interactive mode. You can either see the entire list of DBChange Plus commands, or you can get information about a specific command. To see a listing of all the DBChange Plus commands, issue the HELP command at the DBChange prompt as shown in the following example. After every 20 display lines, you are asked if you want to continue, that is, if you want more commands listed. Type either *y* for yes or *n* for no; note that the default value is displayed in uppercase. For more information, refer to "HELP" in chapter 10, "DBChange Plus Commands."

>help

```
ADD FIELD           Adds a field to a data set.
ADD ITEM            Adds a data item to a database.
ADD PASSWORD        Adds a password to a database.
ADD PATH            Adds a path to an existing detail data set.
ADD SET             Adds a set to an existing database.
ADD SORT            Adds a new sort item to an existing path.
BASE                Specifies the database to be modified.
CANCEL COMMANDS    Cancels all changes since most recent BASE.
CANCEL ERASE        Cancels a previous ERASE SET command.
CANCEL REPACK       Cancels a previous REPACK SET command.
CHANGE ATTRIBUTE    Changes data item attributes.
CHANGE BLOCKFACTOR Changes the block factor of a data set.
CHANGE BLOCKMAX     Changes the maximum block length of data set.
CHANGE CAPACITY     Changes the capacity of a data set.
CHANGE DEVICECLASS  Changes the device class of a data set.
CHANGE ITEM         Changes item attributes, order, security.
CHANGE ITEMSECURITY Changes user class access to a data item.
CHANGE PASSWORD     Changes a password.
CHANGE PRIMARYPATH  Changes primary path of a detail data set.
CHANGE SETSECURITY  Changes user class access to a data set.
CONTINUE [y/N] ? n
>
```

To get information about a specific command, type the command name after issuing the HELP command. In the example below, a list of all the ADD commands is displayed:

>help add

```
ADD FIELD           Adds a field to a data set.
ADD ITEM            Adds a data item to a database.
```

ADD PASSWORD	Adds a password to a database.
ADD PATH	Adds a path to an existing detail data set.
ADD SET	Adds a set to an existing database.
ADD SORT	Adds a new sort item to an existing path.

>

To get a description of the syntax and parameters and an example of a command, use the HELP command, the DBChange command, and the keyword. For example, HELP ADD ITEM produces syntax and parameter information along with an example.

### Specifying the Database to be Restructured

Whenever the DBCPLUS program is invoked, you must issue the BASE command to specify the database that you want to restructure. In the following example, the ORDERS database is specified for restructuring. By default, a new change file is created unless you specify otherwise. A series of change file status messages appear on your screen.

For example:

```
>base orders
    Creating new change file:
    Creating new change file: item information records
    Creating new change file: set information records
    Creating new change file: security table
    Creating new change file: control record
>
```

### How the Change File Works

Whenever you issue the BASE command, you can either create a new change file, update an existing change file, or purge an existing change file, as described below.

#### Creating a New Change File

A new change file is created by using the NEW option, which is the default, with the BASE command. The change file is a privileged file that is named *dbnameCF*.

#### Updating an Existing Change File

To update an existing change file, use the OLD option with the BASE command. The OLD option updates the change file by adding, deleting, or revising information that already exists in the change file. The existing change file is copied to a file named *dbnameCO*. Upon exiting the DBCPLUS program, or when the next BASE command is issued, *dbnameCO* is purged.

## Purging an Existing Change File

To purge an existing change file, use the PURGE CF option with the BASE command. The change file can also be purged by using the CANCEL COMMANDS command. If you use the CANCEL COMMANDS command after issuing the BASE command with the OLD option, the current *dbnameCF* is purged, and *dbnameCO* is renamed *dbnameCF*.

### Caution



- Neither designate a database name as *dbnameCF* nor save and reuse an existing *dbnameCF* after the DBAPLUS program is run; otherwise, the root file and the database structure will not match the change file information.
- Do not restore an old change file after the DBAPLUS program has completed successfully. Instead, create a new change file or use an XEQ command file.

---

For more information about the BASE command options, refer to chapter 10, “DBChange Plus Commands.”

## Using Some DBChange Plus Commands

You are now ready to use some DBChange Plus restructuring commands. In the following examples, a data set is added, a password is changed, and a field is deleted from the ORDERS database, respectively. For more information about the commands used in the following examples and other DBChange Plus commands, refer to the appropriate command heading in chapter 10, “DBChange Plus Commands.”

### Adding a Data Item

To add a data item to the database, use the ADD ITEM command. In the following example, the data item REORDER-QTY is added. It should appear before the data item STATE in the schema listing. The item subtype is J and the sublength is 4. User classes 11 and 14 have read access. User classes 12 and 13 have write access.

```
>add item reorder-qty (state) j 4 (11,14/12,13)
Addition of data item accepted.
```

### Changing a Password

To change a password in the database, use the CHANGE PASSWORD command. In the following example, the password BUYER is changed to AGENT. Note that passwords are case sensitive.

```
>change password BUYER AGENT
Change of password accepted.
```

### Deleting a Field

A **field** is a data item that is associated with a data set. To delete a field, use the DELETE FIELD command. In the following example, the field UNIT-COST is deleted from the detail data set INVENTORY:

```
>delete field inventory unit-cost  
Deletion of field accepted.
```

## Leaving the DBChange Program

Use the EXIT command to leave the DBCPLUS program. When you exit, a series of messages summarizing the information stored in the change file appears on your screen. The MPE XL prompt (:) displays on your screen once the DBCPLUS program terminates.

Following is an example of the EXIT command:

```
>exit
Please wait.....change file is being verified.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

***** DBCHANGE STATISTICS*****

Number of databases prepared for rename:      0
Number of databases prepared for restructure: 0
Number of databases prepared for erase :      0
Number of databases prepared for repack:      0
Number of databases prepared for check :      0
Number of databases prepared for fix   :      0

:
```

---

## Getting to Know the DBChange Plus Command Interface

This section describes the features and operation of the DBChange Plus command interface.

### Command Format

DBChange Plus commands consist of a command, a keyword, and parameters. For a complete list of commands, keywords, and acceptable abbreviations, refer to Table 10-1.

Figure 1-1 illustrates the various parts of a typical DBChange Plus command.

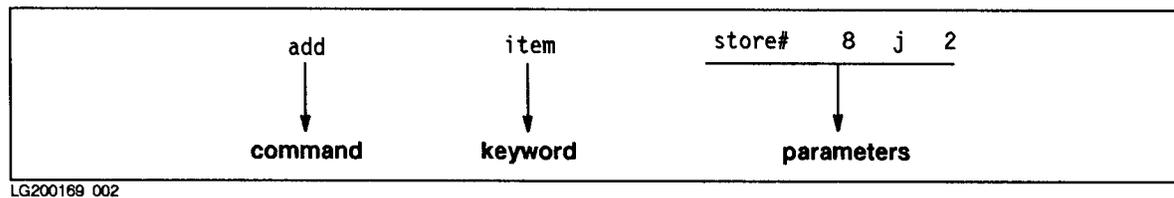


Figure 1-1. Typical DBChange Plus Command Format

For more information about DBChange Plus command formats, refer to the “Command Formats” section in chapter 10, “DBChange Plus Commands.”

## **Difference Between Deferred and Immediate Commands**

DBChange Plus has two types of commands: **immediate** and **deferred**.

Immediate commands are those commands that do not affect the database structure, but instead give information about or help with the operation of the DBCPLUS program. They are executed immediately when they are entered. The following are immediate commands:

- BASE
- CANCEL
- CONTROL
- EXIT
- HELP
- OUTPUT
- PERFORM COMMANDS
- PRINT
- RECOVER
- REDO
- REVIEW
- XEQ

Deferred commands are those commands that are stored in the change file and take effect upon execution of the DBAPLUS program. The following are deferred commands:

- ADD
- CHANGE
- CHECK
- DELETE
- ERASE
- FIX
- RENAME
- REORDER
- REPACK

## **Command Sequence**

The DBAPLUS program executes deferred commands in the order that most effectively protects your data and most efficiently handles your data throughput. For example, if the current change file contains a CHECK command, some ADD and DELETE commands, a FIX command, and a CHANGE command, DBAPLUS executes the commands in the following order:

1. CHECK
2. FIX
3. Other commands

## Terminating a Command

You can cause the DBCPLUS program to ignore the command you are currently entering by typing two slashes (//) and pressing the carriage return key. This returns you to the DBChange prompt (>). You might use this feature, for example, if you decide not to execute the current command or if you make an error and want to start over.

For example:

```
>change it//
>
```

This feature is only available in interactive mode.

## Executing Commands from a File

You can execute DBChange Plus commands from a file in the following two ways:

- Use a command file to make database changes interactively or in batch mode. This is useful if you frequently use the same sequence of commands. To use a command file in batch mode, it must be executed from a job file.
- Use a job file to run DBChange Plus in batch mode. This is useful if you want to make changes to the database at a later time. You can include command files in a job file.

### Including Comments in a File

In a command file or job file, if you want to include DBChange comments, enclose them in double angle brackets, for example, << *comment* >>. In the MPE command portion of a job file, the word comment at the beginning of a line indicates that the text on that line constitutes a comment.

### Using a Command File

A command file is a series of DBChange Plus commands and parameters. When the XEQ command is entered, DBChange Plus reads the specified command file and executes the commands until it reaches the end of the file. After DBChange Plus reads the last line in the command file, control is returned to the original command input device (the user terminal interactively or the the job input device in batch mode). If the command file contains the EXIT command, DBChange Plus terminates and the system prompt appears on your screen.

The following command file CMDFILE deletes the data item DATE, changes the capacity of the manual master data set CUSTOMER to 450, and exits the DBCPLUS program.

The XEQ command executes the command file CMDFILE as follows:

>xeq cmdfile

><<This command file creates a new change file for the>>

><<ORDERS database, deletes the data item DATE, and>>

><<changes the capacity of the manual master data set>>

><<CUSTOMER to 450.>>

>base orders new <---- *DBChange command*

Change file already exist for ORDERS, do you want to purge it [y/N] ? y  
Change file ORDERSCF has been purged.

Creating new change file:

Creating new change file: item information records

Creating new change file: set information records

Creating new change file: security table

Creating new change file: control record

>delete item date <---- *DBChange command*

Deletion of data item accepted.

>change capacity customer 450 <---- *DBChange command*

Change of set capacity, block factor, and/or device class accepted.

>exit <---- *DBChange command*

Please wait ... change file is being verified.

\*\*\*\*\* DBCHANGE STATISTICS\*\*\*\*\*

```
Number of databases prepared for rename:      0
Number of databases prepared for restructure:  1
Number of databases prepared for erase :      0
Number of databases prepared for repack:      0
Number of databases prepared for check :      0
Number of databases prepared for fix   :      0
```

:

### Using a Job File

A job file is a text file that contains DBChange Plus and MPE XL commands. A job file is used in batch mode. Command files can be included in a job file. In the following example, the job DBCJOB executes some MPE XL commands, some DBChange commands, and the command file CMDFILE, which was explained in the previous example. Note that the command file CMDFILE contains the EXIT command.

```
!job dbcjob,kelly/user.finance/mgr.database/all
comment** Show time and date dbcjob begins
!showtime
!run dbcplus.pub.sys
base orders new
change password CLERK AGENT
xeq cmdfile
comment** Show time and date dbcjob ends
!showtime
!tell kelly.finance.database dbcjob is done
!eoj
```

For more information about job files and streaming jobs, refer to “Creating a DBAPLUS Job File from within the DBChange Program” in chapter 9, “Applying Changes to the Database Structure.”

## Issuing MPE XL Commands from within the DBCPLUS Program

Many MPE XL commands, including RUN commands, can be entered interactively. You can issue MPE XL commands from within the DBCPLUS program either by entering a colon (:) after the DBCPLUS prompt and then typing the command, or by using **Break** and RESUME. In the following example, at the DBCPLUS prompt (>) a colon is entered and the MPE XL LISTF command is issued:

```
>:listf

FILENAME

ORDERS      ORDERS01    ORDERS02    ORDERS03    ORDERS04    ORDERS05
ORDERS06    ORDERSCF    ORDERSJJ    PCHGFMT     TESTDB      TESTDB01
TESTDB02    TESTDB03    TESTDB04    TESTDB05    TESTBCF     TESTDBC0

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>
```

If you press **Break** to access MPE XL, type the MPE XL command RESUME when you are ready to return to the DBCPLUS program. When READ PENDING is displayed, press **Return** and the DBChange prompt reappears. For example:

```
Break
:~listf

FILENAME

ORDERS      ORDERS01    ORDERS02    ORDERS03    ORDERS04    ORDERS05
ORDERS06    ORDERSCF    ORDERSJJ    PCHGFMT     TESTDB      TESTDB01
TESTDB02    TESTDB03    TESTDB04    TESTDB05    TESTBCF     TESTDBC0
:resume
READ pending

Return
>
```

### Caution



---

Using **Break** while running the DBAPLUS program or after issuing a PERFORM COMMANDS can result in a damaged database.

---

---

## Redirecting Input, Output, and the Schema Listing

By default, command input comes from \$STDIN (the user terminal interactively; the job file in batch mode). By default, output is directed to \$STDLIST (the user terminal interactively or the job's \$STDLIST spool file in batch mode).

You can redirect input, output, and the schema listing by issuing the following formal file designators either before or during DBChange Plus operation. (To issue them during DBChange Plus operation, be sure to enter a colon (:) after the DBCPLUS prompt (>).)

- DBCIN—the formal file designator for command input. If you set a file equation for DBCIN before running the DBCPLUS program, commands are read from the named file.

To read input from a file, use the following file equation:

```
:file dbcin= InFileName
```

- DBCOUT—the formal file designator for command output. This file can be a temporary file or a permanent file large enough to hold the contents of the output. The DBCOUT file equation must be used with the OUTPUT LP command.

To send output to a new permanent file, use the following file equation:

```
:file dbcout= FileName;dev=disc;rec=-80,16,f,ascii;disc= n;save;nocctl
```

where *FileName* is the name of the new permanent file, and *n* is the approximate number of lines in the output file.

### Printing the Schema to an MPE XL File

To print the schema to a new permanent MPE XL file, use the PRINT SCHEMA command specifying a *FileName*. The *FileName* must adhere to the MPE XL file naming conventions. (Refer to the *MPE XL General Users Reference Manual*.) DBChange automatically builds a fixed ASCII disk file which can accommodate a file size up to 60000 lines. If the file does not use all the allotted disk space, any unused disk space is returned back to the system.

### Caution



When redirecting output to a disk file, be sure the file is large enough to accommodate all output. If the file is not large enough, the DBCPLUS or DBAPLUS programs can abort causing the database to be in an inconsistent state.

---

### Printing the Schema to an MPE XL File in Interactive Mode

When you issue the PRINT SCHEMA *FileName* command in interactive mode, the new file is opened and the DBCPLUS program verifies if a file with the same name already exists. If one does, you are prompted and must confirm if you want the existing file purged. If a file with the same name does not exist, the new file is built and the schema is printed to the new file.

In the following example, the schema for the ORDERS database is interactively printed to a file named ORDERSZ, which already exists:

```
>print schema ordersz
  Old schema file ORDERSZ exists.  Purge and create a new file [y/N]? y
  Schema is printed.

>
```

### Printing the Schema to an MPE XL File in Batch Mode

When you issue the PRINT SCHEMA *FileName* command in batch mode and a file with the same name already exists, the new file is not created and the following message appears in your output:

```
  Duplicate schema file.  Cannot process PRINT SCHEMA.
```

If a file with the same name does not exist, the new file is built and the schema is printed to the new file.

Refer to PRINT SCHEMA in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

---

## How DBChange Plus Handles Errors

When you enter a DBChange Plus command, the DBCPLUS program checks the syntax of the command and, if the command is incorrect or invalid, issues an error message to \$STDLIST.

If you receive an error message while using the DBCPLUS program interactively, reenter the command and continue processing. In batch mode, the DBCPLUS program reports the error and continues reading commands until the EXIT command is reached.

DBChange Plus usually prints an error message immediately after the command that caused it. However, because of the deferred nature of the restructuring commands, some errors are not reported at the time they are detected. Instead, they are reported when the next BASE, EXIT, or PERFORM COMMANDS command is entered.

For example, a blocking factor change could potentially be recommended after each command that changes the media record

size. If you enter several of these commands, you should review the error messages before deciding which corrective action to take.

For a complete description of errors, their cause, and corrective action, refer to appendix A, “DBChange Plus Messages.”

---

## How DBChange Plus Helps You Avoid and Recover from Errors

Every attempt is made to prevent you from inadvertently damaging the database. When you are using the DBCPLUS program interactively, commands that could result in the loss of data, such as COPY, DELETE and ERASE, require your confirmation before they are accepted into the change file.

When you are prompted for confirmation, the default value is always an uppercase letter. For example, if your choice is yes or no, and no is the default, the prompt looks like this:

[y/N]

As an added safety feature, DBChange Plus allows you to cancel the ERASE and REPACK commands by issuing the CANCEL command. Deleted items and sets can be recovered by issuing the RECOVER command.

### Caution



---

You must use CANCEL and RECOVER *before* executing the DBAPLUS program; otherwise, you cannot recover changes made to the database.

---

You can cancel all commands in the current DBChange session by issuing the CANCEL COMMANDS command.

## Using Job Control Words to Check the Status of DBChange Plus

DBChange Plus contains two user **job control words** (JCWs). These user JCWs can be checked prior to executing the next step in processing to determine if errors occurred in either the DBCPLUS program or the DBAPLUS program. For example, the following job checks to make sure the DBCPLUS program processed without any errors, and then checks to make sure the DBAPLUS program processed without any errors:

```
!job checkjob,kelly/user.finance/mgr,database
!run dbcplus.pub.sys
base orders
change capacity sales 80000
exit
!if dbcplus jcw=0 then
!   file dbchgf=orderscf
!   run dbaplus.pub.sys
!   if dbaplus=0 then
!     run yourjob
!   endif
!endif
```

!eoj

The DBCPLUSJCW is set within the DBCPLUS program and checks for such errors as incorrect command syntax and incorrect database name. If no errors are encountered, then the JCW is set to zero and the job is successfully processed. If errors are encountered, the JCW is set to a number other than zero and the DBAPLUS program is not run.

The DBAPLUSJCW is set within the DBAPLUS program. If no errors are encountered, the JCW is set to zero and the job is successfully processed. If errors are encountered, the JCW is set to a number other than zero and the DBAPLUS program aborts.

If errors are encountered, the error messages are directed to \$STDLIST.

©

## Displaying Information about the Database

---

This chapter explains how to display information about the structure of a TurboIMAGE/XL database. It provides instructions for and examples of the commands used to display information about the database. Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn how to do more DBChange Plus tasks, refer to chapters 2 and 4 through 9. If you understand DBChange Plus tasks and only want to see command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### Summary of Commands to Display Information about the Database

DBChange Plus provides specific commands for displaying information about the database. Table 3-1 alphabetically lists the commands used for printing and reviewing database structure information and summarizes their functions.

The DBCPLUS program provides two ways to display information about the database structure, as follows:

- Printing the schema to obtain information about the entire database structure.
- Using the various REVIEW commands to display information about specific parts of the database. For example, if you only want information about a specific data set, then use the REVIEW SET command.

In both cases, all changes currently stored in the change file are displayed. Because information stored in the change file is not permanently applied until you run the DBAPLUS program, the PRINT SCHEMA and REVIEW commands are useful when verifying structural changes before you actually restructure the database or when viewing the original database structure before making changes.

#### Note



Before invoking any of the following commands, you must log on to the system as the database creator, run the DBCPLUS program and specify the database for which you want to display information. Refer to chapter 1, “Introduction to DBChange Plus.”

---

Unless redirected, output is printed to \$STDLIST. Refer to “Redirecting Input, Output, and the Schema” in chapter 1, “Introduction to DBChange Plus.”

**Table 3-1.**  
**Summary of Commands to Display Information about the Database**

Command	Keyword	Description
PRINT	SCHEMA	Prints the database schema.
REVIEW	BLOCKS	Displays information about the data sets in the database. Includes data set type, field count, path count, entry length, capacity, blocking factor, and block length.
	FIELDS	Displays information about the fields (data items) of master and detail data sets. Includes search items, sort items, and linked data sets.
	ITEMS	Displays information about the data items in the database. Includes attributes and security.
	PASSWORDS	Displays information about the user classes and associated passwords in the database.
	PATHS	Displays information about the paths in a data set. Includes key item, linked data set, and sort item.
	SETS	Displays information about the data sets in the database. Includes data set type, capacity, blocking factor, block length, device class, and security.

## Printing the Schema

Use the PRINT SCHEMA command to print the schema. In the following example, the schema output is directed to the printer. A message is displayed confirming that the schema output is printed.

```
>output lp
>print schema
Schema is printed.
```

>

If you want to print the schema to a disk file, use the *filename* parameter as shown in the following example. DBChange Plus automatically builds a 60000-line, ASCII MPE XL file; however, if you have a schema file that uses more than 60000 lines of space, you should use the DBCOUT file equate to build a file large enough to accommodate your schema file.

```
>print schema filename
Schema is printed.
```

Refer to “Redirecting Input, Output, and the Schema Listing” in chapter 1, “Introduction to DBChange Plus,” or refer to “PRINT SCHEMA” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command.

## Aborting the PRINT SCHEMA Command

To abort the PRINT SCHEMA command anytime during an interactive session, press **CONTROL** Y. The DBCPLUS program prints << CONTROL-Y >> and returns the prompt for a new command. For example:

```
>print schema

BEGIN DATABASE ORDERS;

PASSWORDS:
  11 CREDIT ;
  12 BUYER  ;
  13 SHIP-REC;
  14 CLERK  ;
  18 DO-ALL ;

ITEMS:
  ACCOUNT      , I4(0,11,12,13,14,18/);
  BINNUM       , Z2(/13);

CONTROL y

<< CONTROL-Y >>
>
```

---

## Using REVIEW Commands to Display Information about the Database

The DBCPLUS program has several REVIEW commands that display information about the database structure with the changes stored in a change file. The remainder of this chapter provides an example for each REVIEW command.

### Displaying User Classes and Passwords

To display one or more user classes and associated passwords in the database, use the REVIEW PASSWORDS command. In the following example, @ is the parameter used to display all user classes and associated passwords in the ORDERS database:

```
>review passwords @
```

```
REVIEW PASSWORDS:
```

User Class	Password
11	CREDIT
12	BUYER
13	SHIP-REC
14	CLERK
18	DO-ALL

```
>
```

Refer to “REVIEW PASSWORDS” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command.

## Displaying Data Items

To display information about one or more data items in the database, use the REVIEW ITEMS command. The REVIEW ITEMS command displays item numbers as they appear in the schema, along with their item name, attributes, and security information. In the following example, @ is the parameter used to display all the items in the ORDERS database. Because the database has more than 20 data items, the DBCPLUS program asks if you want to continue, that is, review more data items.

```
>review items @
```

```
REVIEW ITEMS:
```

Itm No.	Item Name	Attrib	Security
1	ACCOUNT	I4	(11,12,13,14,18/)
2	BINNUM	Z2	(/13)
3	CITY	X12	(12,13,14/11)
4	CREDIT-RATING	R2	(/14)
5	DATE	X6	(11,12,13,14,18/)
6	DELIV-DATE	X6	(/14)
7	DESCRIPTION	X20	(11,12,13,14,18/)
8	FIRST-NAME	X10	(14/11)
9	INITIAL	U2	(14/11)
10	LAST-NAME	X16	(14/11)
11	LASTSHIPDATE	X6	(12/)
12	ONHANDQTY	J2	(14/12)
13	PRICE	J2	(14/)
14	PURCH-DATE	X6	(11/14)
15	QUANTITY	I1	(/14)
16	STATE	X2	(12,13,14/11)
17	STOCK#	U8	(11,12,14,18/)
18	STREET-ADD	X26	(12,13,14/11)
19	SUPPLIER	X16	(12,13/)
20	TAX	J2	(14/)

CONTINUE [y/N] ?

```
>
```

Refer to “REVIEW ITEMS” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command.

## Displaying Information about Data Sets

This section describes the REVIEW commands used to display information about data sets.

### Displaying Data Sets

To display information about one or more data sets in the database, use either the REVIEW SETS or REVIEW BLOCKS command. Both the REVIEW SETS and REVIEW BLOCKS commands display the data set name, the data set type, capacity, blocking factor and maximum block length. The REVIEW SETS command also displays the data set number as it appears in the schema, the device class, and the security information; while the REVIEW BLOCKS command displays field count, path count, data entry length with and without chain counters and pointers, media records, and block length. Note that the information displayed in the REVIEW BLOCKS command is similar to the data set summary table printed by the schema processor.

In the following example, REVIEW SETS is used to display information about the data set PRODUCT in the ORDERS database:

```
>review sets product
```

```
REVIEW SETS:
```

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
2	PRODUCT	M	307	16	498		(13,14/12,18)

```
>
```

In the example below, the REVIEW BLOCKS command is used to display information about the data set PRODUCT in the ORDERS database:

```
>review blocks product
```

```
REVIEW BLOCKS:
```

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
PRODUCT	M	2	2	14	31	307	16	498	498

```
>
```

Refer to “REVIEW SETS” and “REVIEW BLOCKS” in chapter 10, “DBChange Plus Commands,” for a detailed description of the commands.

## Displaying Fields

To display one or more fields in a data set, use the REVIEW FIELDS command. For both master and detail data sets, the REVIEW FIELDS command displays the data set number, as it appears in the schema, and the data set name and type. However, the master data set output differs from the detail data set output in that the master data set output displays the detail data set associated with the master data set and the search item linking that detail data set to the master data set. In the detail data set output, the REVIEW FIELDS command flags the primary path, search items, and sort item; it also displays the names of the master data set to which the detail data set is linked and the sort item name.

In the following example, the REVIEW FIELDS command is used to display the fields in the master data set CUSTOMER in the ORDERS database:

```
>review fields customer
```

```
REVIEW FIELDS (MASTER SET):
```

```
Data Set No.: 2   Name: CUSTOMER   Type: M
```

Fld No.	Field Name	Detail Set Name	Path's Search Item
1	ACCOUNT	SALES	ACCOUNT
2	LAST-NAME		
3	FIRST-NAME		
4	INITIAL		
5	STREET-ADD		
6	CITY		
7	STATE		
8	ZIP		
9	CREDIT-RATING		

```
>
```

In the example below, the REVIEW FIELDS command is used to display the fields in the detail data set SALES:

```
>review fields sales
```

```
REVIEW FIELDS (DETAIL SET):
```

```
Data Set No.: 6 Name: SALES Type: D
```

Fld No.	Field Name	Pr	Sr	So	Master Set Name	Sort Item Name
1	ACCOUNT			Y	CUSTOMER	PURCH-DATE
2	STOCK#	Y	Y		PRODUCT	
3	QUANTITY					
4	PRICE					
5	TAX					
6	TOTAL					
7	PURCH-DATE			Y Y	DATE-MASTER	
8	DELIV-DATE			Y	DATE-MASTER	

```
>
```

Refer to “REVIEW FIELDS” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command.

### Displaying Paths

To display information about the paths in one or more data sets, use the REVIEW PATHS command. For both master and detail data sets, the REVIEW PATHS command displays the data set number, as it appears in the schema, and the data set name and type. However, the master data set output differs from the detail data set output in that the master data set output displays the key item, the detail data set linked to the master data set, and the sort item name. In the detail data set output, the REVIEW PATHS command displays the search item name linking the detail data set to the master data set, the master data set to which the detail data set is linked, the sort item name, and the primary path.

In the following example, CUSTOMER is the name of a master data set in the ORDERS database:

```
>review paths customer
```

```
REVIEW PATHS (MASTER SET):
```

```
Data Set No.: 1   Name:  CUSTOMER   Type:  M
Key(Search) Item   Detail Set Name   Sort Item Name
-----
ACCOUNT           SALES           PURCH-DATE
```

```
>
```

In the example below, SALES is the name of a detail data set in the ORDERS database:

```
>review paths sales
```

```
REVIEW PATHS (DETAIL SET):
```

```
Data Set No.: 3   Name:  SALES   Type:  D
Search Item Name   Master Set Name   Sort Item Name   Primary
-----
ACCOUNT           CUSTOMER           PURCH-DATE
STOCK#            PRODUCT                                Y
```

```
>
```

Refer to “REVIEW PATHS” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command.

## Changing Data Items

---

This chapter explains how to change data items in a TurboIMAGE/XL database. It provides instructions for and examples of the various commands used to change data items. Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn how to do more DBChange Plus tasks, refer to chapters 2, 3, and 5 through 9.

If you understand DBChange Plus tasks and only want to see the command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### Summary of Commands to Change Data Items

DBChange Plus has several commands for changing data items in the database. Some deal with several types of data item changes; others are specific to data item security. For example, the CHANGE ITEM command deals with changes to the attributes and security of a data item and the order that the data item appears in the data item list; while CHANGE ITEMSECURITY only deals with data item security. If you are making several types of changes to a data item, such as subitem count, data type, and length, then CHANGE ITEM is a more efficient command. However, if you are making only security changes to a data item, then CHANGE ITEMSECURITY is a better choice. This chapter covers only those commands that deal with several types of changes to a data item. For more information about how to use commands that deal specifically with security, refer to chapter 7, “Changing Database Security.”

Table 4-1 alphabetically lists all the commands that involve changes to data items and summarizes their functions.

#### Note



- Before invoking any of the following commands, you must log on to the system as the database creator, run the DBCPLUS program, and specify the database that you want to restructure. Refer to chapter 1, “Introduction to DBChange Plus.”
  - The commands are not actually applied until the DBAPLUS program is used. Refer to chapter 9, “Applying Changes to the Database Structure.”
-

**Table 4-1. Summary of Commands to Change Data Items**

Command	Keyword	Description
ADD	ITEM	Adds a data item to the database.
CHANGE	ATTRIBUTE	Changes the subitem count, type, and/or length of a data item.
	ITEM	Changes the subitem count, type, length, security, and/or order of a data item.
	ITEMSECURITY	Changes user class access to a data item (security command only).
DELETE	ITEM	Deletes a data item and its fields and associated paths from the database.
	ITEMSECURITY	Deletes all security specifications from a data item (security command only).
RECOVER	ITEM	Cancels a data item deletion.
RENAME	ITEM	Renames a data item.
REORDER	ITEM	Reorders the sequence in which the data item resides in the database.

---

## Helpful TurboIMAGE/XL Information

Remember the following TurboIMAGE/XL characteristics when making changes to data items:

- A data item name must be unique within the database. It must be from 1 to 16 characters long, and the first character must be alphabetic.
- No more than 1023 data items can exist in a TurboIMAGE/XL database.
- Valid data item types for TurboIMAGE/XL are I, J, K, R, U, X, Z, or P. For data item type conversion information, refer to Table 4-2.
- The unit of measure for data item length depends upon the type designator and can be a **halfword**, a **byte**, or a **nibble**. A halfword is 16 bits, a byte is 8 bits, and a nibble is 4 bits or a half byte.
- A data item cannot exceed 2047 halfwords in length and must be an even number of bytes.
- The subitem count must be an integer from 1 to 255. For item types U, X, or Z, the subitem count multiplied by the item length must be an even number. For item type P, the subitem count multiplied by the item length must be evenly divisible by 4.

- The subitem length must be from 1 to 255, depending on the type designator as follows:

Type	Length
U, X, Z	Maximum is 255 bytes.
P	Maximum is 255 nibbles.
I, J	Must be 1, 2, or 4 halfwords.
K	Must be 1 or 2 halfwords.
R	Must be 2 or 4 halfwords.

- A compound data item is a data item with a subitem count of more than one. Search and sort items cannot be compound items.

When changing data item length, reevaluate the blocking factor of the sets containing the items.

For more information about TurboIMAGE/XL data items, refer to the *TurboIMAGE/XL Database Management System Reference Manual*.

## Item Type Conversions

DBChange Plus supports type conversion for all data item types defined in the *TurboIMAGE/XL Reference Manual* with the exception of I4 and J4. Table 4-2 shows the data item type conversions supported by DBChange Plus. An x in the appropriate box indicates that you can convert the current item type to the new item type.

**Table 4-2. DBChange Plus Supported Item Type Conversions**

Current Item Type	New Item Type							
	I1,I2	J1,J2	K1	P	R2,R4	U	X	Z
I1,I2	x	x	x	x	x	x	x	x
J1,J2	x	x	x	x	x	x	x	x
K1	x	x	x	x	x	x	x	x
P	x	x	x	x		x	x	x
R2,R4	x	x	x		x			
U						x	x	
X						x	x	
Z	x	x	x	x		x	x	x

**Caution**


---

When converting data item types, overflow, underflow, and truncation may occur if you are reducing data item size. Be sure to verify that data item conversions do not result in loss of data.

---

**Adding a Data Item**

To add a data item to the database, use the ADD ITEM command. In this section, the new data item STORE# is added to the ORDERS database. Before adding the data item, the REVIEW ITEMS output looks like this:

Itn No.	Item Name	Attrib	Security
-----			
1	ACCOUNT	I4	(11,12,13,14,18/)
2	BINNUM	Z2	(/13)
3	CITY	X12	(12,13,14/11)
4	CREDIT-RATING	R2	(/14)
5	DATE	X6	(11,12,13,14,18/)
6	DELIV-DATE	X6	(/14)
7	DESCRIPTION	X20	(11,12,13,14,18/)
8	FIRST-NAME	X10	(14/11)
9	INITIAL	U2	(14/11)
10	LAST-NAME	X16	(14/11)
11	LASTSHIPDATE	X6	(12/)
12	ONHANDQTY	J2	(14/12)
13	PRICE	J2	(14/)
14	PURCH-DATE	X6	(11/14)
15	QUANTITY	I1	(/14)
16	STATE	X2	(12,13,14/11)
17	STOCK#	U8	(11,12,14,18/)
18	STREET-ADD	X26	(12,13,14/11)
19	SUPPLIER	X16	(12,13/)
20	TAX	J2	(14/)
21	TOTAL	J2	(11,14/)
22	UNIT-COST	P8	(/12)
23	ZIP	X6	(12,13,14/11)

In the example below, the data item STORE# is added using the ADD ITEM command. It should appear before the data item TAX in the data item list. The sub-count is 8, the type is J, and the length is 2. User classes 11 and 12 have read access, and user class 14 has write access to the new data item. The DBCPLUS program displays a message confirming that the new data item is accepted.

```
>add item store# (tax) 8 j 2 (11,12/14)
  Addition of data item accepted.
```

>

Refer to “ADD ITEM” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the new data item STORE# has been added to the change file, use the REVIEW ITEMS command. In the following example, the @ parameter is used to display all the data items in the ORDERS database. You can see that the data item STORE# has been added.

```
>review items @
```

REVIEW ITEMS:

Itm No.	Item Name	Attrib	Security	
1	ACCOUNT	I4	(11,12,13,14,18/)	
2	BINNUM	Z2	(/13)	
3	CITY	X12	(12,13,14/11)	
4	CREDIT-RATING	R2	(/14)	
5	DATE	X6	(11,12,13,14,18/)	
6	DELIV-DATE	X6	(/14)	
7	DESCRIPTION	X20	(11,12,13,14,18/)	
8	FIRST-NAME	X10	(14/11)	
9	INITIAL	U2	(14/11)	
10	LAST-NAME	X16	(14/11)	
11	LASTSHIPDATE	X6	(12/)	
12	ONHANDQTY	J2	(14/12)	
13	PRICE	J2	(14/)	
14	PURCH-DATE	X6	(11/14)	
15	QUANTITY	I1	(/14)	
16	STATE	X2	(12,13,14/11)	
17	STOCK#	U8	(11,12,14,18/)	
18	STREET-ADD	X26	(12,13,14/11)	
19	SUPPLIER	X16	(12,13/)	
20	STORE#	8J2	(11,12/14)	<---- data item added
21	TAX	J2	(14/)	
22	TOTAL	J2	(11,14/)	

23	UNIT-COST	P8	(/12)
24	ZIP	X6	(12,13,14/11)

## Deleting a Data Item

To delete a data item, use the DELETE ITEM command. In this section, the data item LASTSHIPDATE is deleted from the ORDERS database.

When deleting data items, remember the following:

- If the data item to be deleted is a field in one or more data sets, those fields are also deleted.
- If the data item to be deleted is associated with one or more paths, those paths are also deleted.
- If the data item to be deleted is a key item in an automatic master data set, the automatic master data set is deleted.
- If the item to be deleted is a sort item in any data set, you must first use the DELETE SORT command.
- When deleting a key item in a manual master data set, an error is not reported until you exit the DBCPLUS program.

### Caution



- If the data item to be deleted is a field in one or more data sets, and those fields contain data, the data in those fields will be lost.
- If the data item to be deleted is the only key item in a manual master data set, an error will occur.

Before deleting a data item, the REVIEW ITEMS output looks like this:

Itm No.	Item Name	Attrib	Security	
1	ACCOUNT	I4	(11,12,13,14,18/)	
2	BINNUM	Z2	(/13)	
3	CITY	X12	(12,13,14/11)	
4	CREDIT-RATING	R2	(/14)	
5	DATE	X6	(11,12,13,14,18/)	
6	DELIV-DATE	X6	(/14)	
7	DESCRIPTION	X20	(11,12,13,14,18/)	
8	FIRST-NAME	X10	(14/11)	
9	INITIAL	U2	(14/11)	
10	LAST-NAME	X16	(14/11)	
11	LASTSHIPDATE	X6	(12/)	<---- data item to be deleted
12	ONHANDQTY	J2	(14/12)	

13	PRICE	J2	(14/)
14	PURCH-DATE	X6	(11/14)
15	QUANTITY	I1	(/14)
16	STATE	X2	(12,13,14/11)
17	STOCK#	U8	(11,12,14,18/)
18	STREET-ADD	X26	(12,13,14/11)
19	SUPPLIER	X16	(12,13/)

20	STORE#	8J2	(11,12/14)
21	TAX	J2	(14/)
22	TOTAL	J2	(11,14/)
23	UNIT-COST	P8	(/12)
24	ZIP	X6	(12,13,14/11)

In the example below, the data item LASTSHIPDATE is deleted using the DELETE ITEM command. LASTSHIPDATE is also a field in the detail data set INVENTORY and will be deleted from the data set. In the following interactive session, a message is displayed asking you to confirm the deletion of the data item. In batch mode, the data item is deleted without confirmation.

```
>delete item lastshipdate
Deleting this data item will cause loss of data, delete [y/N]? y
Deletion of data item accepted.
```

>

Refer to “DELETE ITEM” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data item LASTSHIPDATE has been deleted from the change file, use the REVIEW ITEMS command. In the following example, the @ parameter is used to display all the data items in the ORDERS database. You can see that LASTSHIPDATE has been deleted.

```
>review items @
```

REVIEW ITEMS:

Itm No.	Item Name	Attrib	Security	
1	ACCOUNT	I4	(11,12,13,14,18/)	
2	BINNUM	Z2	(/13)	
3	CITY	X12	(12,13,14/11)	
4	CREDIT-RATING	R2	(/14)	
5	DATE	X6	(11,12,13,14,18/)	
6	DELIV-DATE	X6	(/14)	
7	DESCRIPTION	X20	(11,12,13,14,18/)	
8	FIRST-NAME	X10	(14/11)	
9	INITIAL	U2	(14/11)	
10	LAST-NAME	X16	(14/11)	
11	ONHANDQTY	J2	(14/12)	<---- LASTSHIPDATE deleted
12	PRICE	J2	(14/)	
13	PURCH-DATE	X6	(11/14)	
14	QUANTITY	I1	(/14)	

15	STATE	X2	(12,13,14/11)
16	STOCK#	U8	(11,12,14,18/)
17	STREET-ADD	X26	(12,13,14/11)
18	SUPPLIER	X16	(12,13/)
19	STORE#	8J2	(11,12/14)
20	TAX	J2	(14/)
21	TOTAL	J2	(11,14/)
22	UNIT-COST	P8	(/12)
23	ZIP	X6	(12,13,14/11)

If you want to verify that the field LASTSHIPDATE is deleted from the detail data set INVENTORY, use the REVIEW FIELDS command, as shown in the following example:

```
>review fields inventory
```

```
REVIEW FIELDS (DETAIL SET):
```

```
Data Set No.: 5 Name: INVENTORY Type: D
```

```
Fld No. Field Name Pr Sr So Master Set Name Sort Item Name
```

```
-----
```

1	STOCK#		Y		PRODUCT	
2	ONHANDQTY					
3	SUPPLIER	Y	Y		SUP-MASTER	
4	UNIT-COST					
5	BINNUM					<---- LASTSHIPDATE deleted

---

## Recovering a Data Item

In the previous section, you deleted the data item LASTSHIPDATE from the ORDERS database. Because DBChange Plus stored this deletion in the change file, you can recover it by using the RECOVER ITEM command.

When recovering data items, remember the following:

- If the deleted data item is recovered and the data set which contained the data item as a field was changed between the time the data item was deleted and then recovered, you must explicitly replace the field in the changed data set with the ADD FIELD command.
- If you delete a data item and then add a new data item with the same name, you cannot recover the original data item.
- Once you run the DBAPLUS program, you cannot recover a deleted data item under any circumstances.

In the following example, the data item LASTSHIPDATE is recovered. The DBCPLUS program returns a message confirming recovery of the deleted data item.

```
>recover item lastshipdate  
Recovery of data item accepted.
```

```
>
```

Refer to “RECOVER ITEM” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the deleted data item LASTSHIPDATE is recovered, use the REVIEW ITEMS command. In the following example, the @ parameter is used to display all the data items in the ORDERS database. You can see that the data item LASTSHIPDATE is recovered and inserted in its original place in the data item list.

```
>review items @
```

```
REVIEW ITEMS:
```

Itm No.	Item Name	Attrib	Security	
1	ACCOUNT	I4	(11,12,13,14,18/)	
2	BINNUM	Z2	(/13)	
3	CITY	X12	(12,13,14/11)	
4	CREDIT-RATING	R2	(/14)	
5	DATE	X6	(11,12,13,14,18/)	
6	DELIV-DATE	X6	(/14)	
7	DESCRIPTION	X20	(11,12,13,14,18/)	
8	FIRST-NAME	X10	(14/11)	
9	INITIAL	U2	(14/11)	
10	LAST-NAME	X16	(14/11)	
11	LASTSHIPDATE	X6	(12/)	<---- data item recovered
12	ONHANDQTY	J2	(14/12)	
13	PRICE	J2	(14/)	
14	PURCH-DATE	X6	(11/14)	
15	QUANTITY	I1	(/14)	
16	STATE	X2	(12,13,14/11)	
17	STOCK#	U8	(11,12,14,18/)	
18	STREET-ADD	X26	(12,13,14/11)	
19	SUPPLIER	X16	(12,13/)	
20	STORE#	8J2	(11,12/14)	
21	TAX	J2	(14/)	
22	TOTAL	J2	(11,14/)	
23	UNIT-COST	P8	(/12)	
24	ZIP	X6	(12,13,14/11)	

Before it was deleted, LASTSHIPDATE was also a field in the detail data set INVENTORY. The RECOVER ITEM command put the data item LASTSHIPDATE back into the detail data set INVENTORY in its original place in the field list. If you want to verify this, use the REVIEW FIELDS command, as shown in the following example:

```
>review fields inventory
```

```
REVIEW FIELDS (DETAIL SET):
```

```
Data Set No.: 5 Name: INVENTORY Type: D
```

Fld No.	Field Name	Pr	Sr	So	Master Set Name	Sort Item Name
1	STOCK#			Y	PRODUCT	
2	ONHANDQTY					
3	SUPPLIER	Y	Y		SUP-MASTER	
4	UNIT-COST					
5	LASTSHIPDATE			Y	DATE-MASTER	<---- field recovered
6	BINNUM					

## Reordering a Data Item

To reorder the data item sequence in the data item list, use the REORDER ITEM command. In this section, the data item UNIT-COST is reordered in the ORDERS database. Before reordering the data item, the REVIEW ITEMS output looks like this:

Itm No.	Item Name	Attrib	Security
1	ACCOUNT	I4	(11,12,13,14,18/)
2	BINNUM	Z2	(/13)
3	CITY	X12	(12,13,14/11)
4	CREDIT-RATING	R2	(/14)
5	DATE	X6	(11,12,13,14,18/)
6	DELIV-DATE	X6	(/14)
7	FIRST-NAME	X10	(14/11)
8	INITIAL	U2	(14/11)
9	LAST-NAME	X16	(14/11)
10	LASTSHIPDATE	X6	(12/)
11	ONHANDQTY	J2	(14/12)
12	DESCRIPTION	4U25	(11,14/18)

13	PRICE	J2	(14/)	
14	PURCH-DATE	X6	(11/14)	
15	QUANTITY	I1	(/14)	
16	STATE	X2	(12,13,14/11)	
17	STOCK#	X10	(11,12,14,18/)	
18	STREET-ADD	X26	(12,13,14/11)	
19	VENDOR	X16	(12,13/)	
20	STORE#	8J2	(11,12/14)	
21	TAX	J2	(14/)	
22	TOTAL	J2	(11,14/)	
23	UNIT-COST	P8	(/12)	<---- data item to be reordered
24	ZIP	X6	(12,13,14/11)	

In the example below, the data item UNIT-COST is reordered to appear before the data item TAX in the data item list. If TAX were omitted, the data item UNIT-COST would appear at the end of the data item list. The DBCPLUS program displays a message confirming that the data items are reordered.

```
>reorder item unit-cost (tax)
  Reorder of data items accepted.
```

>

Refer to “REORDER ITEM” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data item has been reordered in the data item list, use the REVIEW ITEMS command. In the following example, the @ parameter is used to display all the data items in the ORDERS database. You can see that the data item UNIT-COST now appears before the data item TAX in the data item list.

```
>review items @
```

```
REVIEW ITEMS:
```

Itm No.	Item Name	Attrib	Security
-----			
1	ACCOUNT	I4	(11,12,13,14,18/)
2	BINNUM	Z2	(/13)
3	CITY	X12	(12,13,14/11)
4	CREDIT-RATING	R2	(/14)
5	DATE	X6	(11,12,13,14,18/)
6	DELIV-DATE	X6	(/14)
7	FIRST-NAME	X10	(14/11)
8	INITIAL	U2	(14/11)
9	LAST-NAME	X16	(14/11)

10	LASTSHIPDATE	X6	(12/)	
11	ONHANDQTY	J2	(14/12)	
12	DESCRIPTION	4U25	(11,14/18)	
13	PRICE	J2	(14/)	
14	PURCH-DATE	X6	(11/14)	
15	QUANTITY	I1	(/14)	
16	STATE	X2	(12,13,14/11)	
17	STOCK#	X10	(11,12,14,18/)	
18	STREET-ADD	X26	(12,13,14/11)	
19	VENDOR	X16	(12,13/)	
20	STORE#	8J2	(11,12/14)	
21	UNIT-COST	P8	(/12)	<---- data item reordered
22	TAX	J2	(14/)	
23	TOTAL	J2	(11,14/)	
24	ZIP	X6	(12,13,14/11)	

---

## Renaming a Data Item

To rename a data item, use the RENAME ITEM command. In this section, the data item SUPPLIER in the ORDERS database is renamed. Before renaming the data item, it looks like this:

Itm No.	Item Name	Attrib	Security
19	SUPPLIER	X16	(12,13/)

In the example below, the data item SUPPLIER is renamed VENDOR. The DBCPLUS program displays a message confirming that the data item is renamed.

```
>rename item supplier vendor
Rename of data item accepted.
```

```
>
```

Refer to “RENAME ITEM” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data item has been renamed, use the REVIEW ITEMS command. In the following example, VENDOR is the new name of the data item to be reviewed:

```
>review items vendor
```

```
REVIEW ITEMS:
```

Itm No.	Item Name	Attrib	Security
---------	-----------	--------	----------

## Changing a Data Item

Use the CHANGE ITEM command to change data item type, length, subitem count, security, and order in the data item list. In this section, the data item DESCRIPTION is changed in the ORDERS database. Before changing a data item, the REVIEW ITEMS output looks like this:

Itm No.	Item Name	Attrib	Security
1	ACCOUNT	I4	(11,12,13,14,18/)
2	BINNUM	Z2	(/13)
3	CITY	X12	(12,13,14/11)
4	CREDIT-RATING	R2	(/14)
5	DATE	X6	(11,12,13,14,18/)
6	DELIV-DATE	X6	(/14)
7	DESCRIPTION	X20	(0,11,12,13,14,18/)<---- data item to be changed
8	FIRST-NAME	X10	(14/11)
9	INITIAL	U2	(14/11)
10	LAST-NAME	X16	(14/11)
11	LASTSHIPDATE	X6	(12/)
12	ONHANDQTY	J2	(14/12)
13	PRICE	J2	(14/)
14	PURCH-DATE	X6	(11/14)
15	QUANTITY	I1	(/14)
16	STATE	X2	(12,13,14/11)
17	STOCK#	U8	(11,12,14,18/)
18	STREET-ADD	X26	(12,13,14/11)
19	SUPPLIER	X16	(12,13/)
20	STORE#	8J2	(11,12/14)
21	TAX	J2	(14/)
22	TOTAL	J2	(11,14/)
23	UNIT-COST	P8	(/12)
24	ZIP	X6	(12,13,14/11)

**Note**



Refer to Table 4-2 for information about data item type conversions supported by DBChange Plus.

In the example below, the CHANGE ITEM command is used to change the data item DESCRIPTION. It should appear before PRICE in the data item list. The subcount is changed to 4, the data type is changed to U, and the sublength is changed to 25. The

security is changed to allow read access for user classes 11 and 14 and write access for user class 18. The DBCPLUS program displays messages confirming that the changes are accepted.

```
>change item description (price) 4 u 25 (11,14/18)
Reorder of data items accepted.
```

Change of item attributes accepted.

Change of item security accepted.

## Caution



If the data item length is changed to a smaller number than the original data item length, the data may truncate or overflow the new field.

Refer to “CHANGE ITEM” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data item DESCRIPTION is changed, use the REVIEW ITEMS command. In the following example, the @ parameter is used to display all the data items in the ORDERS database. You can see that the data item DESCRIPTION now appears before PRICE in the data item list and that the attributes and security are changed.

```
>review items @
```

REVIEW ITEMS:

Itm No.	Item Name	Attrib	Security	
1	ACCOUNT	I4	(11,12,13,14,18/)	
2	BINNUM	Z2	(/13)	
3	CITY	X12	(12,13,14/11)	
4	CREDIT-RATING	R2	(/14)	
5	DATE	X6	(11,12,13,14,18/)	
6	DELIV-DATE	X6	(/14)	
7	FIRST-NAME	X10	(14/11)	
8	INITIAL	U2	(14/11)	
9	LAST-NAME	X16	(14/11)	
10	LASTSHIPDATE	X6	(12/)	
11	ONHANDQTY	J2	(14/12)	
12	DESCRIPTION	4U25	(11,14/18)	<---- data item changed
13	PRICE	J2	(14/)	
14	PURCH-DATE	X6	(11/14)	
15	QUANTITY	I1	(/14)	
16	STATE	X2	(12,13,14/11)	
17	STOCK#	U8	(11,12,14,18/)	
18	STREET-ADD	X26	(12,13,14/11)	
19	SUPPLIER	X16	(12,13/)	
20	STORE#	8J2	(11,12/14)	

21	TAX	J2	(14/)
22	TOTAL	J2	(11,14/)
23	UNIT-COST	P8	(/12)
24	ZIP	X6	(12,13,14/11)

---

## Changing Data Item Attributes

To change the subitem count, the type, and the length of a data item, use the `CHANGE ATTRIBUTES` command. `CHANGE ATTRIBUTES` and `CHANGE ITEM` are similar in that they both change data item attributes. However, `CHANGE ITEM` also changes the order of a data item in the data item list and the data item security. If you only want to change data item attributes, use the `CHANGE ATTRIBUTES` command. If you want to change the data item attributes in conjunction with data item sequence and/or security changes, use the `CHANGE ITEM` command.

In this section, the attributes of the data item `STOCK#` are changed in the `ORDERS` database. Before changing the data item attributes, the `REVIEW ITEMS` output looks like this:

`REVIEW ITEMS:`

Itm No.	Item Name	Attrib	Security
-----			
17	STOCK#	U8	(11,12,14,18/)

### Note



For information about data item type conversions supported by `DBChange Plus`, refer to Table 4-2.

In the example below, the data item `STOCK#` is changed to type `X` with a length of 10. The `DBCPLUS` program displays a message confirming that the attribute changes are accepted.

```
>change attributes stock# x 10
Change of item attributes accepted.
```

```
>
```

Refer to “`CHANGE ATTRIBUTES`” in chapter 10, “`DBChange Plus Commands`,” for a detailed description of the command syntax and parameters.

If you want to verify that the data item attributes are changed, use the `REVIEW ITEMS` command. In the following example, `STOCK#` is the name of the data item to be reviewed. You can see that the data item `STOCK#` is now defined as data type `X` with a length of 10.

```
>review items stock#
```

REVIEW ITEMS:

Itn No.	Item Name	Attrib	Security
17	STOCK#	X10	(11,12,14,18/)

## Changing Data Sets

---

This chapter explains how to change data sets in a TurboIMAGE/XL database. It provides instructions for and examples of the various commands used to change data sets. Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn to do more DBChange Plus tasks, refer to chapters 2 through 4 and 6 through 9.

If you understand DBChange Plus tasks and only want to see the command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### Summary of Commands to Change Data Sets

DBChange Plus has several commands for changing data sets in the database. Some deal with several types of data set changes, while others are specific to data set security or paths. For example, the ADD SET command deals with the data set name, its type, the order it appears in the data set list, the key and search items, and so on; while the ADD PATH command deals only with adding a path to a detail data set. This chapter covers only those commands that deal with several types of changes to a data set. For information about paths and sort items, refer to chapter 6, “Changing Paths in a Detail Data Set.” For information about changing data set security, refer to chapter 7, “Changing Database Security.” Note that some data item commands can affect fields in a data set, for instance, DELETE ITEM and RECOVER ITEM. These commands are discussed in chapter 4, “Changing Data Items,” and chapter 10, “DBChange Plus Commands.”

Table 5-1 alphabetically lists the commands that change data sets and summarizes their functions.

#### Note



- Before invoking any of the following commands, you must log on to the system as the database creator, run the DBCPLUS program, and specify the database that you want to restructure. Refer to chapter 1, “Introduction to DBChange Plus.”
  - The DBChange commands are not actually applied until the DBAPLUS program is used. Refer to chapter 9, “Applying Changes to the Database Structure.”
-

**Table 5-1. Summary of Commands to Change Data Sets**

Command	Keyword	Description
ADD	FIELD	Adds a field (data item) to a manual master or detail data set.
	PATH	Adds a path from a detail data set to a master data set (path command only).
	SET	Adds a new master or detail set to the database.
	SORT	Adds a new sort item for a search item. If a sort item already exists in the data set, the ADD SORT command replaces the existing sort item.
CHANGE	BLOCKFACTOR	Changes the blocking factor of a data set.
	BLOCKMAX	Changes the maximum physical block length of one or more data sets.
	CAPACITY	Changes the capacity of a data set.
	DEVICECLASS	Changes the device (disk) on which a data set resides.
	PRIMARYPATH	Changes the primary path for a detail data set that has at least two paths (path command only).
	SETSECURITY	Changes user class access to a data set (security command only).
DELETE	FIELD	Deletes a field from a manual master or detail data set.
	PATH	Deletes the path from a detail data set to a master data set (path command only).
	SET	Deletes a master or detail data set and its associated paths from the database.
	SETSECURITY	Deletes all security specifications from a master or detail data set (security command only).
	SORT	Deletes a sort item from a search item in a detail data set.
RECOVER	SET	Cancels a data set deletion.
RENAME	SET	Renames a data set.
REORDER	FIELD	Reorders the field sequence in a manual master or detail data set.
	SET	Reorders the data set sequence in the schema data set list.

---

## Helpful TurboIMAGE Information

Remember the following TurboIMAGE characteristics when making changes to data sets:

- A data set name must be unique within the database. It must be from 1 to 16 characters long, and the first character must be alphabetic.
- No more than 199 data sets can exist in a TurboIMAGE database.
- Data set capacity must be less than  $2^{31} - 1$  (2,147,483,647).
- The blocking factor is the number of entries in one block. The default block size is 512 words, but it may have been redefined in the schema.
- When changing the size of an entry, you should reevaluate the data set blocking factor.
- The primary path should be the path most frequently accessed in the data set. A data set can have only one primary path.
- Data sets can reside on different disks. The disk on which a data set resides is the MPE XL device identified by the **device class**. For heavily accessed databases, master and detail data sets should reside on different devices.
- A field is a data item within a data set. It must be in the data item list within the database schema.
- A search item is a field in a detail data set that defines a path to a key item in a master data set. As many as 16 different paths can be defined for each detail data set, but each search item can define only one path. A master data set can be related by its key item to as many as 16 detail data sets. Search items and key items must be simple items; that is, their subcount must equal 1.

For more information about TurboIMAGE/XL data sets, refer to the *TurboIMAGE/XL Database Management System Reference Manual*.

---

## Adding a Master Data Set

To add an automatic or manual master data set to the database, use the ADD SET command. In this section, the new data set BINNUM-MASTER is added to the ORDERS database. Before adding a data set, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
3	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
4	SUP-MASTER	M	201	12	505	DISK1	(13/12,18)
5	INVENTORY	D	450	15	481	DISK2	(12,14/13,18)
6	SALES	D	500	10	371	DISK2	(11/14,18)

In the example below, the master data set BINNUM-MASTER is added. It will appear before the INVENTORY data set in the data set list. BINNUM-MASTER is an automatic master data set. The key item is BINNUM. Note that the key item must already exist in the data item list of the database. User class 14 has read access; user classes 11 and 18 have write access. The data set capacity is 220. The blocking factor is 12, and the device class is DISK1. The DBCPLUS program displays a message confirming the addition of the new data set.

```
>add set binnum-master (inventory) a binnum (14/11,18) 220 (12) disk1
Addition of data set accepted.
```

>

Refer to “ADD SET” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

Because TurboIMAGE/XL does not allow stand-alone automatic master data sets, after adding the master data set BINNUM-MASTER, you must add a path to connect the automatic master data set to a detail data set. To do this, use the ADD PATH command. The example below adds a path from the detail data set INVENTORY. The search item is STOCK#, and BINNUM-MASTER is specified as the master data set to which the detail data set is linked. The DBCPLUS program displays a message confirming the addition of the path.

```
>add path inventory stock# binnum-master
Addition of path accepted.
```

>

After adding a master data set and a path, the new data set contains only one field—the key item. To add more fields, use the ADD FIELD command discussed later in this chapter. To connect the new key item to other detail data sets, use the ADD PATH command.

Refer to “ADD PATH” in chapter 6, “Changing Paths in a Detail Data Set,” or in chapter 10, “DBChange Plus Commands.”

If you want to verify the addition of the new data set, use the REVIEW SETS command. In the following example, the @ parameter is used to display all the data sets in the ORDERS database. You can see that the master data set BINNUM-MASTER is added.

```
>review sets @
```

```
REVIEW SETS:
```

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
3	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
4	SUP-MASTER	M	201	12	505	DISK1	(13/12,18)
5	BINNUM-MASTER	A	220	12	0	DISK1	(14/11,18) <---- *
6	INVENTORY	D	450	15	481	DISK2	(12,14/13,18)
7	SALES	D	500	10	371	DISK2	(11/14,18)

*\*data set added*

## Adding a Detail Data Set

Adding a detail data set is the same as adding a master data set except that the key item parameter is not used for a detail data set. To add a search item to a detail data set, use the ADD PATH command, which is discussed in chapter 6, “Changing Paths in a Detail Data Set.”

---

## Deleting a Data Set

To delete a master or detail data set and its associated paths from the database, use the DELETE SET command. In this section, the data set BINNUM-MASTER is deleted from the ORDERS database. Before deleting the data set, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
3	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
4	SUP-MASTER	M	201	12	505	DISK1	(13/12,18)
5	BINNUM-MASTER	A	220	12	145	DISK1	(14/11,18) <---- *
6	INVENTORY	D	450	15	640	DISK2	(12,14/13,18)
7	SALES	D	500	10	371	DISK2	(11/14,18)

\* data set to be deleted

In the example below, the data set BINNUM-MASTER is deleted. In an interactive session, a message is displayed asking you to confirm the deletion of the data set. In batch mode (as shown below), the data set is deleted without confirmation. The DBCPLUS program displays a message confirming the deletion of the data set.

```
>delete set binnum-master
Deletion of data set accepted.

>
```

---

### Note



If you delete all detail data sets that are connected to an automatic master data set, the automatic master data set to which they are connected is also deleted.

---

Refer to “DELETE SET” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify the deletion of the data set, use the REVIEW SETS command. In the following example, the @ parameter is used to display all the data sets in the ORDERS database. You can see that the data set BINNUM-MASTER is deleted.

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
3	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
4	SUP-MASTER	M	201	12	505	DISK1	(13/12,18) <----- *
5	INVENTORY	D	450	15	640	DISK2	(12,14/13,18)
6	SALES	D	500	10	371	DISK2	(11/14,18)

\*data set deleted

## Recovering a Data Set

In the previous example, you deleted the data set BINNUM-MASTER from the ORDERS database. To recover it, use the RECOVER SET command.

The position in the data set list where the recovered data set is placed depends on whether the original data set was actually part of the database or whether it was only stored in the change file. (A data set is only stored in the change file when it has been added by the DBCPLUS program but DBAPLUS has not yet been used to apply it to the database.) If the DBAPLUS program has not yet been run, then the recovered data set is inserted at the end of the data set list. If the original data set was actually part of the database, then it is reinserted in its original place in the data set list. (If you need to change the order of the data sets, use the REORDER SET command. Refer to “Reordering a Data Set” later in this chapter.)

### Note



- Once you run the DBAPLUS program, you cannot recover a deleted data set.
- If you delete a data set and then add a new data set with the same name, you cannot recover the original data set.

In the following example, the data set BINNUM-MASTER is recovered. The DBCPLUS program returns a message confirming recovery of the deleted data set.

```
>recover set binnum-master
```

Recovery of data set accepted.

>

Refer to “RECOVER SET” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify the recovery of the deleted data set, use the REVIEW SETS command. In the following example, the @ parameter is used to display all the data sets in the ORDERS database. Because the data set BINNUM-MASTER was in the change file and the DBAPLUS program had not been run, the data set is inserted at the end of the data set list. The path connecting the master data set BINNUM-MASTER to the detail data set INVENTORY cannot be recovered. You must specifically add a path back in to the data set using the ADD PATH command.

```
>review sets @
```

```
REVIEW SETS:
```

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
3	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
4	SUP-MASTER	M	201	12	505	DISK1	(13/12,18)
5	INVENTORY	D	450	15	481	DISK2	(12,14/13,18)
6	SALES	D	500	10	371	DISK2	(11/14,18)
7	BINNUM-MASTER	A	220	12	0	DISK1	(14/11,18) <---- *

No paths defined for auto master set (DBC 548).  
No fields defined for data set (DBC 547).

*\*data set recovered*

## Reordering a Data Set

To reorder the sequence of data sets in the data set list, use the REORDER SET command. In this section, the data set BINNUM-MASTER in the ORDERS database is reordered. Before reordering the data set, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
3	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
4	SUP-MASTER	M	201	12	505	DISK1	(13/12,18)

5	INVENTORY	D	450	15	481	DISK2	(12,14/13,18)
6	SALES	D	500	10	371	DISK2	(11/14,18)
7	BINNUM-MASTER	A	220	12	0	DISK1	(14/11,18) <---- *

\* *data set to be reordered*

In the example below, the data set BINNUM-MASTER is reordered to appear before the data set CUSTOMER in the data set list. (When the *nextset* parameter is omitted, the data set appears at the end of the data set list.) The DBCPLUS program displays a message confirming that the data sets are reordered.

```
>reorder set binnun-master customer
Reorder of data set accepted.
```

>

Refer to “REORDER SET” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

To verify that the data sets are reordered, use the REVIEW SETS command. In the following example, the @ parameter is used to display all the data sets in the ORDERS database. You can see that the data set BINNUM-MASTER now appears before the data set CUSTOMER in the data set list.

```
>review sets @
```

REVIEW SETS:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
1	DATE-MASTER	A	365	19	496	DISK1	
2	BINNUM-MASTER	A	220	12	0	DISK1	(14/11,18) <----- *
3	CUSTOMER	M	201	10	640	DISK1	(14/11,18)
4	PRODUCT	M	300	16	498	DISK1	(13,14/12,18)
5	SUP-MASTER	M	201	12	505	DISK1	(13/12,18)
6	INVENTORY	D	450	15	481	DISK2	(12,14/13,18)
7	SALES	D	500	10	371	DISK2	(11/14,18)

\*data set reordered

## Renaming a Data Set

To give a new name to a data set, use the RENAME SET command. In this section, the detail data set SALES in the ORDERS database is renamed. Before renaming the data set, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
7	SALES	D	500	10	1024	DISK2	(11/14,18) <---- *

*\* data set to be renamed*

In the following example, the data set SALES is renamed to LOCAL-SALES. The DBCPLUS program displays a message confirming that the data set is renamed.

```
>rename set sales local-sales
Rename of data set accepted.
```

>

Refer to “RENAME SET” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the SALES data set has been renamed, use the REVIEW SETS command. You can see that the new data set name is LOCAL-SALES.

```
>review sets local-sales
```

REVIEW SETS:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
7	LOCAL-SALES	D	500	10	1024	DISK2	(11/14,18) <---- *

*\* data set renamed*

---

## Changing the Capacity of a Data Set

To change the capacity of a data set, use the `CHANGE CAPACITY` command.

When you change the capacity of a data set, the DBCPLUS program checks to see if the new capacity is greater than or less than the **high-water mark**. The high-water mark is the greatest number of entries the data set has ever contained.

If the new capacity is greater than the high-water mark, the capacity is changed as specified. If the new capacity is less than the high-water mark, use the `REPACK SET` command to change the capacity. To learn more about repacking data sets, refer to chapter 8, “Maintaining the Database,” or “`REPACK SET`” in chapter 10, “DBChange Plus Commands.”

To automatically adjust data set capacities, use the `CONTROL` command with the `NEWFULL` parameter. Refer to “`CONTROL`” in chapter 10, “DBChange Plus Commands.”

In this section, the capacity of the detail data set `INVENTORY` is changed. Before changing the capacity, the `REVIEW SETS` command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
6	INVENTORY	D	450	15	640	DISK2	(12,14/13,18)<---- *

*\*capacity to be changed*

In the example below, the capacity of the detail data set `INVENTORY` is changed from 450 to 510. The DBCPLUS program displays a message confirming that the change of capacity is accepted.

```
>change capacity inventory 510
Change of set capacity, block factor, and/or device class accepted.
>
```

Refer to “`CHANGE CAPACITY`” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

### Note



The capacity must be a number that can be evenly divided by the blocking factor. If you change the capacity to a number that cannot be evenly divided by the blocking factor, DBSCHEMA (called by the DBAPLUS program) adjusts the specified capacity to the nearest multiple of the blocking factor. For example, if you specify a capacity of 503, but the blocking factor is 15, the capacity is automatically

changed to 510, which can be evenly divided by 15. Note, however, that if the REVIEW SETS or REVIEW BLOCKS command is used before invoking the DBAPLUS program, the specified capacity, rather than the actual adjusted capacity, is displayed.

---

If you want to verify that the capacity of the data set INVENTORY is changed, use the REVIEW SETS command as follows. You can see that the capacity is now 510.

```
>review sets inventory
```

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
INVENTORY	D	7	4	24	40	510	15	601	640 <----- *

*\*capacity changed*

## Changing the Blocking Factor of a Data Set

To change the blocking factor of a data set, use the CHANGE BLOCKFACTOR command. In this section, the blocking factor is changed for the data set INVENTORY. Before changing the blocking factor, the REVIEW BLOCKS command output looks like this:

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
INVENTORY	D	7	4	24	40	510	15	481	1024 <----- *

*\*blocking factor to be changed*

In the example below, the blocking factor for the detail data set INVENTORY is changed from 15 to 12. The DBCPLUS program displays a message confirming that the blocking factor is changed.

```
>change blockfactor inventory 12
```

```
Change of set capacity, block factor, and/or device class accepted.
```

```
>
```

Refer to “CHANGE BLOCKFACTOR” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

### Note



The blocking factor must be a number that can be evenly divided into the capacity. If you change the blocking factor to a number that cannot be evenly divided into the capacity, DBChange Plus will

adjust the capacity to a multiple of the new blocking factor closest to the current capacity.

---

If you want to verify that the blocking factor is changed, use the REVIEW BLOCKS command as in the following example. You can see that the blocking factor for the data set INVENTORY is now 12.

```
>review blocks inventory
```

```
REVIEW BLOCKS:
```

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
INVENTORY	D	7	4	24	40	510	12	481	481 <---- *

*\*blocking factor changed*

## Changing the Maximum Block Length of a Data Set

To change the maximum physical block length of a data set, use the CHANGE BLOCKMAX command. The block length is in words. In this section, the maximum block length of all the data sets in the ORDERS database is changed. Before changing the maximum block length, the REVIEW BLOCKS command output looks like this:

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
DATE-MASTER	A	1	3	3	26	365	19	496	496<---- *
BINNUM-MASTER	A	1	1	1	12	220	12	145	145
CUSTOMER	M	8	1	41	52	201	10	521	640
PRODUCT	M	2	2	14	31	300	16	498	498
SUP-MASTER	M	5	1	31	42	201	12	505	505
INVENTORY	D	7	4	24	40	510	12	481	481
SALES	D	8	4	21	37	500	10	371	371

*\* maximum block length for all data sets to be changed*

In the example below, the maximum block length for all the data sets is changed by specifying the first and last data sets in the range. DATE-MASTER is the first data set in the range; SALES is the last data set in the range. The maximum block length changes for all the data sets between DATE-MASTER and SALES in the data set list. The new maximum block length for all the data sets is 1024. The DBCPLUS program displays a message confirming that the data set block length has been changed.

```
>change blockmax date-master sales 1024  
Change of maximum blocksize accepted.
```

```
>
```

**Note**


---

The range specified must reflect the current order of the data sets; that is, if the REORDER SET command was used prior to the CHANGE BLOCKMAX command, the range of the data sets must reflect the new order. To verify the data set order, use the REVIEW BLOCKS command.

---

Refer to “CHANGE BLOCKMAX” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

You can change the maximum block length for a single data set by using only one data set name in the command syntax. If you want to change the maximum block length for multiple data sets, but don’t want to specify a range, you must issue the CHANGE BLOCKMAX command for each data set.

If you want to verify that the maximum block length is changed, use the REVIEW BLOCKS command as shown below. The @ parameter is used to display all the data sets in the ORDERS database. You can see that all the data sets now have a block length of 1024.

```
>review blocks @
```

```
REVIEW BLOCKS:
```

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
DATE-MASTER	A	1	3	3	26	365	19	496	1024 <----- *
BINNUM-MASTER	A	1	1	1	12	220	12	145	1024
CUSTOMER	M	8	1	41	52	201	10	521	1024
PRODUCT	M	2	2	14	31	300	16	498	1024
SUP-MASTER	M	5	1	31	42	201	12	505	1024
INVENTORY	D	7	4	24	40	510	12	481	1024
SALES	D	8	4	21	37	500	10	371	1024

*\*maximum block length for all data sets changed*

## Changing the Device Class of a Data Set

To change the device on which a data set resides, use the CHANGE DEVICECLASS command. The CHANGE DEVICECLASS command actually moves the data set. In this section, a different device class is specified for the data set SUP-MASTER in the ORDERS database. Before changing the device class, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
5	SUP-MASTER	M	201	12	1024	DISK1	(13/12,18) <---- *

*\* device class to be changed*

In the example below, the device class is changed from DISK1 to DISK2. The DBCPLUS program displays a message confirming that the device class has been changed.

```
>change deviceclass sup-master disk2
Change of set capacity, block factor, and/or device class accepted.
```

```
>
```

Refer to “CHANGE DEVICECLASS” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the device class for the SUP-MASTER data set is changed, use the REVIEW SETS command as follows. You can see that the device class is now DISC2.

```
>review sets sup-master
```

```
REVIEW SETS:
```

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
5	SUP-MASTER	M	201	12	1024	DISK2	(13/12,18) <---- *

*\* device class changed*

## Adding a Field to a Data Set

To add a field to a manual master or detail data set, use the ADD FIELD command. In this section, the field STORE# is added to the INVENTORY data set in the ORDERS database. Before adding the field, the REVIEW FIELDS output like this:

```
Data Set No.: 6   Name:  INVENTORY   Type:  D

Fld No. Field Name      Pr Sr So  Master Set Name  Sort Item Name
-----
1       STOCK#          Y      PRODUCT
2       ONHANDQTY
3       SUPPLIER        Y  Y      SUP-MASTER <---- field to be added here
4       UNIT-COST
5       LASTSHIPDATE    Y      DATE-MASTER
6       BINNUM          Y      BINNUM-MASTER
```

In the example below, the field STORE# is added to the INVENTORY data set. It should appear before the field SUPPLIER in the field list of the data set. When adding a field to a data set, the new field must already exist as a data item in either the current change file or in the database. The DBCPLUS program displays a message confirming the addition of the new field.

```
>add field inventory store# supplier
Addition of field accepted.
```

>

Refer to “ADD FIELD” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax.

To add a search item to a detail data set, use the ADD PATH command, which is discussed in chapter 6, “Changing Paths in a Detail Data Set,” and chapter 10, “DBChange Plus Commands.”

If you want to verify the addition of the new field, use the REVIEW FIELDS command. In the following example, you can see that the field STORE# is added to the data set INVENTORY:

```
>review fields inventory
```

```
REVIEW FIELDS (DETAIL SET):
```

```
Data Set No.: 6   Name:  INVENTORY   Type:  D

Fld No. Field Name      Pr Sr So  Master Set Name  Sort Item Name
-----
1       STOCK#          Y      PRODUCT
```

```

2      ONHANDQTY
3      STORE#                               <---- field added
4      SUPPLIER          Y  Y          SUP-MASTER
5      UNIT-COST
6      LASTSHIPDATE          Y          DATE-MASTER
7      BINNUM              Y          BINNUM-MASTER

```

## Deleting a Field from a Data Set

To delete a field from a manual master or detail data set, use the DELETE FIELD command. In this section, the field CREDIT-RATING is deleted from the master data set CUSTOMER in the ORDERS database. Before deleting the field, the REVIEW FIELDS output looks like this:

```
Data Set No.: 3   Name: CUSTOMER   Type: M
```

Fld No.	Field Name	Detail Set Name	Path's Search Item
1	ACCOUNT	SALES	ACCOUNT
2	LAST-NAME		
3	FIRST-NAME		
4	INITIAL		
5	STREET-ADD		
6	CITY		
7	STATE		
8	ZIP		
9	CREDIT-RATING		<---- field to be deleted

In the example below, the field CREDIT-RATING is deleted from the data set CUSTOMER. In an interactive session, a message is displayed asking you to confirm the deletion of the field. In batch mode, the data set is deleted without confirmation. If you attempt to delete a field that is also a search item, you must first delete the path. If you attempt to delete a field that is a sort item in any data set, you must first delete the sort item. Refer to chapter 6, "Changing Paths in a Detail Data Set," for more information. The DBCPLUS program displays a message confirming that the field is deleted.

```
>delete field customer credit-rating
Deletion of field accepted.
```

```
>
```

Refer to "DELETE FIELD" in chapter 10, "DBChange Plus Commands," for a detailed description of the command syntax and parameters.

**Caution**

---

Once you have deleted a field, it cannot be recovered from the change file; you must use the `ADD FIELD` command to add it back in to the data set. However, if the field contained data before it was deleted, the data cannot be recovered.

---

If you want to verify the deletion of the field, use the REVIEW FIELDS command. In the following example, you can see that the field CREDIT-RATING no longer appears in the data set CUSTOMER field list:

```
>review fields customer
```

```
REVIEW FIELDS (MASTER SET):
```

```
Data Set No.: 3 Name: CUSTOMER Type: M
```

```
Fld No. Field Name Detail Set Name Path's Search Item
```

```
-----
```

```
1 ACCOUNT SALES ACCOUNT
2 LAST-NAME
3 FIRST-NAME
4 INITIAL
5 STREET-ADD
6 CITY
7 STATE
8 ZIP
```

```
<---- field deleted
```

## Reordering a Field in a Data Set

To reorder the field list sequence in a manual master or detail data set, use the REORDER FIELD command. In this section, the fields in the data set INVENTORY in the ORDERS database are reordered. Before reordering, the REVIEW FIELDS output looks like this:

```
Data Set No.: 6 Name: INVENTORY Type: D
```

```
Fld No. Field Name Pr Sr So Master Set Name Sort Item Name
```

```
-----
```

```
1 STOCK# Y PRODUCT
2 ONHANDQTY
3 STORE#
4 SUPPLIER Y Y SUP-MASTER
5 UNIT-COST
6 LASTSHIPDATE Y DATE-MASTER
7 BINNUM Y BINNUM-MASTER
```

```
<---- field to be reordered
```

In the example below, the field BINNUM is reordered to appear before the field ONHANDQTY in the field list. The DBCPLUS program displays a message confirming that the fields are reordered.

```
>reorder field inventory binnum onhandqty
Reorder of fields accepted.
```

>

To verify that the fields in the INVENTORY data set are reordered, use the REVIEW FIELDS command as shown below. You can see that the field BINNUM now appears before the field ONHANDQTY in the field list.

```
>review fields inventory
```

```
REVIEW FIELDS (DETAIL SET):
```

```
Data Set No.: 6 Name: INVENTORY Type: D
```

Fld No.	Field Name	Pr	Sr	So	Master Set Name	Sort Item Name
1	STOCK#			Y	PRODUCT	
2	BINNUM			Y	BINNUM-MASTER	<---- field reordered
3	ONHANDQTY					
4	STORE#					
5	SUPPLIER	Y	Y		SUP-MASTER	
6	UNIT-COST					
7	LASTSHIPDATE			Y	DATE-MASTER	

## Adding a Sort Item to a Detail Data Set

To add a sort item to a search item in a detail data set, use the ADD SORT command. The sort item must already be a field in the data set. In this section, a new sort item is added to the INVENTORY detail data set of the ORDERS database. Before adding a sort item, the REVIEW PATHS command output looks like this:

```
DATA SET NO: 6 NAME: INVENTORY TYPE: D
```

SEARCH ITEM NAME	MASTER SET NAME	SORT ITEM NAME	PRIMARY
STOCK#	PRODUCT		
SUPPLIER	SUP-MASTER		Y
LASTSHIPDATE	DATE-MASTER		

>

In the example below, INVENTORY is the name of the detail data set that will contain the sort item. SUPPLIER is the name of the search item whose like values will be sorted based on the new sort item values. LASTSHIPDATE is the name of the new sort item. The DBCPLUS program displays a message confirming the addition of the new sort item.

```
>add sort inventory supplier lastshipdate
   New sort item added.
```

>

Refer to “ADD SORT” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify the addition of the sort item, use the REVIEW PATHS command. In the following example, you can see that LASTSHIPDATE is now the sort item for the search item SUPPLIER:

```
>review paths inventory
```

```
DATABASE NAME:  ORDERS
```

```
REVIEW PATHS (FOR DETAIL DATA SET):
```

```
DATA SET NO:  6  NAME:  INVENTORY  TYPE:  D

SEARCH ITEM NAME  MASTER SET NAME  SORT ITEM NAME  PRIMARY
=====
STOCK#           PRODUCT
SUPPLIER         SUP-MASTER     LASTSHIPDATE    Y  <---- *
LASTSHIPDATE    DATE-MASTER
```

*\*LASTSHIPDATE is now the sort item for the search item SUPPLIER.*

>

## Deleting a Sort Item from a Detail Data Set

To remove a sort item from a search item in a detail data set, use the DELETE SORT command. In this section, a sort item is removed from the search item SUPPLIER in the detail data set INVENTORY. Before deleting the sort item, the REVIEW PATHS command output looks like this:

```

DATA SET NO: 6   NAME: INVENTORY   TYPE: D

SEARCH ITEM NAME   MASTER SET NAME   SORT ITEM NAME   PRIMARY
=====
STOCK#             PRODUCT                               Y
SUPPLIER           SUP-MASTER       LASTSHIPDATE
LASTSHIPDATE      DATE-MASTER

```

>

In the example below, INVENTORY is the name of the data set containing the sort item to be deleted. SUPPLIER is the name of the search item containing the sort item to be deleted. This command does not delete the LASTSHIPDATE field from the data set; it only removes its role as a sort item. The DBCPLUS program displays a message confirming the deletion of the sort item.

```

>delete sort inventory supplier
Deletion of sort item accepted.

```

>

Refer to “DELETE SORT” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify the deletion of the sort item, use the REVIEW PATHS command. In the following example, you can see that LASTSHIPDATE sort item is deleted:

```

>review paths inventory

```

```

DATABASE NAME: ORDERS

```

```

REVIEW PATHS (FOR DETAIL DATA SET):

```

```

DATA SET NO: 6   NAME: INVENTORY   TYPE: D

SEARCH ITEM NAME   MASTER SET NAME   SORT ITEM NAME   PRIMARY
=====
STOCK#             PRODUCT                               Y
SUPPLIER           SUP-MASTER

```

<---- \*

LASTSHIPDATE      DATE-MASTER

*\* The sort item LASTSHIPDATE has been deleted from the data set.*

>

## Changing Paths in a Database

---

This chapter explains how to change paths in a TurboIMAGE/XL database. It provides instructions for and examples of the commands used to change paths. Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn how to do more DBChange Plus tasks, refer to chapters 2 through 5 and 7 through 9. If you understand DBChange Plus tasks and only want to see command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### Summary of Commands to Change Paths

DBChange Plus provides specific commands to change paths in a database. Table 6-1 alphabetically lists the commands that change paths and summarizes their functions:

**Table 6-1. Summary of Commands to Change Paths**

Command	Keyword	Description
ADD	PATH	Adds a path from a detail data set to a master data set.
CHANGE	PRIMARYPATH	Redefines the primary path of a detail data set with at least two paths.
DELETE	PATH	Deletes a path from a detail data set.

#### Note



- Before invoking any of the following commands, you must log on to the system as the database creator, run the DBCPLUS program, and specify the database for which you want to change paths. Refer to chapter 1, “Introduction to DBChange Plus.”
  - The commands are not actually applied until the DBAPLUS program is used. Refer to chapter 9, “Applying Changes to the Database Structure.”
  - Paths can only be added, redefined, or deleted from a detail data set. For example, paths to a manual master data set must be added through the related detail data set.
-

---

## Helpful TurboIMAGE/XL Information

Remember the following TurboIMAGE/XL characteristics when changing paths:

- A search item is a data item in a detail data set that defines a path to a related master data set. As many as 16 different paths can be defined for each detail data set, but each search item can define only one path. A master data set can be related to as many as 16 detail data sets. A search item must be a simple item (subitem count equals 1).
- A sort item must be a data item of type U, K, or X. The same item cannot be both a search item and a sort item for the same path.
- The primary path should be the most frequently accessed path in the data set. A data set can have only one primary path.
- Master key items and detail search items must be the same subtype and sublength.
- Only one key item is allowed for each master data set.

For more information, refer to the *TurboIMAGE/XL Database Management System Reference Manual*.

---

## Adding a Path to a Detail Data Set

To add a path linking a detail data set to a master data set, use the ADD PATH command. In this section, a new path is added to the SALES detail data set of the ORDERS database. Before adding a path, the REVIEW PATHS command output looks like this:

```
DATA SET NO: 4   NAME: SALES   TYPE: D

SEARCH ITEM NAME   MASTER SET NAME   SORT ITEM NAME   PRIMARY
=====
ACCOUNT            CUSTOMER          PURCH-DATE      Y
STOCK#             PRODUCT
PURCH-DATE         DATE-MASTER
DELIV-DATE         DATE-MASTER
```

>

### Note



---

When adding a path, remember that the master data set must appear before the detail data set in the data set list in the schema. To do this, use the REORDER SET command. Refer to chapter 10, “DBChange Plus Commands,” for information about the REORDER SET command.

---

In the example below, SALES is the name of the detail data set that will contain the new search item. FILL-DATE is the name of the search item for the new path. The exclamation point (!) indicates that the new path is to be the primary path. DATE-MASTER is the name of the master data set containing the key item to be related to the search item. STOCK# is the name of the sort item for the new path. DBChange returns a message confirming the addition of the new path.

```
>add path sales fill-date !date-master stock#
Addition of path accepted.
```

```
>
```

Refer to “ADD PATH” in chapter 10, “DBChange Plus Commands” for a detailed description of the command syntax and parameters.

If you want to verify the addition of the path, use the REVIEW PATHS command. In the following example, you can see that FILL-DATE is now the search item for the new primary path to the data set DATE-MASTER:

```
>review paths sales
```

```
DATABASE NAME:  ORDERS
```

```
REVIEW PATHS (FOR DETAIL DATA SET):
```

```
DATA SET NO:  4  NAME:  SALES  TYPE:  D
```

SEARCH ITEM NAME =====	MASTER SET NAME =====	SORT ITEM NAME =====	PRIMARY =====
ACCOUNT	CUSTOMER	PURCH-DATE	
STOCK#	PRODUCT		
PURCH-DATE	DATE-MASTER		
DELIV-DATE	DATE-MASTER		
FILL-DATE	DATE-MASTER	STOCK#	Y <---- *

\* *FILL-DATE is now the search item for the primary path to the DATE-MASTER data set. STOCK# is the sort item.*

---

## Redefining the Primary Path in a Detail Data Set

To redefine the primary path in a detail data set containing two or more paths, use the `CHANGE PRIMARYPATH` command. In this section, a new primary path is redefined in the `INVENTORY` detail data set of the `ORDERS` database. Before redefining a primary path, the `REVIEW PATHS` command output looks like this:

```
DATA SET NO: 6  NAME:  INVENTORY  TYPE:  D

SEARCH ITEM NAME  MASTER SET NAME  SORT ITEM NAME  PRIMARY
=====
STOCK#           PRODUCT
SUPPLIER         SUP-MASTER     LASTSHIPDATE    Y <---- *
LASTSHIPDATE    DATE-MASTER
```

>

*\*Supplier is currently the search item for the primary path.*

In the example below, `INVENTORY` is the name of the detail data set containing the primary path to be redefined. `STOCK#` is the search item for the new primary path. Note that the search item must already be an existing data item in the data set. The `DBCPLUS` program displays a message confirming redefinition of the primary path.

```
>change primarypath inventory stock#
Redefinition of primary path accepted.
```

>

Refer to “`CHANGE PRIMARYPATH`” in chapter 10, “`DBChange Plus Commands`,” for a detailed description of the command syntax and parameters.

If you want to verify that the primary path is changed, use the `REVIEW PATHS` command. In the following example, you can see that the primary path is changed to the path whose search item is `STOCK#`:

>review paths inventory

DATABASE NAME: ORDERS

REVIEW PATHS (FOR DETAIL DATA SET):

```
DATA SET NO: 6  NAME: INVENTORY  TYPE: D

SEARCH ITEM NAME  MASTER SET NAME  SORT ITEM NAME  PRIMARY
=====
STOCK#           PRODUCT                Y <----- *
SUPPLIER         SUP-MASTER          LASTSHIPDATE
LASTSHIPDATE    DATE-MASTER
```

*\*STOCK# is now the search item for the primary path.*

>

---

## Deleting a Path from a Detail Data Set

To delete a path from a detail data set, use the DELETE PATH command. In this section, the path associated with the FILL-DATE search item is deleted from the SALES detail data set in the ORDERS database. Note that the search item is not actually deleted from the data set; however, it is no longer a search item. Before deleting the path, the REVIEW PATHS command output looks like this:

```
DATA SET NO: 4  NAME: SALES  TYPE: D

SEARCH ITEM NAME  MASTER SET NAME  SORT ITEM NAME  PRIMARY
=====
ACCOUNT          CUSTOMER          PURCH-DATE
STOCK#           PRODUCT
PURCH-DATE      DATE-MASTER
DELIV-DATE      DATE-MASTER
FILL-DATE       DATE-MASTER      STOCK#          Y <----- *
```

*\*The path associated with the search item FILL-DATE will be deleted.*

>

In the example below, SALES is the name of the data set containing the path to be deleted. FILL-DATE is the search item associated with the path to be deleted. DBChange displays a message confirming the deletion of the path.

```
>delete path sales fill-date
Deletion of path accepted.
```

```
>
```

Refer to “DELETE PATH” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify the deletion of a path, use the REVIEW PATHS command. In the following example, you can see that the path associated with the FILL-DATE search item is deleted:

```
>review paths sales
```

```
DATABASE NAME:  ORDERS
```

```
REVIEW PATHS (FOR DETAIL DATA SET):
```

```
DATA SET NO:  4   NAME:  SALES   TYPE:  D
```

SEARCH ITEM NAME =====	MASTER SET NAME =====	SORT ITEM NAME =====	PRIMARY =====
ACCOUNT	CUSTOMER	PURCH-DATE	Y
STOCK#	PRODUCT		
PURCH-DATE	DATE-MASTER		
DELIV-DATE	DATE-MASTER		

```
<---- *
```

\* *The path associated with the search item FILL-DATE has been deleted from the data set.*

```
>
```

## Note



---

After deleting a primary path, DBChange Plus automatically designates the path associated with the first search key in the detail data set as the new primary path. To change this, use the CHANGE PRIMARYPATH command discussed earlier in this chapter.

---

## Changing Database Security

---

This chapter explains how to change user access to a TurboIMAGE/XL database. It provides instructions for and examples of the commands used to change user access. Before reading this chapter, you should be familiar with the information in the chapter 1, “Introduction to DBChange Plus.” To learn how to do more DBChange Plus tasks, refer to chapters 2 through 6, 8, and 9. If you understand DBChange Plus tasks and only want to see command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### Summary of Commands to Change Database Security

DBChange Plus provides specific commands to change and review database security. Some deal only with database security, while others are multipurpose commands. For example, `CHANGE ITEMSECURITY` only makes changes to data item security; while `CHANGE ITEM` includes other data item changes in addition to data item security. If you are making several changes to a data item, then `CHANGE ITEM` is a more efficient command than `CHANGE ITEMSECURITY`. However, if you are making only data item security changes, then `CHANGE ITEMSECURITY` is a better choice. This chapter discusses only those commands that deal specifically with security. For more information about how to use multipurpose commands, refer to chapter 4, “Changing Data Items,” or chapter 5, “Changing Data Sets.”

Table 7-1 alphabetically lists all DBChange commands that involve database security and summarizes their functions.

#### Note



- Before invoking any of the following commands, you must log on to the system as the database creator, run the `DBCPLUS` program, and specify the database for which you want to change security. Refer to chapter 1, “Introduction to DBChange Plus.”
  - The commands are not actually applied until the `DBAPLUS` program is used. Refer to chapter 9, “Applying Changes to the Database Structure.”
-

**Table 7-1. Summary of Commands to Change Database Security**

Command	Keyword	Description
ADD	ITEM	Adds a data item to the database. (Multipurpose command.)
	PASSWORD	Adds a password and user class number to the database.
	SET	Adds a new master or detail data set to the database. (Multipurpose command.)
CHANGE	ITEM	Changes the type, length, subcount, security, and/or order of a data item. (Multipurpose command.)
	ITEMSECURITY	Changes user class access to a data item.
	PASSWORD	Changes a password.
	SETSECURITY	Changes user class access to a data set.
DELETE	ITEM	Deletes a data item and all associated fields and paths from the database.
	ITEMSECURITY	Deletes all security specifications from a data item.
	PASSWORD	Deletes a password and its associated user class from the database.
	SET	Deletes a data set and all associated paths from the database. (Multipurpose command.)
	SETSECURITY	Deletes all security specifications from a data set.
PRINT	SCHEMA	Prints the database schema including changes. (Multipurpose command.)
RECOVER	ITEM	Cancels a data item deletion. (Multipurpose command.)
	SET	Cancels a data set deletion. (Multipurpose command.)

---

## Helpful TurboIMAGE/XL Information

Remember the following TurboIMAGE/XL characteristics when making changes to the database security:

- A password is associated with a user class number from 1 to 63 in the schema. This number is used in the schema read/write lists to define data item and data set access for the password.
- A password must be from 1 to 8 ASCII characters, excluding carriage returns, slashes, semicolons, and blanks. Passwords, unlike other TurboIMAGE/XL elements, are case sensitive. This means that the password ship-rec is not the same as the password SHIP-REC.
- The user class number and the password must be unique for the database.
- If a user does not enter a password or enters an invalid password, TurboIMAGE/XL assigns user class 0. By default, user class 0 is assigned read-only access to any data items or data sets that do not have specific read/write lists.
- If write access is assigned, read access is assumed. This is true for both data items and data sets.
- If write access is assigned to a data set, read and write access is given to all items in the set. Data item security is not checked.
- If read access is assigned to a data set, data item security is checked to determine access to items in the set. If write access is assigned at the item level, updates are allowed to the data item. Otherwise, read access is granted.
- If the user class is omitted at the item level, no access is allowed.

For more information about database security, refer to the *TurboIMAGE/XL Database Management System Reference Manual*.

---

## Adding a Password to the Database

To add a password and its associated user class to the database, use the ADD PASSWORD command. In this section, a new password is added to the ORDERS database. Before adding a password, the REVIEW PASSWORDS command output looks like this:

User Class	Password
11	CREDIT
12	BUYER
13	SHIP-REC
14	CLERK
18	DO-ALL

>

In the example below, the password `MANAGER` and its associated user class 15 are added to the `ORDERS` database. (Remember that passwords are case sensitive.) The `DBCPLUS` program returns a message confirming the addition of the user class and password.

```
>add password MANAGER 15
      Addition of user class and password accepted.
```

```
>
```

Refer to “`ADD PASSWORD`” in chapter 10, “`DBChange Plus Commands`,” for a detailed description of the command syntax and parameters.

If you want to verify that the password and user class are entered correctly, use the `REVIEW PASSWORDS` command. In the following example, you can see that user class 15 and its associated password `MANAGER` are added to the `ORDERS` database:

```
>review passwords

REVIEW PASSWORDS:

      User Class      Password
      -----
      11              CREDIT
      12              BUYER
      13              SHIP-REC
      14              CLERK
      15              MANAGER      <---- *
      18              DO-ALL
```

\* *The new password `MANAGER` and its associated user class 15 have been added and are automatically inserted in numerical order by user class in the password list.*

---

## Changing a Password in the Database

To change a password, use the `CHANGE PASSWORD` command. In this section, a password is changed in the `ORDERS` database. Before changing a password, the `REVIEW PASSWORDS` command output looks like this:

```
      User Class      Password
      -----
      11              CREDIT
      12              BUYER
      13              SHIP-REC
      14              CLERK
      15              MANAGER
      18              DO-ALL
```

>

In the example below, the password CLERK is changed to the password SALESREP. The DBCPLUS program returns a message confirming the change of the password.

```
>change password CLERK SALESREP
Change of password accepted.
```

>

Refer to “CHANGE PASSWORD” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify the password change, use the REVIEW PASSWORDS command. In the following example, you can see that the password is changed:

```
>review passwords @
```

REVIEW PASSWORDS:

User Class	Password	
11	CREDIT	
12	BUYER	
13	SHIP-REC	
14	SALESREP	<---- *
15	MANAGER	
18	DO-ALL	

*\*The password CLERK is changed to the password SALESREP.*

---

## Deleting a Password from the Database

To delete a password and its associated user class from the database, use the DELETE PASSWORD command. In this section, a password and its associated user class are deleted from the ORDERS database. Before deleting a password, the REVIEW PASSWORDS command output looks like this:

User Class	Password	
11	CREDIT	
12	BUYER	
13	SHIP-REC	
14	SALESREP	
15	MANAGER	
18	DO-ALL	

>

In the example below, the password SHIP-REC and its associated user class 13 are deleted from the ORDERS database. The DBCPLUS program asks if you want the associated user class deleted. In this example, Y is entered for yes. The DBCPLUS program then returns a message confirming the deletion of the user class and password.

```
>delete password SHIP-REC 13
Delete password user class [y/N] ? y
Deletion of user class and password accepted.
```

>

Refer to “DELETE PASSWORD” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the password and user class are deleted, use the REVIEW PASSWORDS command. In the following example, you can see that the password SHIP-REC and its associated user class are deleted from the ORDERS database:

```
>review passwords
```

```
REVIEW PASSWORDS:
```

User Class	Password	
11	CREDIT	
12	BUYER	
14	SALESREP	<---- *
15	MANAGER	
18	DO-ALL	

>

*\*The password SHIP-REC and its associated user class 13 have been deleted.*

---

## Changing Data Item Security

To change data item security, use the `CHANGE ITEMSECURITY` command. `CHANGE ITEMSECURITY` changes only one user class access at a time. If you want to change more than one user class at a time, use the `CHANGE ITEM` command. Refer to “CHANGE ITEM” in chapter 10, “DBChange Plus Commands.”

In this section, the security of the data item `QUANTITY` is changed in the `ORDERS` database. Before making changes, the `REVIEW ITEMS` command output looks like this:

Itm No.	Item Name	Attrib	Security
15	QUANTITY	I1	(/14)

>

In the example below, the security of the data item QUANTITY is changed to add user class 18 to the write access class list. The DBCPLUS program returns a message confirming the change of security.

```
>change itemsecurity quantity 18 w  
Change of item security accepted.
```

>

Refer to “CHANGE ITEMSECURITY” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the item security is changed, use the REVIEW ITEMS command. In the following example, you can see that the security for the data item QUANTITY is changed:

```
>review items quantity
```

REVIEW ITEMS:

Itm No.	Item Name	Attrib	Security	
15	QUANTITY	I1	(/14,18)	<---- *

>

\* *The security of the data item QUANTITY has been changed to put user class 18 in the write access class list.*

---

## Deleting Data Item Security

To delete all security specifications from a data item, use the DELETE ITEMSECURITY command. This results in an **absent list**. In this section, the security of the data item STATE is changed in the ORDERS database. Before making changes, the REVIEW ITEMS command output looks like this:

Itm No.	Item Name	Attrib	Security	
16	STATE	X2	(12,13,14/11)	

>

In the example below, all security specifications are deleted from the data item STATE. The DBCPLUS program returns a message confirming deletion of the security specifications.

```
>delete itemsecurity state  
Change of item security accepted.
```

>

Refer to “DELETE ITEMSECURITY” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data item security is deleted, use the REVIEW ITEMS command. In the following example, you can see that all security specifications are deleted from the data item STATE, resulting in an absent list.

```
>review items STATE
```

```
REVIEW ITEMS:
```

Itm No.	Item Name	Attrib	Security
16	STATE	X2	<---- *

>

*\*All security specifications have been deleted from the data item STATE.*

**Note**



Deleting all security specifications from a data item means that all users can read the data item. This is the default security assigned by DBSCHEMA when processing a data item without security specifications.

## Changing Data Set Security

To change data set security, use the CHANGE SETSECURITY command. In this section, the security of the data set INVENTORY is changed in the ORDERS database. Before making changes, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security
6	INVENTORY	D	450	15	481		(12,14/13,18)

>

In the example below, the security of the data set INVENTORY is changed to give user class 14 write access in the user class list. The DBCPLUS program returns a message confirming the change of security.

```
>change setsecurity inventory 14 w
Change of set security accepted.
```

>

Refer to “CHANGE SETSECURITY” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data set security is changed, use the REVIEW SETS command. In the following example, you can see that the security of the data set INVENTORY is changed:

```
>review sets inventory
```

```
REVIEW SETS:
```

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security	
6	INVENTORY	D	450	15	481		(12/13,14,18)	<----- *

>

*\* The security of the data set INVENTORY has been changed to give user class 14 write access to the data set.*

---

## Deleting Data Set Security

To delete all security specifications from a data set, use the DELETE SETSECURITY command. This results in an **absent list**. In this section, the security of the data set CUSTOMER is changed in the ORDERS database. Before making changes to the data set security, the REVIEW SETS command output looks like this:

Set No.	Data Set Name	Type	Capacity	Blk Fac	Blk Max	Dev Class	Security	
1	CUSTOMER	M	221	10	640		(14/11,18)	

>

In the example below, all security specifications are deleted from the data set CUSTOMER. The DBCPLUS program returns a message confirming deletion of the security specifications.

```
>delete setsecurity customer
Change of set security accepted.
```

>

Refer to “DELETE SETSECURITY” in chapter 10, “DBChange Plus Commands,” for a detailed description of the command syntax and parameters.

If you want to verify that the data set security is deleted, use the REVIEW SETS command. In the following example, you can see that all security specifications are deleted from the data set CUSTOMER resulting in an absent list:

>review sets CUSTOMER

REVIEW SETS:

Set					Blk	Blk	Dev	
No.	Data Set Name	Type	Capacity	Fac	Max	Class	Security	
-----								
1	CUSTOMER	M	221	10	640			<----- *

>

*\*All security specifications have been deleted from the data set CUSTOMER.*

**Note**



---

Deleting all security specifications from a data set means that all users can read the data set. This is the default security assigned by DBSCHEMA when processing a data set without security specifications.

---

## Maintaining the Database

---

This chapter explains how to use DBChange Plus to do the following maintenance and performance tuning tasks on a TurboIMAGE/XL database:

- Checking for structural integrity.
- Fixing structural problems.
- Improving database performance.
- Managing database capacity.

In this chapter, the section “Guidelines for Maintaining the Database” discusses when to use DBChange Plus to optimize the performance and integrity of the database. The remainder of this chapter combines the CHECK and FIX commands in several sections. For example, checking and fixing data sets are in the same section, and so on. However, your database maintenance tasks may be implemented differently than the method and order presented in this chapter, depending on your maintenance schedule and the size and activity of your database. For example, you may find it more efficient to create a job to be run on a regular basis to repack or check your database.

Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn how to do more DBChange Plus tasks, refer to chapters 2 through 7 and 9. If you understand DBChange Plus tasks and only want to see command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

## Summary of Commands to Maintain the Database

DBChange Plus provides specific commands that generate statistics about the database structure to help you determine if there are any structural problems, such as broken chains, incorrect pointer linkages, and inconsistent pointers. Several of the commands correct known problems. The commands used for database maintenance are listed alphabetically in Table 8-1 along with a summary of their functions.

### Note



- Before invoking any of the following commands, you must log on to the system as the database creator, run the DBCPLUS program, and specify the database to be used. Refer to chapter 1, “Introduction to DBChange Plus.”
- The commands are not actually applied until the DBAPLUS program has been used. Refer to chapter 9, “Applying Changes to the Database Structure.”

---

**Table 8-1. Summary of Commands to Maintain the Database**

Command	Keyword	Description
CHECK	BASE	Checks the root file, all data sets, and all paths in the database for structural integrity and reports any problems.
	PATH	Checks paths for broken chains and incorrect pointer linkages and reports any problems.
	SET	Checks data sets for internal or pointer inconsistency and reports any problems.
FIX	BASE	Examines all sets and paths in the database and attempts to correct known problems. Does not fix root problems.
	PATH	Examines paths for broken chains and incorrect pointer linkages and attempts to correct known problems.
	SET	Examines data sets for internal pointer or structural inconsistency and attempts to correct known problems.
REPACK	SET	Rebuilds a detail data set by eliminating deleted entry gaps and by adjusting pointers.

---

## Guidelines for Maintaining the Database

To optimize the performance and integrity of the database, you can implement a regular maintenance schedule. For example:

1. Establish baseline statistics for your database against which future statistics can be compared. To do this, use the `CHECK BASE` command with the `QUICK` parameter and archive the resulting output. Note that your database should be performing optimally and should not contain structural problems when you establish your baseline.
2. Periodically check the database and compare the output to the original baseline statistics. To do this, use the `CHECK BASE` command with the `QUICK` parameter.
3. Correct problems that are detected. To do this, use the `FIX` command to correct structural problems in both master and detail data sets. Use the `REPACK` command with the new blocking factor, if necessary, to correct disk space utilization of detail data sets.

Regularly checking and fixing the database can prevent the following problems:

- Large clusters in master data sets.
- Poor packing on heavily used chains.
- Large gaps caused by deleted records in detail data sets.

Optionally, you can use the `FIX PATH` or `FIX BASE` commands only. The `FIX` commands invoke the checking process as explained in the following section.

### How the `FIX` Commands Operate

The `FIX PATH` and `FIX BASE` commands operate as follows:

1. A `CHECK BASE QUICK` (serial read only) is performed. If no problems are found, the `FIX` command stops processing and a message is returned informing you that no problems are detected. If problems are found, then the next step occurs.
2. To further isolate any problems located by `CHECK BASE QUICK`, a standard `CHECK PATH` (serial read followed by a chained read) is performed for each path that has a consistency problem. Corrections are applied to the problems, and messages are returned informing you where the corrections occurred.

The `REPACK`, `CHECK`, and `FIX` commands, along with the `QUICK` option, are discussed in detail later in this chapter.

---

#### Note



---

If a correct backup of the root file does not exist, you must manually correct root file and file structure errors.

---

---

## Repacking Data Sets

To repack a detail data set, use the REPACK SET command. Repacking a detail data set can improve database performance by eliminating long delete chain counts (deleted entry gaps) and optimizing chains by adjusting pointers. Repacking a detail data set physically regroups the records on disk to improve storage efficiency. To further improve performance, you can also change the capacity, blocking factor, and device class of a detail data set with the REPACK SET command.

You can repack a detail data set in one of two ways:

- Serial repack.
- Chained repack.

A serial repack moves up the records to fill deleted entry gaps. A serial repack is faster than a chained repack. A serial repack is useful when you're not sure by which path to organize your repack, or when you've just reduced capacity below the high-water mark.

A chained repack eliminates deleted entry gaps and then places the records sequentially in the data set according to the chosen path. Although slower than a serial repack, the data can be more quickly accessed. When performing a chained repack, select the path that is most frequently accessed. Make sure you select a valid path and that you specify the correct search item. Do not select a path that is either being deleted or added (DELETE PATH or ADD PATH commands are in the current change file) or has no key item data.

In the following example, the detail data set SALES is repacked serially. The capacity of the data set is changed to 1500, the blocking factor is changed to 16, and the device class is changed to DISK3.

```
>repack set sales serial 1501 (16) disk3  
REPACK SET pending for data set.
```

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

PAGE 1 HEWLETT-PACKARD 30391C.02.00 TurboIMAGE/3000: DBSCHEMA  
FRI, APR 27, 1990, 10:40 AM (C) HEWLETT-PACKARD CO. 1987

BEGIN DATABASE ORDERS;

PASSWORDS:

11 CREDIT ;

12 AGENT ;

:

NAME: SALES ,D(11/14,18), DISK3; <---- device class changed

ENTRY: ACCOUNT ( CUSTOMER (PURCH-DATE)),

STOCK# (!PRODUCT ),

QUANTITY ,

PRICE ,

TAX ,

TOTAL ,

PURCH-DATE ,

DELIV-DATE ;

CAPACITY: 1501(16); <---- capacity and blocking  
factor changed

:

NUMBER OF ERROR MESSAGES: 0

ITEM NAME COUNT: 23 DATA SET COUNT: 5

ROOT LENGTH: 1044 BUFFER LENGTH: 541 TRAILER LENGTH: 256

ROOT FILE ORDERS CREATED.

New database schema is saved as temporary file ORDERSSC

Temporary file created for set SALES. <---- data set being repacked

Data set CUSTOMER : No conversion necessary.

Data set PRODUCT : No conversion necessary.

Data set SUP-MASTER : No conversion necessary.

Data set SALES : 17 entries expected.

Data set SALES : 17 entries repacked successfully.

Data set INVENTORY : No conversion necessary.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

---

## Checking and Fixing Paths

This section explains how to check a data set path for inconsistencies and how to fix a path that contains inconsistencies. To check a path for broken chains and incorrect pointer linkages, use the CHECK PATH command. To correct problems found after using CHECK PATH, use the FIX PATH command. You can use CHECK PATH and FIX PATH for either master or detail data sets.

### Checking Paths

When used on a master data set, CHECK PATH checks the synonym chain pointers in the master set. When used on a detail data set, CHECK PATH checks the chain pointers in both the detail data set and associated master data sets. The output for both master data sets and detail data sets is the same. In the following example, the master data set CUSTOMER is checked:

```
>check path customer
    Check Set issued for 1 set(s).
```

```
>
```

If you want to verify that the CHECK PATH command is pending, use the REVIEW PATHS command. Refer to “REVIEW PATHS” in chapter 10, “DBChange Plus Commands.”

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

```
Check file ORDERSRX exists; it will be overwritten (DBA 125).
Starting data set serial reads.
```

```
Tabulating statistics.
```

```
CHECK PATH
```

Set Name	Type	Max	Avg	Std	Pct	Avg	Packing
Search Item	(PS)	Chain	Chain	Dev	Ptrs	Blocks	Ratio
-----							
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A

```
Starting data set chained reads.
```

## CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

---

CUSTOMER	M	Userlabel entry count does not match bitmap entry count (CHK 560). Record 6 is not linked into the proper synonym chain (CHK 360).
SALES	D	
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)	Sum of chain counts in the related master set does not match bitmap entry count for this set (CHK 580).

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

### Analyzing the CHECK PATH Statistics

CHECK PATH performs a serial read followed by a chained read on the data set specified. If you use the QUICK parameter, only a serial read is performed. The output is in two sections: CHECK PATH and CHECK INFORMATION.

When reviewing the CHECK PATH statistics, first examine the Packing Ratio and Avg Chain statistics to help you determine which problems may exist.

Under the CHECK PATH section of the output, the following information is displayed for the above example:

- The data set name. CUSTOMER is the name of the master data set.
- The search item name. Because no search item exists in a master data set, the text SYNONYM CHAINS appears in the field. For a detail data set, the search item name would appear in this field.
- The data set type. The following are valid data set types:
  - M is a manual master data set.
  - A is an automatic master data set.
  - D is a detail data set.
- Max Chain and Avg Chain are the number of entries in the longest synonym or detail chain and the average number of entries per chain, respectively. While long chains are not necessarily harmful to the database, you may want to consider the following guidelines when evaluating chain lengths:
  - Choosing an efficient capacity will keep synonym chains to a minimum.

- The Avg Chain should not exceed 40 percent of the capacity unless it's necessary for the application program.
- The data item most heavily used by an application should be specified as the search item for a detail data set repack. This makes for a more efficient repack.
- Sorted chains should be less than 50 records unless key values are added in ascending order over time. For example, a date field where the currently added date will always be greater than any of the prior dates can exceed more than 50 records.
- Std Dev (standard deviation) is an indication of the accuracy of the Avg Chain statistic. The closer to 0.00 this number is, the more accurate the Avg Chain statistic is.
- Pct Far Ptrs is the percentage of forward pointers that point outside the current block. This statistic can be used to enhance the packing ratio described below. The Pct Far Ptrs represents the number of pointers that point beyond the current block.
- Avg Blocks is the average number of blocks per chain.
- Packing Ratio is the efficiency of the path. It is the optimal average number of blocks per chain divided by the actual average number of blocks per chain. A value of 100 percent means that every individual chain for the specified path occupies the minimum number of blocks possible. Although packing ratios vary from application to application, try to maintain a packing ratio of at least 60 percent on primary paths.

The CHECK INFORMATION section of the output identifies specific records in the data set that contain problems. If no problems are found, the following message is displayed:

```
No problems were detected by CHECK.
```

## Fixing Paths

When used on a master data set, FIX PATH examines and attempts to correct problems with the synonym chain pointers. When used on a detail data set, FIX PATH examines and attempts to correct the path chain pointers in both the detail data set and associated master data sets. In the following example, FIX PATH is used to correct the problems found in the previous CHECK PATH example:

```
>fix path customer
  Fix Path issued for 1 set(s).
```

```
>
```

If you want to verify that the FIX PATH command is pending, use the REVIEW PATHS command. Refer to “REVIEW PATHS” in chapter 10, “DBChange Plus Commands.”

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

Check file ORDERSRX exists; it will be overwritten (DBA 125).  
Starting data set serial reads.

Tabulating statistics.

CHECK PATH

Set Name	Type	Max	Avg	Std	Pct	Avg	Packing
Search Item	(PS)	Chain	Chain	Dev	Far	Blocks	Ratio
-----							
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A

Starting data set chained reads.

CHECK INFORMATION

Set Name	Type
Search Item	Message(s)
-----	
CUSTOMER	M
	Userlabel entry count does not match bitmap entry count (CHK 560).
SALES	D
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)
	Sum of chain counts in the related master set does not match bitmap entry count for this set (CHK 580).

FIX INFORMATION

-----  
All problems detected have been fixed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The FIX PATH output is similar to the CHECK PATH output. The same CHECK PATH and CHECK INFORMATION reporting occurs. In addition, FIX INFORMATION reports the status of the FIX PATH process. In the above example, FIX PATH corrected all problems found.

---

## Checking and Fixing Data Sets

This section explains how to check data sets for internal pointer or structural inconsistency and how to fix data sets that contain inconsistencies. To check a data set for internal or pointer inconsistency, use the CHECK SET command. To correct problems found after using CHECK SET, use the FIX SET command. You can use CHECK SET and FIX SET for either master data sets or detail data sets.

### Note



CHECK SET checks only the internal pointer consistency of the set(s) specified. It does not check the pointer consistency across sets. For example, the pointer from a chain head in a master data set to the first record on the chain in the associated detail set is not checked. To check across data sets, use the CHECK PATH command.

## Checking a Master Data Set

To check a master data set, use the CHECK SET command. In the following example, the master data set CUSTOMER is checked.

```
>check set customer
  Check Set issued for 1 set(s).

>
```

If you want to verify that the CHECK SET command is pending, use the REVIEW SETS command. Refer to “REVIEW SETS” in chapter 10, “DBChange Plus Commands.”

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

```
Starting data set serial reads.
```

```
Tabulating statistics.
```

```
CHECK SET (MASTER)
```

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0

```
CHECK INFORMATION
```

```
Set Name          Type
  Search Item
  Message(s)
```

```
-----
CUSTOMER          M
  Userlabel entry count does not match bitmap entry count (CHK 560).
SALES             D
  ACCOUNT         (Path 1; linked to master CUSTOMER, path 1)
```

Sum of chain counts in the related master set does not  
match bitmap entry count for this set (CHK 580).

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

### **Analyzing the CHECK SET Statistics for a Master Data Set**

CHECK SET performs a serial read followed by a chained read on the master data set specified. If you use the QUICK parameter, only a serial read is performed. The output is in two sections: CHECK SET and CHECK INFORMATION.

Under the CHECK SET section of the output, the following information is reported for the above example:

- The data set name. CUSTOMER is the name of the master data set.
- The data set type. The following are valid data set types:
  - M is a manual master data set.
  - A is an automatic master data set.
- Entries is the number of entries in the data set.
- Capacity is the capacity of the data set.
- Pct Full is the percentage of the data set capacity that is not available, rounded to the nearest whole percentage. This statistic is useful for capacity planning. The recommended percent full for a master data set is between 60 percent and 80 percent. If a master data set capacity is significantly less than 60 percent full, a serial read becomes slower. If a master data set is significantly more than 80 percent full, DBPUT intrinsics involving the master data set can slow down. If you are experiencing performance problems, consider changing the capacity of the data set. (Refer to “CHANGE SET” in chapter 10, “DBChange Plus Commands.”)
- Pct Sec is the percentage of secondary entries. In general, the lower the percentage, the better. A high percentage indicates that the hashing algorithm is creating many synonyms. To decrease the percentage of secondaries, increase the data set capacity to a larger number that is not a power of two. (Refer to “CHANGE SET” in chapter 10, “DBChange Plus Commands.”)
- Longest Cluster and Average Cluster are the longest and average number of TurboIMAGE blocks that must be read to find an open slot to place a synonym. If two records with the same hash value are added to a master data set, one of them must be placed in another available slot. Acceptable guidelines range from a cluster of 10 on a very heavily loaded transaction processing system to as much as 200 on a very lightly loaded system. A collection of historical statistics about the database can help you evaluate if the longest and average clusters are within a reasonable range. In addition, your database users can help identify clustering

problems by reporting slow response time when adding records to a particular data set. Once you have determined if the range should be changed, increase or decrease the data set capacity accordingly. (Refer to “CHANGE SET” in chapter 10, “DBChange Plus Commands.”)

The CHECK INFORMATION section of the output identifies specific records in the data set that contain problems. If no problems are found, the following message is displayed:

```
No problems were detected by CHECK.
```

## Fixing a Master Data Set

To correct problems identified by CHECK SET, use the FIX SET command.

When fixing a master data set, the following messages may occur:

- Dataset *DataSetName* record *n* is duplicate manual master (FIX 170). Key value: *n*

To correct this problem, examine the key values in the data set *DataSetName* and delete the key that has no entries. Be sure to print the data before deleting.

- Please verify key for dataset *DataSetName* (FIX 170). Key value: *n*

To correct this problem, examine the key value and the record to determine if the data is damaged.

In the following example, FIX SET is used to correct the problems found in the previous CHECK SET example:

```
>fix set customer
  Fix Set issued for 1 set(s).
```

```
>
```

If you want to verify that the FIX SET command is pending, use the REVIEW SETS command. Refer to “REVIEW SETS” in chapter 10, “DBChange Plus Commands.”

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

```
Check file ORDERSRX exists; it will be overwritten (DBA 125).
Starting data set serial reads.
```

```
Tabulating statistics.
```

```
CHECK SET (MASTER)
```

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0

CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

-----

CUSTOMER	M	Userlabel entry count does not match bitmap entry count (CHK 560).
SALES	D	
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)	
		Sum of chain counts in the related master set does not match bitmap entry count for this set (CHK 580).

FIX INFORMATION

-----

All problems detected have been fixed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The FIX SET output is similar to the CHECK SET output. The same CHECK PATH and CHECK INFORMATION reporting occurs. In addition, FIX INFORMATION reports the status of the FIX SET process. All problems found in the previous example are corrected.

**Checking a Detail Data Set**

To check a detail data set, use the CHECK SET command. In the following example, the detail data set SALES is checked:

```
>check set sales
Check Set issued for 1 set(s).
```

>

If you want to verify that the CHECK SET command is pending, use the REVIEW SETS command. Refer to "REVIEW SETS" in chapter 10, "DBChange Plus Commands."

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

```
Check file ORDERSRX exists; it will be overwritten (DBA 125).
Starting data set serial reads.
```

```
Tabulating statistics.
```

```
CHECK SET (DETAIL)
```

```
Pct      Highest      Delete
```

Detail Set Name	Entries	Capacity	Full	Entry Used	Chain Count
SALES	17	504	3	17	0

CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	
SALES	D
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)
Path Chain inconsistencies detected (CHK 400).	

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

### Analyzing the CHECK SET Statistics for a Detail Data Set

CHECK SET performs a serial read followed by a chained read on the detail data set specified. If you use the QUICK parameter, only a serial read is performed. The output is in two sections: CHECK SET and CHECK INFORMATION.

Under the CHECK SET section of the output, the following information is displayed for the previous example:

- The data set name. SALES is the name of the detail data set.
- Entries is the number of entries in the data set.
- Capacity is the capacity of the data set.
- Pct Full is the percentage of the data set capacity that is not available, rounded to the nearest whole percentage. This statistic is useful for capacity planning.
- Highest Entry Used is the record number of the highest entry ever used (the high-water mark) in the data set. When reducing data set capacity, do not reduce it below the highest entry used; otherwise, you must repack the data set to recover the unused space in the middle of the data set. (Refer to “REPACK” in chapter 10, “DBChange Plus Commands.”)
- Delete Chain Count is the number of records in the delete chain. This number should be as close to 0 as possible. A high delete chain count may mean a problem with a large quantity of deletes. For example, if you have a program that performs a large number of deletes, gaps may be left in your detail data set. Subsequent record additions may disburse data randomly within the data set. As a result, serial reads may be slow. To correct a high Delete Chain Count, repack the data set. (Refer to “REPACK” in chapter 10, “DBChange Plus Commands.”)

The CHECK INFORMATION section of the output identifies specific records in the data set that contain problems. If no problems are found, the following message is displayed:

No problems were detected by CHECK.

### Fixing a Detail Data Set

Use the FIX SET command to correct problems identified by CHECK SET. In the following example, FIX SET is used to correct the problems found in the previous example:

```
>fix set customer
```

If you want to verify that the FIX SET command is pending, use the REVIEW SETS command. Refer to “REVIEW SETS” in chapter 10, “DBChange Plus Commands.”

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

```
Check file ORDERSRX exists; it will be overwritten (DBA 125).
Starting data set serial reads.
.
Tabulating statistics.
```

CHECK SET (MASTER)

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0

CHECK INFORMATION

Set Name	Type	
Search Item		
Message(s)		
-----		
CUSTOMER	M	Userlabel entry count does not match synonym chain count (CHK 570).
SALES	D	
ACCOUNT		(Path 1; linked to master CUSTOMER, path 1)
		Sum of chain counts in the related master set does not match bitmap entry count for this set (CHK 580).

FIX INFORMATION

-----

All problems detected have been fixed.

PERFORM COMMANDS completed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The FIX SET output is similar to the CHECK SET output. The same CHECK PATH and CHECK INFORMATION reporting occurs. In addition, the FIX INFORMATION section reports how the problem was corrected and the status of the FIX SET process. All problems found in the previous example are corrected.

---

## Checking and Fixing the Entire Database

You can combine the CHECK PATH and FIX PATH and CHECK SET and FIX SET commands to check and fix the entire database by using the CHECK BASE and FIX BASE commands. CHECK BASE and FIX BASE check *all* the paths and sets in the database; while CHECK PATH and FIX PATH are used only for paths in a specific data set, and CHECK SET and FIX SET only for a specific data set.

Refer to "CHECK BASE" and "FIX BASE" in chapter 10, "DBChange Plus Commands," for a detailed description of the command syntax and output statistics.

---

## Using the QUICK Option with CHECK Commands

When used with the CHECK PATH, CHECK SET, and CHECK BASE commands, the QUICK option displays a summarized version of the standard CHECK output. The QUICK option performs a serial read only. This method is quicker than standard checking, which uses both serial and chained reads. The QUICK option can identify that a problem exists, but it cannot isolate the specific records that need to be fixed. If you already know that a problem exists, but you need more detailed information, such as which records are in error, use the standard checking method.

### Note



---

If a record contains only a damaged key, but does not contain any other damage, the CHECK BASE QUICK command will not detect the damaged key. To locate damaged keys on a large database, create a copy of your database and perform a standard CHECK BASE on the database copy. If you have a small database, perform a standard CHECK BASE followed by a FIX BASE.

---

The QUICK option is useful if you have a large database and want to determine if a problem exists but you don't need to know exactly where. In the following example, the QUICK option is used to check the paths in the detail data set SALES. ACCOUNT is the search item linking SALES to the master data set CUSTOMER.

```
>check path sales (account) quick  
Check Path pending for 1 set(s).
```

```
>
```

After issuing a PERFORM COMMANDS to invoke DBAPLUS, the following output is displayed:

Check file ORDERSRX exists; it will be overwritten (DBA 125).  
Starting data set serial reads.

Tabulating statistics.

CHECK PATH

Set Name	Type	Max	Avg	Std	Pct	Avg	Packing
Search Item	(PS)	Chain	Chain	Dev	Far	Blocks	Ratio
-----							
SALES	D						
ACCOUNT	( S)	17231	2874.67	7033.14	6	1.17	6.86

Starting data set chained reads.

CHECK INFORMATION

Set Name	Type
Search Item	Message(s)
-----	
SALES	D
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1) Path Chain inconsistencies detected (CHK 400).

PERFORM COMMANDS completed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

Refer to "CHECK PATH" and "FIX PATH" in chapter 10, "DBChange Plus Commands," for a detailed description of the command syntax and output statistics.

## Applying Changes to the Database Structure

---

This chapter explains how to use the DBAPLUS program to apply changes to the database structure.

Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn how to do more DBChange Plus tasks, refer to chapters 2 through 8. If you understand DBChange Plus tasks and only want to see command syntax information, refer to chapter 10, “DBChange Plus Commands.”

---

### What is DBAPLUS?

DBChange Plus consists of two programs—DBCPLUS and DBAPLUS. In earlier chapters in this guide, changes made to the ORDERS database were actually stored in the change file using the DBCPLUS program and not applied to the database. The DBAPLUS program actually applies the changes to the database. It also initiates pending CHECK and FIX commands stored in the change file.

The DBAPLUS program requires exclusive access to the database. Note that the DBAPLUS program disables Intrinsic Level Recovery (ILR), logging, and recovery for the database being restructured. If ILR needs to be performed, it will be done when ILR is disabled.

---

### Before Restructuring the Database

Before you use the DBAPLUS program to restructure the database, be sure to follow the steps below:

1. Obtain a current backup copy of the database as a precautionary measure. If the DBAPLUS program is unable to complete the restructure because of a fatal error, your original database may be left in an inconsistent state. In this case, you will need to restore a backup copy of the database.

To be sure you have a current copy, the DBAPLUS program checks if the database has been modified since the last DBStore. If it has, the DBAPLUS program asks if you want to continue with the restructure. You should continue only if you have a backup copy of the database.

2. Make sure that all security criteria have been met. To use the DBAPLUS program, you must be logged on as the database creator. The DBAPLUS program checks that the logged-on user matches the logon stored in the change file. If it does not, a security violation message is displayed and the DBAPLUS program terminates.
3. Ensure that adequate disk space is available for the DBAPLUS program to perform the restructure. To estimate how much disk space you need, first identify the data sets being altered. Assume that the DBAPLUS program will use at least as much space as the data sets currently occupy. Adjust your estimate depending on the nature of the changes stored in the change file. Modifications, such as adding new fields or changing data set capacity, can affect the disk space needed by the new data sets. The method the DBAPLUS program uses to restructure the database also affects the disk space needed. Refer to “How DBAPLUS Restructures a Database” later in this chapter for more information.

If not enough disk space exists to create all the necessary temporary files before beginning the restructure, the DBAPLUS program prompts you to make a choice as listed below. Note, however, that neither of these methods guarantee that you can successfully run the DBAPLUS program if you have very large data sets and very little available disk space.

- a. You can choose to exit the DBAPLUS program without performing the restructure. Before using the DBAPLUS program again, obtain more disk space or remove some of your changes from the change file with DBChange and restructure the database in stages.
- b. You can direct the DBAPLUS program to perform the restructure using an alternate method which uses less disk space. However, be aware that if a fatal error occurs during the restructure, the database is left in an inconsistent state. This method creates one temporary data set at a time. When the changes are successfully made to the temporary data set, the DBAPLUS program purges the original data set. The danger is that if a fatal error occurs during this process, some of the original data sets will have been purged before the new ones have been saved. In this case, you need to run DBUTIL to purge the database and restore an archival version.

---

## Saving the New Schema

The DBAPLUS program creates a new schema based on the changes stored in the change file. This schema is used when the DBAPLUS program calls DBSCHEMA to build a new root file. The new schema is placed in a temporary file named *dbnameSC*. (Any temporary file with the same name will be purged.) You must use the MPE XL SAVE command on this new schema file if you want to keep it as a permanent MPE XL file.

To maintain the security of the database passwords when the DBAPLUS program is run from a batch job, the listing of the new schema is suppressed with a file equation which sets DBSLIST to \$NULL. You must alter the job stream file when running the DBAPLUS program in batch mode if you wish to save the schema listing. The job stream example later in this chapter shows how to do this.

### Saving the Schema Processor Output

The schema processor output (DBSLIST) is sent to \$STDLIST. You can redirect this listing to a permanent MPE XL file by issuing the following file equation *before* running the DBAPLUS program:

```
:file dbslis= FileName;dev=disc;rec=-80,,f,ascii;disc= n;save;nocctl
```

where *n* is the number of records in the file. If *n* is not specified, the default value is 1023, which may not be sufficient for a large database.

---

## Methods of Using the DBAPLUS Program

The DBAPLUS program can be invoked independently in either interactive or batch mode. Because the DBAPLUS program requires exclusive access to the database, the method you choose depends on the number and/or type of changes to be made. Because batch mode can be set to run at times when the database is not being accessed, this method is desirable when user access to the database is important and/or when lengthy changes are stored in the change file. Use the DBAPLUS program interactively only if the changes stored in the change file do not take long to perform, or if user access to the database is not critical.

To perform a restructure, the following three methods are available:

- Initiating the DBAPLUS program interactively from within the DBCPLUS program by using a PERFORM COMMANDS.
- Creating a job file from within the DBCPLUS program by using a PERFORM COMMANDS to execute the DBAPLUS program. Then exit the DBCPLUS program and stream the job.
- Exiting the DBCPLUS program and running the DBAPLUS program independently.

Each of these methods is explained on the following pages.

## **Initiating the DBAPLUS Program Interactively**

The DBAPLUS program can be invoked interactively to perform a restructure from within the DBCPLUS program. To invoke the DBAPLUS program this way, do the following:

1. Run the DBCPLUS program (DBCPLUS).
2. Issue the BASE command to specify the database to be restructured.
3. Issue one or more DBCPLUS restructure commands. Issue a **PERFORM COMMANDS** to immediately invoke the DBAPLUS program. When you are asked if changes should be performed online or if a job stream file should be created, enter Y for yes. The example below shows how to do this:

>perform commands

PERFORM COMMANDS now or create job file [job/y/N]? y

Please wait ...

HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985  
THU, JAN 18, 1990 7:15 AM

Database has been modified since last DBSTORE (DBA 220).

Continue [y/N]? y

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

### **Creating a DBAPLUS Job File from within the DBChange Program**

The DBAPLUS program can be invoked by creating a job file from within the DBCPLUS program. To invoke the DBAPLUS program this way, do the following:

1. Run the DBCPLUS program (DBCPLUS).
2. Issue the BASE command to specify the database to be restructured.
3. Issue one or more DBChange restructure commands.
4. Issue PERFORM COMMANDS from within the DBCPLUS program and create a job stream. When you enter PERFORM COMMANDS, include the JOB and PARM parameters. After the job stream file is created, exit the DBCPLUS program. When you are ready to stream the job, enter the job ( *DatabaseNameJJ*) at the MPE XL system prompt (:). For example:

>perform commands job 1

Please wait.....change file is being verified.

New job stream file ORDERSJJ created.

>exit

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

\*\*\*\*\* DBCHANGE STATISTICS\*\*\*\*\*

Number of databases prepared for rename:	0
Number of databases prepared for restructure:	1
Number of databases prepared for erase :	0
Number of databases prepared for repack:	0
Number of databases prepared for check :	0
Number of databases prepared for fix :	0

:stream ordersjj

For more information about PARM values, refer to “Using PARM Values” later in this chapter.

## Modifying the Job File

When instructed to do so, DBChange creates a job stream file for each existing change file. The name of this job stream file is the database name with JJ appended ( *DatabaseNameJJ*). Use this job stream file to perform a restructure in batch mode. This job stream file does the following:

- Sets a file equation for the change file.
- Sets DBSLIST to \$NULL to suppress the listing of the new schema.
- Contains commands to purge the old version of the schema and to save the temporary file containing the new schema. Note that these commands have been “commented out.” To use them, alter the job stream file with a text editor.
- Contains TELL commands that report the beginning and the completion of the restructure.

You may need to make the following alterations to a job stream file:

- Add any necessary user, group, and/or account passwords.
- Alter the run-time parameter of the RUN DBALTER command. (This parameter provides answers to questions DBAPLUS may ask. Refer to Table 9-1 later in this chapter.)
- Remove the :comment\*\* portion of the commands that save the temporary file containing the new schema.

The example below shows the unedited job stream file before editing created by DBChange for the database ORDERS:

```
:job JDBALTER,KELLY.FINANCE,DATABASE
:comment*****
:comment*      This job will perform the actual
:comment*      restructuring of the database by
:comment*      calling DBALTER.
:comment*****
:
:tell  KELLY.FINANCE;  /-->Start JDBALTER for ORDERS
:
:file DBCHGF = ORDERSCF
:file DBSLIST = $NULL
:
:comment*****
:comment*      DBALTER creates the new schema file as a
:comment*      job temporary file. To save it, purge the
:comment*      old schema file before running DBALTER and
:comment*      save the new schema file after DBALTER
:comment*      completes.
:comment*****
:comment*      :purge ORDERSSC
:comment*****
```

```

:
:comment*      verify that DBALTER parmvalue is correct
:
:run dbaplus; parm = 1
:
:comment*****
:comment*      :save ORDERSSC
:comment*****
:
:tell  KELLY.FINANCE;  /-->End JDBALTER for ORDERS
:
:eoj

```

In the example below, the ORDERS job stream file, ORDERSJJ, has been edited to add passwords, to change the parameter of the RUN DBALTER command, and to remove the “commented out” portions that save the new schema file.

```

:job JDBALTER,KELLY/USER.FINANCE/MGR,DATABASE/ALL
:comment*****
:comment*      This job will perform the actual
:comment*      restructuring of the database by
:comment*      calling DBALTER.
:comment*****
:
:tell  KELLY.FINANCE;  /-->Start JDBALTER for ORDERS
:
:file DBCHGF = ORDERSCF
:file DBSLIST = $NULL
:
:comment*****
:comment*      DBALTER creates the new schema file as a
:comment*      job temporary file. To save it, purge the
:comment*      old schema file before running DBALTER and
:comment*      save the new schema file after DBALTER
:comment*      completes.
:comment*****
:
:purge ORDERSSC
:
:comment*      verify that DBALTER parmvalue is correct
:
:run dbaplus; parm = 15
:
:save ORDERSSC
:
:tell  KELLY.FINANCE;  /-->End JDBALTER for ORDERS
:
:eoj

```

## Running the DBAPLUS Program Independently

You can invoke the DBAPLUS program independent of the DBCPLUS program in two ways, as follows:

- Issue a file equation for DBCHGF, the formal file designator for the change file. Next, run the DBAPLUS program providing a parm value. For example:

```
:hello kelly.finance,database  
:file dbchgf=orderscf  
:run dbaplus.pub.sys;parm=1
```

The DBAPLUS program determines the database name from the name of the change file in the file equation. This file equation is reset by the DBAPLUS program upon completion of the restructure. Note that a run-time parameter, PARM=1, is included. Use this parameter if you want to keep the change file. For more information about PARM values, refer to “Using PARM Values” later in this chapter.

- Run the DBAPLUS program, but allow it to prompt you for the database name. For example:

```
:hello kelly.finance,database  
:run dbaplus.pub.sys
```

```
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985  
THU, JAN 18, 1990, 9:11 AM
```

Which database? orders

The DBAPLUS program determines the correct change file by reading the database name. Because no parameter is specified, the change file is automatically purged upon completion of the restructure.

---

## Using PARM Values

PARM values are used in the following situations:

- When creating a job file from within the DBCPLUS program.
- When running the DBAPLUS program independently.

To restructure the database, the DBAPLUS program may need you to answer four questions. In interactive mode, you can answer these questions if and when they are asked. However, batch mode requires a run-time parameter (PARM value) to provide answers to these questions in case they are asked. Optionally, a parameter can also be used to keep the change file.

PARM values can be determined by adding them together. For example, if you want both PARM=1 and PARM=4 to take effect, add them together, then specify PARM=5.

Table 9-1 describes the parameter values and which DBAPLUS questions they answer.

**Table 9-1. DBAPLUS Parameters**

PARM Value	You do not have enough disk space to build all necessary temporary data sets before beginning the restructure. Do you want DBAPLUS to continue with the restructure using an alternative restructuring method?	The database has been modified since the last DBSTORE. Do you wish to continue?	Your root file is not in normal DBSCHEMA format. Do you want DBAPLUS to fix it and continue with the restructure?	The change file is normally purged after successful completion of DBAPLUS. Do you want to save the change file?
1				Y
2			Y	
3			Y	Y
4		Y		
5		Y		Y
6		Y	Y	
7		Y	Y	Y
8	Y			
9	Y			Y
10	Y		Y	
11	Y		Y	Y
12	Y	Y		
13	Y	Y		Y
14	Y	Y	Y	
15	Y	Y	Y	Y

---

## How DBAPLUS Restructures a Database

Before actually beginning the restructure, DBAPLUS creates an intermediate version of the database. The new root file and any data sets which have structural changes are created as temporary files. Once DBAPLUS has created all these temporary files, it begins the restructure. If the restructure is successful, the original data sets are purged and the new ones saved.

## Caution



- 
- If one of the stored changes in the change file is the reordering of data sets, you should not have any temporary files named ALTER $nn$  where  $nn$  is the number of one of the data sets in the database. The DBAPLUS program creates temporary files with these names and overwrites your temporary files.
  - Do not create a database with the name TEMPDB. The DBAPLUS program creates a file name TEMPDB to restructure master data set. Your TEMPDB file will be purged and can also prevent the DBAPLUS from processing.
- 

## Possible Restructuring Problems

If a fatal error occurs during the restructuring process, the following situations may occur leaving the original database in an inconsistent state. If this happens, resolve the error and restore the database from your backup copy.

- If a detail data set path is changed, the associated master set in the original database is altered during the restructuring process. If the detail set whose path you changed has more than one path, all the master data sets associated with these additional paths are also altered. Their chain counts are reset to zero to allow the chains to be rebuilt while processing the path change. If a fatal error occurs during this change or after it is processed, those associated master data sets which are permanent files in the original database are left in an inconsistent state.
- If data sets are reordered from top to bottom, the original database is altered during the restructuring process. For example, if the original SET02 becomes SET03 after the resequence, the DBAPLUS program places SET02 in a temporary file named SET03 to prevent a naming conflict with the original SET03. Therefore, SET02 in the original database is altered or no longer exists. This top to bottom reordering occurs when a set is moved above its original position in the schema, or when a set is added or deleted from the database. If a fatal error occurs during this change or after it is processed, the original database is left in an inconsistent state.
- If the entries in a detail data set are erased, the automatic master data set(s) linked to the detail data set may be purged and re-created with zero entries. The chain count and chain pointers of the path to the detail data set may also be updated to zero. Additionally, the manual master data set(s) linked to the detail data set may have chain count and chain pointers updated to zero. The original data set is purged and re-created with zero entries.
- If the entries in a manual master data set are erased, the original data set is purged, then re-created with zero entries. If a fatal error occurs after the original data set is purged, no manual master data set exists.

- When a detail data set is repacked, the chain pointers in the master data set linked to the detail data set are adjusted. The pointers in the detail data set are also adjusted.

---

## Restructuring the ORDERS Database

Following is an example of the messages the DBAPLUS program sends to your terminal when it is performing the restructure interactively. In this example, the DBAPLUS program is restructuring the ORDERS database:

```
>perform commands
Please wait.....change file is being verified.

PERFORM COMMANDS now or create job file [job/y/N]? y

Please wait ...
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
MON, JAN 8, 1990, 11:18 AM

Database has been modified since last DBSTORE (DBA 220).
Continue [y/N] ? y

PAGE 1          HEWLETT-PACKARD 30391C.00.52 TurboIMAGE/3000: DBSCHEMA
          MON, JAN 8, 1990, 11:18 AM (C) HEWLETT-PACKARD CO. 1987

BEGIN DATABASE ORDERS;

PASSWORDS:
  11 CREDIT ;
:

NUMBER OF ERROR MESSAGES: 0
ITEM NAME COUNT: 26          DATA SET COUNT: 6
ROOT LENGTH: 1176          BUFFER LENGTH: 561          TRAILER LENGTH: 256

ROOT FILE ORDERS CREATED.

New database schema is saved as temporary file ORDERSSC
Data set CUSTOMER          : No conversion necessary.
Data set DATE-MASTER       : No conversion necessary.
Data set PRODUCT           : No conversion necessary.
Data set SUP-MASTER        : No conversion necessary.
Data set SALES              : No conversion necessary.
Data set INVENTORY         : No conversion necessary.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985
```

>

In the previous example, the DBAPLUS program was able to find enough disk space to create all the temporary files before beginning the restructure. However, if the DBAPLUS program is unable to create all the temporary files before beginning the restructure, the message delivered to the terminal reveals at what point in the process the DBAPLUS program ran out of disk space. The message the DBAPLUS program displays will be exactly the same as the previous message until the DBAPLUS program begins to build temporary files.

The following example illustrates what happens when the DBAPLUS program attempts to create a temporary file for the data set BRANCH-STORES. The example begins just after the new schema has been displayed on the screen.

:

DATA SET NAME	TYPE	FLD CNT	PT CT	ENTR LGTH	MED REC	CAPACITY	BLK FAC	BLK LGTH	DISC SPACE
CUSTOMER	M	9	1	43	54	221	10	541	128
DATE-MASTER	A	1	3	3	26	211	19	496	64
PRODUCT	M	3	2	16	33	307	16	529	112
SALES	D	9	4	24	40	504	14	561	192
SUP-MASTER	M	5	1	31	42	211	12	505	80
INVENTORY	D	6	3	20	32	450	15	481	128
BRANCH-STORES	D	5	2	24	32	225	15	481	64
TOTAL DISC SECTORS INCLUDING ROOT: 768									

NUMBER OF ERROR MESSAGES: 0

ITEM NAME COUNT: 26

DATA SET COUNT: 6

ROOT LENGTH: 1176

BUFFER LENGTH: 561

TRAILER LENGTH: 256

ROOT FILE ORDERS CREATED.

New database schema is saved as temporary file ORDERSSC

Data set CUSTOMER : No conversion necessary.

Data set DATE-MASTER : No conversion necessary.

Data set PRODUCT : No conversion necessary.

Data set SUP-MASTER : No conversion necessary.

Data set SALES : No conversion necessary.

Data set INVENTORY : No conversion necessary.

DBAPLUS has tried to create all the necessary temporary data sets before beginning the restructuring process and has run out of disk space. You can:

1) exit DBAPLUS and obtain more disk space or remove some of the changes in the change file with DBChange and run DBAPLUS again.

2) ask DBAPLUS to try again using a second restructuring method. This method creates one temporary data set at a time. The original data set is purged when the changes for that set are successfully processed. Although this requires less disk space, there is no guarantee that there is enough disk space for the entire restructure.

If you receive the preceding message, choose one of the two options given to complete the restructure. If you choose method 2, be sure you have a backup copy of the database.

## DBChange Plus Commands

---

This chapter provides detailed command syntax descriptions of the DBChange Plus commands. The commands appear alphabetically.

Before reading this chapter, you should be familiar with the information in chapter 1, “Introduction to DBChange Plus.” To learn about more DBChange Plus tasks, refer to chapters 2 through 9. For a complete list of tasks and where to find them in this guide, refer to the preface.

---

### DBChange Plus Command Formats

- DBChange commands are in the format of one or two keywords plus parameters. The keywords specify the function. The number of parameters varies according to the function. Some parameters are required, while others are optional. If you omit an optional parameter, the DBCPLUS program uses the default value.
- Commands and keywords can be abbreviated. In the command syntax, the portion of the command or keyword enclosed in brackets ([ ]) can be omitted. For example, A[DD] FIELD can be abbreviated to A FIELD. Command and keyword abbreviations are shown in Table 10-1.
- Parameters are positional; they must be entered in the exact order shown in the command syntax.
- A minimum of one blank space is required between keywords. A blank space or a comma must be used between parameters. A blank space or comma can be used interchangeably.
- Some commands use a slash (/), a left parenthesis, or a right parenthesis as a delimiter between two parameters.
- Some commands use an exclamation point (!) to designate a primary path.
- Some commands use an @ symbol to designate “all.”
- With the exception of passwords, DBChange Plus is not case sensitive. DBChange Plus automatically upshifts all alphabetic input. Passwords, however, must be entered by the user exactly as they appear in the schema.
- The maximum length of a command is 3 lines, each line containing 72 characters, for a total command length of 216 characters. DBChange Plus does not read any characters exceeding the 72

per line maximum or the 216 total maximum. If a command does not fit on one line, use an ampersand (&), preceded by a blank or comma, as the last character on a line to indicate that the command is continued on the next line.

- Only one command per command line is allowed. A command can be up to three lines.

---

## **Summary of DBChange Plus Commands**

Table 10-1 summarizes the DBChange Plus commands and provides command abbreviations.

**Table 10-1. Summary of DBChange Plus Commands**

Command	Keyword	Abbreviation	Description
ADD	FIELD	A F	Adds a field (data item) to a manual master or detail data set.
	ITEM	A ITEM	Adds a data item to the database.
	PASSWORD	A PAS	Adds a password to the database.
	PATH	A PAT	Adds a path from a detail data set to a master data set and defines a search item.
	SET	A SET	Adds a master or detail data set to the database.
	SORT	A SO	Adds a sort item to a search item.
BASE		B	Specifies the database to be modified.
CANCEL	COMMANDS	CA COM	Cancels all changes entered since the last BASE command.
	ERASE	CA E	Cancels the previous ERASE SET command.
	REPACK	CA RE	Cancels the previous REPACK SET command.
CHANGE	ATTRIBUTES	CHA A	Changes the subitem count, type, and length of a data item.
	BLOCKFACTOR	CHA BLOCKF	Changes the blocking factor of a data set.
	BLOCKMAX	CHA BLOCKM	Changes the maximum block length of one or more data sets.
	CAPACITY	CHA CA	Changes the capacity of a data set.
	DEVICECLASS	CHA D	Moves a data set from one device (disk) to another.
	ITEM	CHA ITEM	Changes the subitem count, type, length, security, and/or order of a data item.
	ITEMSECURITY	CHA ITEMSE	Changes user class access to a data item.
	PASSWORD	CHA PAS	Changes a password.
	PRIMARYPATH	CHA PR	Changes the primary path for a detail data set that has at least two paths.
	SETSECURITY	CHA SETSE	Changes user class access to a data set.

**Table 10-1. Summary of DBChange Plus Commands (continued)**

Command	Keyword	Abbreviation	Description
CHECK	BASE	CHE BA	Checks the root file, all data sets, and all paths in the database for structural integrity and reports any problems.
	PATH	CHE PAT	Checks paths for broken chains and incorrect pointer linkages and reports any problems.
	SET	CHE SET	Checks data sets for internal or pointer inconsistency and reports any problems.
CONTROL		CON	Specifies the setting of several options controlling the way DBChange operates. Automatically adjusts data set capacities.
COPY		COP	Copies a database.
DELETE	FIELD	D F	Deletes a field (data item) from a data set entry.
	ITEM	D ITEM	Deletes a data item and its associated fields and paths from the database.
	ITEMSECURITY	D ITEMSE	Deletes all security specifications for a data item.
	PASSWORD	D PAS	Deletes a password and all associated user classes from the database.
	PATH	D PAT	Deletes a path from a detail data set.
	SET	D SET	Deletes a master or detail data set and its associated paths from the database.
	SETSECURITY	D SETS	Deletes all security specifications from a master or detail data set.
ERASE	SET	ER SET	Erases entries in manual master and detail data sets.
		E	Stops the execution of the DBCPLUS program.
FIX	BASE	F BA	Examines all sets and paths in the database and attempts to correct known problems.
	PATH	F PAT	Examines paths for broken chains and incorrect pointer linkages and attempts to correct known problems.
	SET	F SET	Examines data sets for internal pointer or structural inconsistency and attempts to correct known problems.
HELP		H	Provides information about DBChange Plus commands.
OUTPUT		O	Redirects DBChange Plus output to a file or device.

**Table 10-1. Summary of DBChange Plus Commands (continued)**

Command	Keyword	Abbreviation	Description
PERFORM	COMMANDS	PE COM	Invokes DBAPLUS to restructure, check, or maintain the database according to the commands stored in the current change file.
PRINT	SCHEMA	PR SC	Prints the database schema, applying any changes stored in the current change file.
RECOVER	ITEM	REC ITEM	Cancels a data item deletion.
	SET	REC SET	Cancels a data set deletion.
REDO		R	Allows a user to correct or modify the last DBChange Plus command entered.
RENAME	BASE	REN BA	Renames the database.
	ITEM	REN ITEM	Renames a data item.
	SET	REN SET	Renames a data set.
REORDER	FIELD	REO F	Reorders the field sequence in a manual master or detail data set.
	ITEM	REO ITEM	Reorders the sequence in which the data item resides in the database.
	SET	REO SET	Reorders the sequence in which the data set resides in the database.
REPACK	SET	REP SET	Rebuilds a detail data set by removing gaps left by deleted entries and by adjusting pointers.
REVIEW	BLOCKS	REV BLOCK	Displays information about the data sets in the database. Includes data set type, field count, blocking factor, and block length.
	FIELDS	REV F	Displays information about the fields in master and detail data sets. Includes search items, sort items, and linked data sets.
	ITEMS	REV ITEM	Displays information about the data items in the database. Includes attributes and security.
	PASSWORDS	REV PAS	Displays information about the user classes and associated passwords in the database.
	PATHS	REV PAT	Displays information about the paths in a data set. Includes key item, device class, and security.
	SETS	REV SET	Displays information about the data sets in the database. Includes data set type, capacity, blocking factor, block length, device class, and security.
XEQ		X	Executes DBChange Plus commands from a file instead of the standard input device.

---

## ADD FIELD

Adds a field (data item) to a manual master or detail data set.

**Syntax**            A[DD] F[IELD] *DataSetName* *FieldName* [( *NextField*)]

**Parameters**

<i>DataSetName</i>	is the name of the manual master or detail data set to which the field will be added.
<i>FieldName</i>	is the name of the field to be added. It must already exist as a data item in either the change file or the database.
( <i>NextField</i> )	is the existing field that will follow the newly added field in the schema. If this parameter is omitted, the new field becomes the last field in the data set.

**Description**    Use this command to add a field (data item) to a manual master or detail data set. The field must already exist as a data item in either the change file or the database. Use the optional *NextField* parameter to place the field in a specified order in the data set. Otherwise, the newly added field is placed last. Note that the occurrence of a data item in a data set is referred to as a field to distinguish it from the definition of the data item itself.

Use the ADD ITEM command to add a new data item to the database.

**Example**            >add field inventory store# (supplier)

In this example, the field STORE# is added to the INVENTORY detail data set. The existing field, SUPPLIER, will appear directly after the newly added field, STORE#, in the schema listing for the data set.

Before the addition, the schema entry for the INVENTORY detail data set looks like this:

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (PRODUCT      ),
           ONHANDQTY      ,
           SUPPLIER       (!SUPMASTER ), <---- field to be added above
           UNIT-COST      ,
           SUPPLIER
           LASTSHIPDATE   ( DATE-MASTER),
           BINNUM         ;
CAPACITY:  450(15);
```

## ADD FIELD

After the addition, the schema entry for the INVENTORY detail data set looks like this:

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (PRODUCT      ),
           ONHANDQTY      ,
           STORE#         , <---- new field added
           SUPPLIER       (!SUPMASTER  ),
           UNIT-COST      ,
           LASTSHIPDATE   ( DATE-MASTER),
           BINNUM         ;
CAPACITY:  450(15);
```

---

## ADD ITEM

Adds a data item to the database.

### Syntax

```
A[DD] ITEM ItemName [( NextItem)] [ SubitemCount] SubitemType SubitemLength
[[ [ ReadClassList]/[ WriteClassList]]
```

<b>Parameters</b>	<i>ItemName</i>	is the name of the data item to be added. Each data item within the database must have a unique name. This name must be from 1 to 16 characters, the first of which must be alphabetic. Characters after the first must be chosen from this set: letters A - Z, digits 0 - 9, and + * / ? ' # % & @ only.
	( <i>NextItem</i> )	is the existing data item that follows the newly added data item in the schema. If this parameter is omitted, the new data item is placed last in the ITEM part of the schema.
	<i>SubitemCount</i>	must be an integer from 1 to 255. For data item types U, X, or Z, the product of the subitem count and the subitem length must equal an even number. For data item type P, the product of the subitem count and the subitem length must be evenly divisible by 4. If this parameter is omitted, the subitem count is assumed to be 1.
	<i>SubitemType</i>	must be a valid data item type: I, J, K, R, U, X, Z, or P.
	<i>SubitemLength</i>	has a maximum of 255 for data item types U, X, Z, and P. The length for types I or J must be 1, 2, or 4. For type K it must be 1 or 2, and for type R it must be 2 or 4. For data item types U, X, or Z, the product of the subitem count and the subitem length must equal an even number. For data item type P, the product of the subitem count and the subitem length must be evenly divisible by 4.
	( <i>ReadClassList</i> / <i>WriteClassList</i> )	defines user class access to the data item. If this entire parameter is omitted (an absent list), all user

classes can read the data item unless access is further restricted by data set security. Note that access to a data item is only possible when the data item is a field in a data entry. The two class lists are defined as follows:

*(ReadClassList)* is a list of the user classes that have read-only access to the added item. It can consist of user class numbers from 0 to 63, separated by commas. If this parameter is empty [for example, (/12,14)], no user classes have read access to the data item unless access is allowed by data set security. However, the database creator can read the data item.

*(WriteClassList)* is a list of the user classes that have write (and implied read) access to the data item. It can consist of user class numbers from 0 to 63, separated by commas. If this parameter is empty [for

## ADD ITEM

example,  
(12,14/)], no  
user class has  
write access to  
the data item  
unless access  
is allowed  
by data set  
security.  
However,  
the database  
creator can  
write to the  
data item.

**Description** Use this command to add a new data item to the database.

When defining data item security, note that the absence of a *ReadClassList/WriteClassList* (an absent list), allows all user classes to read the data item unless prevented by data set security, while the null (or empty) list (/), prevents all user classes from accessing the data item unless allowed by data set security. Note also that data set security can further restrict or enhance user class access to data items.

**Example** `>add item store# j 2 (11,12/14)`

In this example, a new data item STORE# is added to the database. The data item is of type J and has a subitem length of 2. User classes 11 and 12 have read access and user class 14 has write access to the new data item. Because *NextItem* and *SubitemCount* parameters are not specified, the new data item appears at the end of the ITEMS part of the schema and has a subitem count of 1.

Before the addition, the ITEMS part of the schema looks like this:

```
ITEMS :
  ACCOUNT      , J2(0,11,12,13,14,18/);
  BINNUM       , Z2(/13);
  :
  TOTAL        , J2(11,14);
  UNIT-COST    , P8(/12);
  ZIP          , X6(12,13,14/11);    <---- item to be added after ZIP
```

## ADD ITEM

After the addition, the ITEMS part of the schema looks like this:

```
ITEMS:
  ACCOUNT      , J2(0,11,12,13,14,18/);
  BINNUM       , Z2(/13);
  :
  TOTAL        , J2(11,14);
  UNIT-COST    , P8(/12);
  ZIP          , X6(12,13,14/11);
  STORE#       , J2(11,12/14);      <---- new item added
```

---

## ADD PASSWORD

Adds a password to the database.

**Syntax**            A[DD] PAS[SWORD] *Password UserClass*

**Parameters**    *Password*            is the new password. Passwords are case sensitive and are not upshifted like other schema names. Therefore, an uppercase character is distinguished from a lowercase character. A password must be from 1 to 8 characters, excluding carriage returns, slashes, semicolons, and blanks.

*UserClass*            is the user class number associated with the new password. A user class number from 1 to 63 must be associated with one password in the database and must be unique within the PASSWORD part of the schema. This number is used to define data item and data set security associated with the password.

**Description**    Use this command to add a new password to the database. Both the *Password* and *UserClass* parameters are required.

**Example**            >add password MANAGER 20

In this example, the new password “MANAGER” is added. The user class number 20 is associated with this new password and it, instead of the password, is used in the data item and data set read and write class lists to define data item and data set security. Note that the user must enter “MANAGER” exactly as it appears; “manager” or “Manager” are not equivalent.

Before the addition, the PASSWORDS part of the schema looks like this:

```
PASSWORDS :
  11 CREDIT      ;
  12 BUYER       ;
  13 SHIP-REC    ;
  14 CLERK       ;
  18 DO-ALL      ;    <---- password to be added after DO-ALL
```

## ADD PASSWORD

After the addition, the PASSWORDS part of the schema looks like this:

```
PASSWORDS :  
  11 CREDIT ;  
  12 BUYER ;  
  13 SHIP-REC ;  
  14 CLERK ;  
  18 DO-ALL ;  
  20 MANAGER ; <---- password added
```

---

## ADD PATH

Adds a path to a detail data set.

### Syntax

```
A [DD] PAT[H] DetailDataSetName SearchItemName [!] MasterDataSetName [ SortItemName]
```

<b>Parameters</b>	<i>DetailDataSetName</i>	is the name of the detail data set that will contain the new path.
	<i>SearchItemName</i>	is the name of the search item for the new path.
	!	indicates that the new path is to be the primary path. Only one path in a detail data set can be designated as the primary path. If this parameter is omitted, a primary path previously designated will remain in effect. If no primary path has been designated, the first unsorted path listed for the data set in the schema becomes the primary path by default. No space is allowed between ! and <i>MasterDataSetName</i> .
	<i>MasterDataSetName</i>	is the master data set related to the search item. It must be an existing data set and must be listed in the schema before the specified detail data set containing.
	<i>SortItemName</i>	is the name of the sort item for the new path. If this parameter is omitted, the entries will not be sorted. The sort item must be an existing field in the detail data set.

**Description** Use this command to add a new path to a detail data set. Place an exclamation point (!) before the name of the master data set to define the path as the primary path. The primary path should be the path in the data set most frequently accessed. A newly defined primary path replaces any existing primary path.

Note that a master data set must be listed in the schema before any related detail data sets. Use the REORDER SET command if it is necessary to reorder data sets.

**Example**

```
>add path inventory stock# !product
```

In this example, a new path is added to the detail data set INVENTORY and is designated as the primary path. PRODUCT is the related master data set. No sort item has been specified for the path. Use the REORDER SET command if it is necessary to reorder data sets.

## ADD PATH

Before the addition, the schema entry for the INVENTORY data set looks like this (note that the path to SUP-MASTER is the existing primary path):

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         , <---- path to be added
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER ), <---- path to be changed
           UNIT-COST      ,
           LASTSHIPDATE  (DATE-MASTER ),
           BINNUM         ;
CAPACITY:  450(15);
```

After the addition, the schema entry for the INVENTORY data set looks like this:

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (!PRODUCT   ), <---- primary path added
           ONHANDQTY      ,
           SUPPLIER       (SUP-MASTER ), <---- path no longer primary
           UNIT-COST      ,
           LASTSHIPDATE  (DATE-MASTER ),
           BINNUM         ;
CAPACITY:  450(15);
```

---

## ADD SET

Adds a new master or detail data set to the database.

### Syntax

$$A[DD]SET \textit{DataSetName} [( \textit{NextSet} )] \left\{ \begin{array}{l} A \textit{KeyItem} \\ M \textit{KeyItem} \\ D \end{array} \right\}$$
$$[( [ \textit{ReadClassList} ] / [ \textit{WriteClassList} ] )] \textit{SetCapacity}$$
$$[( \textit{BlockFactor} )] [ \textit{DeviceClass} ]$$

### Parameters

*DataSetName*

is the name of the data set to be added. This name must be from 1 to 16 characters, the first of which must be alphabetic. Characters after the first must be chosen from this set: letters A - Z, digits 0 - 9, and + \* / ? ' # % & @ only.

*(NextSet)*

is the data set that follows the newly added data set in the schema. If this parameter is omitted, the new data set is placed last in the SETS part of the schema. Detail data sets must be defined after any master data sets related by a shared path.

A

is used if you are adding an automatic master data set.

M

is used if you are adding a manual master data set.

D

is used if you are adding a detail data set.

*KeyItem*

is the key item for a master data set. This parameter is required when adding a master data set and cannot be used when adding a detail data set.

*(ReadClassList/  
WriteClassList)*

defines user class access to the data set. If this entire parameter is omitted (an absent list), all user classes can read the data set. (Data item security may allow update access to individual fields within the data set if the data item has read access to the data set and is in the write class list of the data item.) The class lists are defined below:

*(ReadClassList)*

is a list of user classes having read access to

## ADD SET

the new data set. It can consist of user class numbers from 0 to 63, separated by commas. If this parameter is empty [for example, (/12,14)], only users with write access can read the data set. However, the database creator can read the data set.

*(WriteClassList)*

is a list of user classes that have write access to the new data set. It can consist of user class numbers from 0 to 63, separated by commas. If this parameter is empty [for example, (12,14/)], no user class has write access to the data set. However, the database creator can write to the data set.

*SetCapacity*

must be less than  $2^{31} - 1$  (2,147,483,647).

*(BlockFactor)*

is the new block factor for the data set. If this parameter is omitted, DBSCHEMA calculates the optimal block factor for a block size of 512

## ADD SET

words or for a block size specified by the DBSchema \$CONTROL command or by the CHANGE BLOCKMAX command.

*DeviceClass*

is the disk on which the data set will reside. If this parameter is omitted, DBSCHEMA defaults to device class DISK.

### Description

Use this command to add a new data set to the database. When adding master data sets, the *KeyItem* parameter must be used. Also remember that all master data sets must appear in the schema before their related detail data sets. Use the REORDER SET command if it is necessary to reorder data sets.

Use the ADD FIELD command on detail and manual master data sets and the ADD PATH command on detail data sets to complete the definition of the newly created data set.

### Data Set Security

When defining data set security, note that the absence of a *ReadClassList/WriteClassList* (an absent list) allows all user classes to read the data set, while the null (or empty) list (/), prevents everyone except the database creator from accessing the data set. Note that user classes with read-only access to a data set may be allowed to update fields (data items) within the data set if they are allowed write access to the data item.

### Example

```
>add set binnum-master (inventory) A binnum (14/11,18) 221 disk2
```

In this example, a new automatic master data set, BINNUM-MASTER, is added to the database. It has a capacity of 221. DISK2 is the device class for the newly added data set. User class 14 has read access to the data set, and user classes 11 and 18 have write access (and implied read access). Because the *NextSet* parameter is used, the data set appears directly before the INVENTORY data set in the schema. The ADD PATH command can now be used to define paths for the data set.

Before the addition, the SETS part of the schema looks like this:

**ADD SET**

SETS:

NAME: CUSTOMER ,M(14/11,18);  
 ENTRY: ACCOUNT (1) ,  
 LAST-NAME ,  
 FIRST-NAME ,  
 INITIAL ,  
 STREET-ADD ,  
 CITY ,  
 STATE ,  
 CREDIT-RATING ;  
 CAPACITY: 221(10);

NAME: DATE-MASTER ,A(0,11,12,13,14,18/)  
 ENTRY: DATE (3)  
 CAPACITY: 211(9);

NAME: PRODUCT ,M(13,14/12,18)  
 ENTRY: STOCK# (2)  
 DESCRIPTION  
 CAPACITY: 307(16)

NAME: SALES ,D(11/14,18)  
 ENTRY: ACCOUNT ( CUSTOMER (PURCH-DATE) )  
 STOCK# (!PRODUCT )  
 QUANTITY  
 PRICE  
 TAX  
 TOTAL  
 PURCH-DATE ( DATE-MASTER )  
 DELIV-DATE ( DATE-MASTER )  
 CAPACITY: 504(14);

NAME: SUP-MASTER ,M(13/12,18);  
 ENTRY: SUPPLIER (1) ,  
 STREET-ADD ,  
 CITY ,  
 STATE ,  
 ZIP ;  
 CAPACITY: 211(12);

<---- data set to be added  
 before INVENTORY

NAME: INVENTORY ,D(12,14/13,18);  
 ENTRY: STOCK# (PRODUCT) ,  
 ONHANDQTY ,  
 SUPPLIER (!SUP-MASTER) ,  
 UNIT-COST ,  
 LASTSHIPDATE ( DATE-MASTER ) ,

**ADD SET**

```
          BINNUM          ;  
CAPACITY: 450(15);  
  
END.
```

**ADD SET**

After the addition, the SETS part of the schema looks like this:

SETS:

NAME: CUSTOMER ,M(14/11,18);

ENTRY: ACCOUNT (1) ,  
 LAST-NAME ,  
 FIRST-NAME ,  
 INITIAL ,  
 STREET-ADD ,  
 CITY ,  
 STATE ,  
 CREDIT-RATING ;

CAPACITY: 221(10);

NAME: DATE-MASTER ,A(0,11,12,13,14,18/);

ENTRY: DATE (2) ;

CAPACITY: 211(19);

NAME: PRODUCT ,M(13,14/12,18);

ENTRY: STOCK# (1) ,  
 DESCRIPTION ;

CAPACITY: 307(16);

NAME: SALES ,D(11/14,18);

ENTRY: ACCOUNT ( CUSTOMER (PURCH-DATE) ),  
 STOCK# (!PRODUCT ) ,  
 QUANTITY ,  
 PRICE ,  
 TAX ,  
 TOTAL ,  
 PURCH-DATE ( DATE-MASTER ) ,  
 DELIV-DATE ( DATE-MASTER ) ;

CAPACITY: 504(14);

NAME: SUP-MASTER ,M(13/12,18);

ENTRY: SUPPLIER ,  
 STREET-ADD ,  
 CITY ,  
 STATE ,  
 ZIP ;

CAPACITY: 211(12);

NAME: BINNUM-MASTER ,A(14/11,18);

ENTRY: BINNUM ;

CAPACITY: 221(10);

<---- data set added

**ADD SET**

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (PRODUCT                ),
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER      ),
           UNIT-COST      ,
           LASTSHIPDATE ( DATE-MASTER      ),
           BINNUM        ;
CAPACITY:  450(15);

END.
```

---

**ADD SORT**

Adds a new sort item to a search item of a path.

**Syntax**

```
A[DD] SO[RT] DetailDataSetName SearchItemName SortItemName
```

**Parameters**

*DetailDataSetName* is the name of the detail data set for the new sort item.

*SearchItemName* is the name of the search item whose chains are sorted based on the new sort item values.

*SortItemName* is the name of the new sort item. It must be an existing field in the data set.

**Description**

Use this command to add a new sort, or to replace an existing sort item, to a search item of a path. All parameters are required. If a sort item already exists for the path, the new sort item replaces it. The sort item must be a data item other than the current search item for that path. Only data items types U, K, or X can be sort items.

**Example**

```
>add sort sales account purch-date
```

In this example, the sort item PURCH-DATE is added to the path defined by the search item ACCOUNT in the detail data set SALES.

Before the addition, the SALES data set entry in the schema looks like this:

```
NAME: SALES ,D(11/14,18);
ENTRY: ACCOUNT ( CUSTOMER ), <---- sort item to be added
      STOCK# (!PRODUCT ),
      QUANTITY ,
      PRICE ,
      TAX ,
      TOTAL ,
      PURCH-DATE ( DATE-MASTER ),
      DELIV-DATE ( DATE-MASTER );
CAPACITY: 504(14);
```

## ADD SORT

After the addition, the SALES data set entry in the schema looks like this:

```
NAME:      SALES      ,D(11/14,18);
ENTRY:     ACCOUNT   ( CUSTOMER (PURCH-DATE)), <---- sort item added
           STOCK#    (!PRODUCT           ),
           QUANTITY  ,
           PRICE     ,
           TAX       ,
           TOTAL     ,
           PURCH-DATE ( DATE-MASTER     ),
           DELIV-DATE ( DATE-MASTER     );
CAPACITY:  504(14);
```

**BASE**

Specifies the database to be modified.

**Syntax**

$$B [ ASE ] DatabaseName \left[ \begin{array}{l} NEW \\ OLD \\ PURGECF \end{array} \right]$$
**Parameters**

<i>DatabaseName</i>	is the name of the database to be modified. It cannot be qualified with the group and account names.
NEW	creates a new change file ( <i>DatabaseNameCF</i> ). If an old change file exists, it is purged. This parameter is the default.
OLD	uses the existing change file and adds to it.
PURGECF	purges the existing change file without creating a new change file.

**Description**

Use this command to specify the name of the database you want to modify. The BASE command must be entered before any other DBChange commands, except COPY, HELP, OUTPUT, REDO, and XEQ. The BASE command has the three following change file options:

- NEW—used to create a new change file. If an old change file exists it is purged. In interactive mode, if NEW is specified and a change file already exists, a confirmation message is displayed asking if you want to purge the old change file and create a new one. In batch mode, the old change file is purged automatically and a new one is created.
- OLD—used to enter additional changes to an existing change file. In interactive mode, if no change file exists, the DBCPLUS program asks if you want to create a new change file. In batch mode, if no change file exists, an error occurs and all commands after that are invalid.
- PURGECF—used to purge the existing change file without creating a new change file. In interactive mode, you are prompted for confirmation and asked if a new change file should be created after the existing change file is purged.

## BASE

### Example

```
>base orders purgecf
To purge change file, confirm purge [y/N] ? y
Change file ORDERSCF has been purged.

No old change file. Create new file [y/N] ? y
Creating new change file:

Creating new change file: item information records

Creating new change file: set information records

Creating new change file: security table

Creating new change file: control record

>
```

In this example, the existing change file for the ORDERS database is purged. The DBCPLUS program asks you to confirm the purge and returns a message confirming that the change file ORDERSCF has been purged. The DBCPLUS program then asks if you want a new change file created. Messages are returned confirming the creation of the new change file.

## CANCEL COMMANDS

Cancels all changes entered in the change file since the last BASE command.

**Syntax** CA[NCEL] COM[MANDS]

**Description** Use this command to cancel all changes entered since the last BASE command. After issuing CANCEL COMMANDS, the change file returns to the state it was in just after the most recent BASE command. To continue using the DBCPLUS program after issuing CANCEL COMMANDS, you must issue a new BASE command.

### Example

```
>add item ship-cost j 4 (/12,13,14,18)
Addition of data item accepted.
```

```
>add field sales ship-cost
Addition of field accepted.
```

```
>cancel commands
Completed CANCEL COMMANDS, existing change file is intact.
```

```
>base orders
```

```
>
```

In this example, some restructuring commands are canceled and the DBCPLUS program returns the change file to the state it was in when the BASE command was issued.

Before issuing the restructuring commands to the ORDERS change file, the ITEMS part of the schema and the SALES data entry in the schema look like this:

```
ITEMS :
ACCOUNT          , I4(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14/11);
CREDIT-RATING    , R2(/14);
DATE            , X6(11,12,13,14,18/);
DELIV-DATE       , X6(/14);
DESCRIPTION      , X20(0,11,12,13,14,18/);
FIRST-NAME       , X10(14/11);
INITIAL          , U2(14/11);
LAST-NAME        , X16(14/11);
LASTSHIPDATE     , X6(12/);
ONHANDQTY        , J2(14/12);
PRICE            , J2(14/);
PURCH-DATE       , X6(11/14);
QUANTITY         , I1(/14);
```

## CANCEL COMMANDS

```
STATE           , X2(12,13,14/11);
STOCK#          , U8(0,11,12,14,18/);
STREET-ADD     , X26(12,13,14/11);
SUPPLIER       , X16(12,13/);
TAX            , J2(14/);
TOTAL          , J2(11,14/);
UNIT-COST      , P8(/12);
ZIP            , X6(12,13,14/11);
```

SETS:

:

```
NAME:   SALES           ,D(11/14,18);
ENTRY:  ACCOUNT        ( CUSTOMER      (PURCH-DATE    )),
        STOCK#         (!PRODUCT      ),
        QUANTITY       ,
        PRICE          ,
        TAX             ,
        TOTAL          ,
        PURCH-DATE     ( DATE-MASTER  ),
        DELIV-DATE     ( DATE-MASTER  );
CAPACITY: 500(10);
```

Before canceling the restructuring commands, the ITEMS part of the schema and the SALES data set entry in the schema look like this:

ITEMS:

```
ACCOUNT        , I4(0,11,12,13,14,18/);
BINNUM         , Z2(/13);
CITY           , X12(12,13,14/11);
CREDIT-RATING  , R2(/14);
DATE           , X6(11,12,13,14,18/);
DELIV-DATE     , X6(/14);
DESCRIPTION     , X20(0,11,12,13,14,18/);
FIRST-NAME     , X10(14/11);
INITIAL        , U2(14/11);
LAST-NAME      , X16(14/11);
LASTSHIPDATE   , X6(12/);
ONHANDQTY     , J2(14/12);
PRICE          , J2(14/);
PURCH-DATE     , X6(11/14);
QUANTITY       , I1(/14);
STATE         , X2(12,13,14/11);
STOCK#         , U8(0,11,12,14,18/);
STREET-ADD     , X26(12,13,14/11);
SUPPLIER       , X16(12,13/);
TAX           , J2(14/);
```

**CANCEL COMMANDS**

```
TOTAL          , J2(11,14/);
UNIT-COST      , P8(/12);
ZIP            , X6(12,13,14/11);
SHIP-COST      , J4(/12,13,14,18); <---- data item added
```

SETS:

:

```
NAME:   SALES          ,D(11/14,18);
ENTRY:  ACCOUNT        ( CUSTOMER      (PURCH-DATE    )),
        STOCK#         (!PRODUCT      ),
        QUANTITY       ,
        PRICE          ,
        TAX             ,
        TOTAL          ,
        PURCH-DATE     ( DATE-MASTER  ),
        DELIV-DATE     ( DATE-MASTER  ),
        SHIP-COST      ; <---- field added
```

After canceling the restructuring commands, the ITEMS part of the schema and the SALES data set entry in the schema look like this:

```
ITEMS:
ACCOUNT        , I4(0,11,12,13,14,18/);
BINNUM         , Z2(/13);
CITY           , X12(12,13,14/11);
CREDIT-RATING  , R2(/14);
DATE           , X6(11,12,13,14,18/);
DELIV-DATE     , X6(/14);
DESCRIPTION     , X20(0,11,12,13,14,18/);
FIRST-NAME     , X10(14/11);
INITIAL        , U2(14/11);
LAST-NAME      , X16(14/11);
LASTSHIPDATE   , X6(12/);
ONHANDQTY     , J2(14/12);
PRICE          , J2(14/);
PURCH-DATE     , X6(11/14);
QUANTITY       , I1(/14);
STATE          , X2(12,13,14/11);
STOCK#         , U8(0,11,12,14,18/);
STREET-ADD     , X26(12,13,14/11);
SUPPLIER       , X16(12,13/);
TAX            , J2(14/);
TOTAL          , J2(11,14/);
UNIT-COST      , P8(/12);
ZIP            , X6(12,13,14/11); <---- 1
```

## CANCEL COMMANDS

SETS:

:

```
NAME:    SALES           ,D(11/14,18);
ENTRY:   ACCOUNT        ( CUSTOMER      (PURCH-DATE   )),
        STOCK#          (!PRODUCT      ),
        QUANTITY        ,
        PRICE           ,
        TAX              ,
        TOTAL           ,
        PURCH-DATE      ( DATE-MASTER  ),
        DELIV-DATE      ( DATE-MASTER  );
                                <----- 2

CAPACITY: 500(10);
```

<sup>1</sup> *data item addition canceled*

<sup>2</sup> *field addition canceled*

---

**CANCEL ERASE**

Cancels the previous ERASE SET command.

**Syntax**            CAN[CEL] E[RASE] *DataSetName*

**Parameters**    *DataSetName*            is the name of the data set which should *not* be erased.

**Description**    Use this command to cancel a previous ERASE SET command stored in the change file for a specified data set. Note that once DBAPLUS has been run, the erased data set entries cannot be recovered.

**Example**            >erase set inventory  
                      To confirm erase of data set, enter [y/N].y  
  
                      Erase Set issued for data set.  
  
                      >cancel erase inventory  
                      Cancel erase of data set.  
  
                      >

In this example, the ERASE SET command issued for the INVENTORY data set entries is canceled.

---

## CANCEL REPACK

Cancels the previous REPACK SET command.

**Syntax**            CAN[CEL] RE[PACK] *DataSetName*

**Parameters**    *DataSetName*            is the name of the data set which should *not* be repacked.

**Description**    Use this command to cancel a previous REPACK SET command stored in the change file for a specified data set. Note that once DBAPLUS has been run, the REPACK SET command cannot be canceled.

**Example**            >repack set sales chained  
                      REPACK SET issued for data set.

>cancel repack sales  
                      Cancel repack of data set.

                      >

In this example, the REPACK SET command issued for the SALES data set is canceled.

## CHANGE ATTRIBUTES

Changes the subitem count, type, and/or length of a data item.

### Syntax

```
CHA [NGE] A [TTRIBUTES] DataItemName [ SubitemCount] SubitemType SubitemLength
```

<b>Parameters</b>	<i>DataItemName</i>	is the name of the data item whose attributes are changed. Each data item within the database must have a unique name. This name must be from 1 to 16 characters, the first of which must be alphabetic. Characters after the first must be chosen from this set: letters A - Z, digits 0 - 9, and + * / ? ' # % & @ only.
	<i>SubitemCount</i>	must be an integer from 1 to 255. For data item types U, X, or Z, the product of the subitem count and the subitem length must equal an even number. For data item type P, the product of the subitem count and the subitem length must be evenly divisible by 4. If this parameter is omitted, the subitem count is assumed to be 1.
	<i>SubitemType</i>	must be a valid data item type: I, J, R, U, X, Z, or P.
	<i>SubitemLength</i>	must be an integer from 1 to 200 for data item types U, X, Z, and P. The length for types I or J must be 1, 2, or 4. For type K, it must be 1 or 2, and for type R, it must be 2 or 4. For data item types U, X, or Z, the subitem count and the subitem length must be an even number. For data item type P, the product of the subitem count and the subitem length must be evenly divisible by 4.

**Description** Use this command to change the subitem count, type, and/or length of a data item. This command is similar to CHANGE ITEM; however, use the CHANGE ATTRIBUTE command when you only want to change data item attributes but do not want to change the security or order in which the data item appears in the ITEMS part of the schema.

## CHANGE ATTRIBUTES

### Example `>change attributes store# X 4`

In this example, the attributes of the data item STORE# are changed. The subitem type is changed to X and the subitem length to 4. Because the *SubitemCount* parameter is not specified, the subitem count does not change.

Before the change, the ITEMS part of the schema looks like this:

```
ITEMS:
ACCOUNT          , J2(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14,/11);
CREDIT-RATING    , R2(/14);
DATE            , X6(0,11,12,13,14,18/);
DELIV-DATE       , X6(/14);
DESCRIPTION      , X20(0,11,12,13,14,18/);
FIRST-NAME       , X10(14/11);
INITIAL          , U2(14/11);
LAST-NAME        , X16(14/11);
LASTSHIPDATE     , X6(12/);
ONHANDQTY        , J2(14/12);
PRICE            , J2(14/);
PURCH-DATE       , X6(11/14);
QUANTITY         , I1(/14);
STATE            , X2(12,13,14,/11);
STOCK#           , U8(0,11,12,13,14,18/);
STREET-ADD       , S26(12,13,14/11);
SUPPLIER         , X16(12,13/);
TAX              , J2(14/);
TOTAL            , J2(11,14);
UNIT-COST        , P8(/12);
ZIP              , X6(12,13,14/11);
STORE#           , J2(11,12/14);      <---- attributes to be changed
```

## CHANGE ATTRIBUTES

After attributes change, the ITEMS part of the schema looks like this:

```
ITEMS:
ACCOUNT          , J2(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14,/11);
CREDIT-RATING   , R2(/14);
DATE            , X6(0,11,12,13,14,18/);
DELIV-DATE      , X6(/14);
DESCRIPTION      , X20(0,11,12,13,14,18/);
FIRST-NAME      , X10(14/11);
INITIAL         , U2(14/11);
LAST-NAME       , X16(14/11);
LASTSHIPDATE    , X6(12/);
ONHANDQTY       , J2(14/12);
PRICE           , J2(14/);
PURCH-DATE      , X6(11/14);
QUANTITY        , I1(/14);
STATE           , X2(12,13,14,/11);
STOCK#          , U8(0,11,12,13,14,18/);
STREET-ADD      , S26(12,13,14/11);
SUPPLIER        , X16(12,13/);
TAX             , J2(14/);
TOTAL           , J2(11,14);
UNIT-COST       , P8(/12);
ZIP             , X6(12,13,14/11);
STORE#          , X4(11,12/14);           <---- attributes changed
```

---

## CHANGE BLOCKFACTOR

Changes the blocking factor of a data set.

**Syntax**           CHA[NGE] BLOCKF[ACTOR] *DataSetName NewBlockFactor*

**Parameters**    *DataSetName*                    is the name of the data set to be reblocked.

*NewBlockFactor*                is the number of data set records in one block. If zero is specified, it takes the value calculated by DBSCHEMA.

**Description**    Use this command to change the blocking factor of a data set. When selecting a blocking factor, remember that the resulting block length must not be larger than the maximum block length for the data set. That is, if the media record is 6 words and the maximum block length is 85, the blocking factor cannot be greater than 14 ( $85/6=14$ ). The default maximum block size is 512, but it may have been redefined with a DBSCHEMA \$CONTROL command or with a previous CHANGE BLOCKMAX command.

The blocking factor must be a number that can be evenly divided into the capacity.

**Note**



---

If you change the blocking factor to a number that cannot be evenly divided into the capacity, DBSCHEMA (called by the DBAPLUS program) adjusts the capacity to a multiple of the new blocking factor closest to the current capacity.

---

To change the maximum block size for a data set, use the CHANGE BLOCKMAX command.

**Example**            >change blockfactor sales 16

In this example, the blocking factor of the SALES data set is changed to 16.

Before the change, the SALES data set entry in the schema looks like this:

```
NAME:    SALES            ,D(11/14,18);
ENTRY:   ACCOUNT        ( CUSTOMER   ( PURCH-DATE   )),
          STOCK#        (!PRODUCT                    ),
          QUANTITY                                    ,
          PRICE                                     ,
          TAX                                        ,
          TOTAL                                     ,
          PURCH-DATE   ( DATE-MASTER                ),
          DELIV-DATE   ( DATE-MASTER                );
```

**CHANGE BLOCKFACTOR**

CAPACITY: 504(14);

<---- *blocking factor to be changed*

## CHANGE BLOCKFACTOR

After the change, the SALES data set entry in the schema looks like this:

```
NAME:    SALES      ,D(11/14,18);
ENTRY:   ACCOUNT   ( CUSTOMER   ( PURCH-DATE  )),
        STOCK#    (!PRODUCT    ),
        QUANTITY   ,
        PRICE      ,
        TAX        ,
        TOTAL      ,
        PURCH-DATE ( DATE-MASTER ),
        DELIV-DATE ( DATE-MASTER );
CAPACITY: 504(16);                                <---- blocking factor changed
```

## CHANGE BLOCKMAX

Changes the maximum block length of one or more data sets.

### Syntax

$$\text{CHA[NGE]BLOCKM[AX] } \left\{ \begin{array}{l} \textit{DataSetName} \\ \textit{FirstDataSetName/LastDataSetName} \\ \textit{@} \end{array} \right\} \\ \textit{BlockMaxNumber}$$

### Parameters

<i>DataSetName</i>	is the name of the data set to be assigned the new maximum block length.
<i>FirstDataSetName/LastDataSetName</i>	are names that define a range of data sets to be assigned the new maximum block length.
@	is used to change the maximum block length for all the data sets in the database.
<i>BlockMaxNumber</i>	is the new maximum physical block length in words. It must be a number from 128 to 2048 or 0. If 0 is specified, it takes the value calculated by DBSCHEMA.

### Description

Use this command to change the maximum block length of one or more data sets. Multiple CHANGE BLOCKMAX commands may be issued. If multiple commands include overlapping ranges of data sets, the *MaxBlockNumber* in the last CHANGE BLOCKMAX command is in effect for the data set range indicated. For example, if one CHANGE BLOCKMAX command is issued for data sets 1 through 4 and the next CHANGE BLOCKMAX command is issued for data sets 3 through 5, the first *BlockMaxNumber* is in effect for data sets 1 and 2 and the second *BlockMaxNumber* is in effect for data sets 3 through 5. To verify the order of the data sets before using the CHANGE BLOCKMAX command, use the REVIEW BLOCKS or REVIEW SETS command.

The BLOCKMAX must be large enough to accommodate the blocking factor. That is, if the media record is 6 words, and the blocking factor is 14, the BLOCKMAX must be at least 85 words.

To change the blocking factor of the data set, use CHANGE BLOCKFACTOR.

### Example

```
>change blockmax sales 1024
```

In this example, the maximum block length of the SALES data set is changed to 1024.

## CHANGE BLOCKMAX

Before the change, the SALES data set entry in the schema looks like this:

```
$CONTROL BLOCKMAX = 640                                <---- block length to be changed
NAME:   SALES      ,D(11/14,18);
ENTRY:  ACCOUNT   ( CUSTOMER (PURCH-DATE )),
        STOCK#    (!PRODUCT                ),
        QUANTITY                                     ,
        PRICE                                           ,
        TAX                                              ,
        TOTAL                                           ,
        PURCH-DATE ( DATE-MASTER                ),
        DELIV-DATE ( DATE-MASTER                );
CAPACITY: 504(14);
```

After the change, the SALES data set entry in the schema looks like this:

```
$CONTROL BLOCKMAX = 1024                                <---- block length changed
NAME:   SALES      ,D(11/14,18);
ENTRY:  ACCOUNT   ( CUSTOMER (PURCH-DATE )),
        STOCK#    (!PRODUCT                ),
        QUANTITY                                     ,
        PRICE                                           ,
        TAX                                              ,
        TOTAL                                           ,
        PURCH-DATE ( DATE-MASTER                ),
        DELIV-DATE ( DATE-MASTER                );
CAPACITY: 504(14);
```

---

**CHANGE CAPACITY**

Changes the capacity of a data set.

**Syntax**      `CHA[NGE] CA[PACITY] DataSetName NewCapacity`

**Parameters**

*DataSetName*      is the name of the data set whose capacity is to be changed.

*NewCapacity*      is the new capacity of the data set. It must be less than  $2^{31} - 1$  (2,147,483,647), but cannot be less than the current number of entries in the data set.

**Description**      Use this command to change the capacity of a data set. For performance reasons, it is recommended that you do not use a number that is a power of two for the capacity of a master data set. The capacity must be a number that can be evenly divided by the blocking factor. If you change the capacity to a number that cannot be evenly divided by the blocking factor, DBChange Plus adjusts the specified capacity to the nearest multiple of the blocking factor.

**Note**       If a data set capacity is lowered below the greatest number of entries the data set has ever contained (the high-water mark), DBChange Plus displays a recommendation to use the REPACK SET command.

---

**Example**      `>change capacity inventory 503`

In this example, the capacity of the data set INVENTORY is changed from 450 to 503. However, because 503 cannot be evenly divided by the blocking factor of 15, the capacity is automatically adjusted to the nearest multiple of the blocking factor, which is 510.

Before the change, the INVENTORY data set entry in the schema looks like this:

```

NAME:      INVENTORY      ,D(12/13,14,18);
ENTRY:    STOCK#          ( PRODUCT      ),
          ONHANDQTY          ,
          SUPPLIER          (!SUP-MASTER ),
          UNIT-COST          ,
          LASTSHIPDATE      ( DATE-MASTER ),
          BINNUM              ;
CAPACITY: 450(15);                <---- capacity to be changed

```

## CHANGE CAPACITY

After the change, the INVENTORY data set entry in the schema looks like this:

```
NAME:    INVENTORY      ,D(12/13,14,18);
ENTRY:   STOCK#         ( PRODUCT      ),
         ONHANDQTY      ,
         SUPPLIER       (!SUP-MASTER ),
         UNIT-COST      ,
         LASTSHIPDATE   ( DATE-MASTER ),
         BINNUM         ;
CAPACITY: 510(15);           <----- capacity changed
```

## CHANGE DEVICECLASS

Moves a data set from one device (disk) to another.

**Syntax**           CHA[NGE] D[EVICECLASS] *DataSetName NewDeviceClass*

**Parameters**   *DataSetName*           is the name of the data set whose device class is to be changed.

*NewDeviceClass*       is the class of the device where the data set will reside.

**Description**   Use this command to move a data set to another device. A device class can span multiple disk drives.

**Example**        >change deviceclass sales disk2

In this example, the SALES data set is moved from the default class DISK to a disk drive identified by the device class of DISK2.

Before the change, the SALES data set entry in the schema looks like this:

```
NAME:      SALES      ,D(11/14,18);           <---- device class to be changed
ENTRY:     ACCOUNT   ( CUSTOMER ( PURCH-DATE )),
           STOCK#    (!PRODUCT                ),
           QUANTITY                                     ,
           PRICE                                           ,
           TAX                                              ,
           TOTAL                                           ,
           PURCH-DATE ( DATE-MASTER                    ),
           DELIV-DATE ( DATE-MASTER                    );
CAPACITY:  504(14);
```

After the change, the SALES data set entry in the schema looks like this:

```
NAME:      SALES      ,D(11/14,18),DISC2;     <---- device class changed
ENTRY:     ACCOUNT   ( CUSTOMER ( PURCH-DATE )),
           STOCK#    (!PRODUCT                ),
           QUANTITY                                     ,
           PRICE                                           ,
           TAX                                              ,
           TOTAL                                           ,
           PURCH-DATE ( DATE-MASTER                    ),
           DELIV-DATE ( DATE-MASTER                    );
CAPACITY:  504(14);
```

---

## CHANGE ITEM

Changes the subitem count, type, length, security, and/or order of a data item. (For information about supported item type conversions, refer to Table 10-2 later in this section.)

### Syntax

```
CHA[NGE] ITEM DataItemName [( NextItem)] [ SubitemCount] SubitemType SubitemLength  
[[ [ ReadClassList]/[ WriteClassList]]
```

<b>Parameters</b>	<i>DataItemName</i>	is the name of the data item to be changed. It must be an existing data item in the database.
	( <i>NextItem</i> )	is the name of the data item that will follow the changed data item in the schema. If this parameter is omitted, the data item order is not changed.
	<i>SubitemCount</i>	must be an integer from 1 through 255. For data item types U, X, or Z, the product of the subitem count and the subitem length must be an even number. For data item type P, the product of the subitem count and the subitem length must be evenly divisible by 4. The default is 1 subitem. If this parameter is selected, the <i>SubitemType</i> and the <i>SubitemLength</i> parameters are also required. If this parameter is omitted, the subitem count remains unchanged.
	<i>SubitemType</i>	must be a valid data item type: I, J, K, R, U, X, Z, or P. If this parameter is selected, the <i>SubitemLength</i> parameter is also required. If this parameter is omitted, the subitem type remains unchanged.
	<i>SubitemLength</i>	must be an integer from 1 to 255 for data item types U, X, Z, and P. The length for types I or J must be 1, 2, or 4. If this parameter is selected, the <i>SubitemType</i> parameter is also required. If this parameter is omitted, the subitem length remains unchanged.

*(ReadClassList/  
WriteClassList)*

defines user class access to the data set. If this entire parameter is omitted, the data item security remains unchanged. The class lists are defined as follows:

*(ReadClassList)* is a list of user classes having read access to the data set. It can consist of user class numbers from 0 to 63, separated by commas. If this parameter is empty [for example, (/12,14)], read access to the data set remains unchanged.

*(WriteClassList)* is a list of user classes that have write (and implied read) access to the data set. It can consist of user class numbers from 0 to 63, separated by commas. If this parameter is empty [for example (12,14/)], write access to the data set remains unchanged.

## CHANGE ITEM

**Description** Use this command when making several data item changes at the same time.

**Note**



Several parameters are interrelated. For example, if you enter the subitem count, you must enter the subitem type and the subitem length, even if the last two parameters have not changed. The *SubitemType* and *SubitemLength* parameters are also interrelated. If you enter one, you must enter the other, even if only one has changed.

When making changes to data item security, remember that access to data items within data sets may be further allowed or restricted by data set security.

### Item Type Conversions

DBChange Plus supports type conversion for all data item types defined in the *TurboIMAGE/XL Reference Manual* with the exception of I4 and J4. The following table shows the data item type conversions supported by DBChange Plus. An x in the appropriate box indicates that you can convert the current item type to the new item type.

**Table 10-2. DBChange Plus Supported Item Type Conversions**

Current Item Type	New Item Type							
	I1,I2	J1,J2	K1	P	R2,R4	U	X	Z
I1,I2	x	x	x	x	x	x	x	x
J1,J2	x	x	x	x	x	x	x	x
K1	x	x	x	x	x	x	x	x
P	x	x	x	x		x	x	x
R2,R4	x	x	x		x			
U						x	x	
X						x	x	
Z	x	x	x	x		x	x	x

**Caution**



When converting data item types, overflow, underflow, and truncation may occur if you are reducing data item size. Be sure to verify that data item conversions do not result in loss of data.

**Example**     >change item deliv-date 2 u 6 (/11,14)

In this example, the subitem count of the data item DELIV-DATE is changed to 2, the subitem type to U, and the subitem length to 6. User class 11 has also been given write access to the data item. Note that the subitem length must be entered even though it has not been changed. Because *NextItem* was not entered, the data item order remains the same.

Before the change, the ITEMS part of the schema looks like this:

```

ITEMS:
ACCOUNT          , J2(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14,/11);
CREDIT-RATING   , R2(/14);
DATE            , X6(0,11,12,13,14,18/);
DELIV-DATE      , X6(/14);           <---- item to be changed
DESCRIPTION     , X20(0,11,12,13,14,18/);
FIRST-NAME      , X10(14/11);
INITIAL         , U2(14/11);
LAST-NAME       , X16(14/11);
LASTSHIPDATE    , X6(12/);
ONHANDQTY       , J2(14/12);
PRICE           , J2(14/);
PURCH-DATE      , X6(11/14);
QUANTITY        , I1(/14);
STATE           , X2(12,13,14,/11);
STOCK#          , U8(0,11,12,13,14,18/);
STREET-ADD      , S26(12,13,14/11);
SUPPLIER        , X16(12,13/);
TAX             , J2(14/);
TOTAL           , J2(11,14);
UNIT-COST       , P8(/12);
ZIP             , X6(12,13,14/11);

```

After the change, the ITEMS part of the schema looks like this:

```

ITEMS:
ACCOUNT          , J2(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14,/11);
CREDIT-RATING   , R2(/14);
DATE            , X6(0,11,12,13,14,18/);
DELIV-DATE      , 2U6(/11,14);       <---- item changed
DESCRIPTION     , X20(0,11,12,13,14,18/);
FIRST-NAME      , X10(14/11);
INITIAL         , U2(14/11);
LAST-NAME       , X16(14/11);

```

## CHANGE ITEM

LASTSHIPDATE	, X6(12/);
ONHANDQTY	, J2(14/12);
PRICE	, J2(14/);
PURCH-DATE	, X6(11/14);
QUANTITY	, I1(/14);
STATE	, X2(12,13,14,/11);
STOCK#	, U8(0,11,12,13,14,18/);
STREET-ADD	, S26(12,13,14/11);
SUPPLIER	, X16(12,13/);
TAX	, J2(14/);
TOTAL	, J2(11,14);
UNIT-COST	, P8(/12);
ZIP	, X6(12,13,14/11);
STORE#	, J2(11,12/14);

## CHANGE ITEMSECURITY

Changes user class access to a data item.

**Syntax**      `CHA [NGE] ITEMSE [CURITY] DataItemName UserClassNumber`  
                    $\left. \begin{array}{c} \text{R} \\ \text{W} \\ \text{N} \end{array} \right\}$

**Parameters**    *DataItemName*      is the name of a data item whose security is to be changed.

*UserClassNumber*    is a user class number defined in the PASSWORD part of the schema. User class numbers are used in the schema security lists to define access to each data set and data item.

                  R                    allows read-only access to the data item.

                  W                    allows write access (and implied read access) to the data item.

                  N                    denies any access to the data item.

**Description**    Use this command to change user class access to a data item. To deny all users access, create a null (or empty) list (/), which allows only the creator access to the data item. When making changes to data item security, remember that access to data items within data sets may be further allowed or restricted by data set security.

**Example**            >change itemsecurity quantity 18 w

>change itemsecurity quantity 14 r

>change itemsecurity credit-rating 14 n

In this example, user class 18 is given write access to the data item QUANTITY. User class 14 is given read access to the data item QUANTITY and is denied any access to the data item CREDIT-RATING.

Before the change, the ITEMS part of the schema looks like this:

```
ITEMS:
ACCOUNT          , J2(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14,/11);
CREDIT-RATING   , R2(/14);           <---- security to be changed
DATE            , X6(0,11,12,13,14,18/);
DELIV-DATE      , X6(/14);
```

## CHANGE ITEMSECURITY

```
DESCRIPTION      , X20(0,11,12,13,14,18/);
FIRST-NAME       , X10(14/11);
INITIAL          , U2(14/11);
LAST-NAME        , X16(14/11);
LASTSHIPDATE     , X6(12/);
ONHANDQTY        , J2(14/12);
PRICE            , J2(14/);
PURCH-DATE       , X6(11/14);
QUANTITY         , I1(/14);                <---- security to be changed
STATE            , X2(12,13,14,/11);
STOCK#           , U8(0,11,12,13,14,18/);
STREET-ADD       , S26(12,13,14/11);
SUPPLIER         , X16(12,13/);
TAX              , J2(14/);
TOTAL            , J2(11,14);
UNIT-COST        , P8(/12);
ZIP              , X6(12,13,14/11);
```

After the change, the ITEMS part of the schema looks like this:

```
ITEMS:
ACCOUNT          , J2(0,11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14,/11);
CREDIT-RATING    , R2(/);                <---- security changed
DATE            , X6(0,11,12,13,14,18/);
DELIV-DATE       , X6(/14);
DESCRIPTION      , X20(0,11,12,13,14,18/);
FIRST-NAME       , X10(14/11);
INITIAL          , U2(14/11);
LAST-NAME        , X16(14/11);
LASTSHIPDATE     , X6(12/);
ONHANDQTY        , J2(14/12);
PRICE            , J2(14/);
PURCH-DATE       , X6(11/14);
QUANTITY         , I1(14/18);            <---- security changed
STATE            , X2(12,13,14,/11);
STOCK#           , U8(0,11,12,13,14,18/);
STREET-ADD       , S26(12,13,14/11);
SUPPLIER         , X16(12,13/);
TAX              , J2(14/);
TOTAL            , J2(11,14);
UNIT-COST        , P8(/12);
ZIP              , X6(12,13,14/11);
```

**CHANGE  
PASSWORD**

Changes a password.

**Syntax**CHA[NGE] PAS[SWORD] *OldPassword NewPassword* [ *UserClass*]**Parameters**

*OldPassword* is the password to be changed.

*NewPassword* is the new password. Passwords are case sensitive; that is, uppercase is distinguished from lowercase by TurboIMAGE/XL. A password must be from 1 to 8 characters long. It cannot include carriage returns, slashes (/ or \), semicolons (;), or blanks.

*UserClass* is the user class to be changed. If this parameter is omitted, the user class associated with the old password is changed.

**Description**

Use this command to change a password. The *OldPassword* and *NewPassword* parameters are required.

**Example**

```
>change password CLERK SALESREP
```

In this example, the password CLERK is changed to SALESREP. The user must enter SALESREP exactly as it is; “salesrep” is not equivalent.

Before the change, the PASSWORDS part of the schema looks like this:

```
PASSWORDS:
  11 CREDIT  ;
  12 BUYER   ;
  13 SHIP-REC ;
  14 CLERK   ; <---- password to be changed
  18 DO-ALL  ;
```

After the change, the PASSWORDS part of the schema looks like this:

```
PASSWORDS:
  11 CREDIT  ;
  12 BUYER   ;
  13 SHIP-REC ;
  14 SALESREP ; <---- password changed
  18 DO-ALL  ;
```

---

## CHANGE PRIMARYPATH

Changes the primary path of a detail data set.

**Syntax**           CHA[NGE] PRI[MARYPATH] *DataSetName SearchItemName*

**Parameters**    *DataSetName*           is the name of the detail data set whose primary path is to be changed.

*SearchItemName*       is the name of the search item for the new primary path.

**Description**   Use this command to change the primary path of a detail data set.

**Example**           >change primarypath inventory stock#

In this example, the primary path in the detail data set INVENTORY is changed to the path defined by the STOCK# search item.

Before the change, the INVENTORY data set entry in the schema looks like this:

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (PRODUCT      ), <---- to become primary path
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER ), <---- path to be changed
           UNIT-COST      ,
           LASTSHIPDATE  ( DATE-MASTER ),
           BINNUM         ;
CAPACITY:  450(15);
```

After the change, the INVENTORY data set entry in the schema looks like this:

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (!PRODUCT     ), <---- path now primary path
           ONHANDQTY      ,
           SUPPLIER       ( SUP-MASTER ), <---- path no longer primary
           UNIT-COST      ,
           LASTSHIPDATE  ( DATE-MASTER ),
           BINNUM         ;
CAPACITY:  450(15);
```

**CHANGE  
SETSECURITY**

Changes user class access to a data set.

**Syntax**

$$\text{CHA}[\text{NGE}]\text{SETSE}[\text{CURITY}] \textit{DataSetName} \textit{UserClassNumber} \left\{ \begin{array}{l} \text{R} \\ \text{W} \\ \text{N} \end{array} \right\}$$
**Parameters**

*DataSetName* is the name of a data set whose security is to be changed.

*UserClassNumber* is a user class number defined in the PASSWORDS part of the schema. User class numbers are used in the schema read and write class lists to define access to each data set and data item.

R allows read-only access to the data set.

W allows write access to the data set (and implied read access).

N denies any access to the data set.

**Description**

Use this command to change user class access to a data set. To deny all users access, create a null (or empty) list (/), which allows only the creator to access the data set.

When making changes to data set security, remember that access to data items within data sets may be further enhanced or restricted by data item security.

**Example**

```
>change setsecurity inventory 18 w
>change setsecurity inventory 12 n
```

In this example, user class 18 is given write access (and implied read access) to the data set INVENTORY. User class 12 no longer has any access to the data set.

Before the changes, the INVENTORY data set entry in the schema looks like this:

```
NAME:      INVENTORY    ,D(12,14/13);      <---- security to be changed
ENTRY:     STOCK#      (PRODUCT      ),
           ONHANDQTY    ,
           SUPPLIER    (!SUP-MASTER  ),
           UNIT-COST    ,
           LASTSHIPDATE ( DATE-MASTER ),
           BINNUM      ;
CAPACITY:  450(15);
```

## CHANGE SETSECURITY

After the changes, the data set entry in the schema looks like this:

```
NAME:      INVENTORY      ,D(14/13,18);      <---- security changed
ENTRY:     STOCK#        ( PRODUCT        ),
           ONHANDQTY      ,
           SUPPLIER      (!SUP-MASTER  ),
           UNIT-COST      ,
           LASTSHIPDATE ( DATE-MASTER  ),
           BINNUM        ;
CAPACITY:  450(15);
```

---

**CHECK BASE**

Checks all data sets and all paths in the database for structural integrity and reports any problems.

**Syntax**      CHE[CK] BA[SE] [ *Quick*]

**Parameters**    *Quick*                      is a fast method of checking the database to validate pointer integrity. If you specify the *Quick* parameter, the database is examined using a serial read. Using this option, the command reports if the database has a problem, but it does not indicate the location.

**Description****Note**


---

Before using the CHECK BASE command, read chapter 8, “Maintaining the Database.”

---

Use this command to check the database, or a specified portion of it, for various structural integrity problems. CHECK BASE examines the root file for incorrect path sequences, examines the data sets for inconsistent pointers or bit maps, and examines the paths for broken chains.

The database checking process begins only after you issue the PERFORM COMMANDS command.

**Example 1**

In the example below, the ORDERS database is checked using the CHECK BASE command without the *Quick* parameter. A description of the output follows the example.

```
>base orders
>check base
CHECK BASE (CHECK ROOT, CHECK SET and CHECK PATH) pending for 5 set(s).
>perform commands

PERFORM COMMANDS now or create job file [job/y/N]? y
Please wait ...
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
FRI, MAY 25, 1990, 9:03 AM

Database has been modified since last DBSTORE (DBA 220).
Continue [y/N] ? y

Checking root file.
Starting data set serial reads.
```

**CHECK BASE**

.....  
 Tabulating statistics.

CHECK SET (MASTER)

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0
PRODUCT	M	32	307	10	6.3	0	0.0
SUP-MASTER	M	6	211	3	0.0	0	0.0

CHECK SET (DETAIL)

Detail Set Name	Entries	Capacity	Pct Full	Highest Entry Used	Delete Chain Count
SALES	17	504	3	17	0
INVENTORY	5	510	1	5	0

CHECK PATH

Set Name	Type	Max Chain	Avg Chain	Std Dev	Pct Far Ptrs	Avg Blocks	Packing Ratio
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
PRODUCT	M						
SYNONYM CHAINS		2	1.07	0.25	0	1.00	N/A
SALES	D						
ACCOUNT	( S )	17231	2874.67	7033.14	6	1.17	6.86
STOCK#	( P )	5	2.13	1.64	6	1.13	0.89
SUP-MASTER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
INVENTORY	D						
STOCK#		1	1.00	0.00	0	1.00	1.00
SUPPLIER	( P )	5	5.00	0.00	0	1.00	1.00

Starting data set chained reads.

.....

## CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

```

-----
CUSTOMER          M
  Userlabel entry count does not match synonym chain count (CHK 570).
  Record 6 is not linked into the proper synonym chain (CHK 360).
SALES            D
  ACCOUNT          (Path 1; linked to master CUSTOMER, path 1)
  Path Chain inconsistencies detected (CHK 400).
  Sum of chain counts in the related master set does not
    match bitmap entry count for this set (CHK 580).
  Chainhead record 113 chain count mismatches entries on chain (CHK 450).
  Record 3 contains bad backward pointer (CHK 440).
  Record 14 contains bad backward pointer (CHK 440).
  Record 15 is not linked into the proper chain (CHK 460).
  Record 16 contains bad forward pointer (CHK 430).
  STOCK#           (Path 2; linked to master PRODUCT, path 1)
  Path Chain inconsistencies detected (CHK 400).
  Chainhead record 267 contains bad forward pointer (CHK 410).
  Chainhead record 267 contains bad backward pointer (CHK 420).
  Chainhead record 267 chain count mismatches entries on chain (CHK 450).
  Record 15 is not linked into the proper chain (CHK 460).

```

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

## CHECK BASE

Following is a description of the CHECK BASE output:

### CHECK SET (Master) Output

Master Set Name	is the name of the master data set being checked.
Type	is the type of master data set being checked. A is for automatic master; M is for manual master.
Entries	is the number of entries the data set contains.
Capacity	is the capacity of the data set.
Pct Full	is the percentage of the data set capacity, rounded to the nearest whole percentage, that is not available.
Pct Sec	is the percentage of secondary entries.
Longest Cluster	is the longest number of TurboIMAGE/XL blocks that must be read to find an open slot to place a synonym.
Average Cluster	is the average number of TurboIMAGE/XL blocks that must be read to find an open slot to place a synonym.

### CHECK SET (Detail) Output

Detail Set Name	is the name of the detail data set being checked.
Entries	is the number of entries the data set contains.
Capacity	is the capacity of the data set.
Pct Full	is the percentage of the data set capacity, rounded to the nearest whole percentage, that is not available.
Highest Entry Used	is the record number of the highest entry ever used in the data set.
Delete Chain Count	is the number of records in the delete chain.

### CHECK PATH output

Data Set Name	is the name of data set whose paths are being checked.
---------------	--

Search Item Name	is the name of the search item. Because master data sets do not contain search items, but rather key items, the text SYNONYM CHAINS appears in the field instead. For detail data sets, the search item name appears in the field.
Type	is the data set type. Valid data set types are as follows:  M        is for manual master data sets. A        is for automatic master data sets. D        is for detail data sets.  Other codes that appear in this column are defined as follows:  P        designates primary paths. S        designates sort items.
Max Chain	is the maximum number of entries in the longest synonym or detail chain.
Avg Chain	is the average number of entries per chain.
Std Dev	is the standard deviation, which indicates the accuracy of the Avg Chain statistic. The closer to 0.00 this number is, the more accurate the Avg Chain statistic is.
Pct Far Ptrs	is percent far pointers. This number is the percentage of forward pointers that point outside the current block.
Avg Blocks	is the average number of blocks per chain.
Packing Ratio	is the efficiency of the path. This is calculated by the optimal number of blocks per chain divided by the actual number of blocks per chain.

**CHECK INFORMATION Output**

The CHECK INFORMATION output contains a series of messages containing information about inconsistencies in the database and where they occur; that is, which data set they occur in, which record number, and so on.

## CHECK BASE

**Example 2** In the example below, the ORDERS database is checked using the CHECK BASE command with the *Quick* parameter. A description of the output follows the example.

```
>base orders

>check base quick
CHECK BASE (CHECK ROOT, CHECK SET and CHECK PATH) pending for 5 set(s).

>perform commands
PERFORM COMMANDS now or create job file [job/y/N]? y
Please wait ...
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
FRI, MAY 25, 1990, 9:04 AM

Database has been modified since last DBSTORE (DBA 220).
Continue [y/N] ? y

Checking root file.
Starting data set serial reads.
.....
Tabulating statistics.
```

**CHECK BASE**

CHECK SET (MASTER)

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0
PRODUCT	M	32	307	10	6.3	0	0.0
SUP-MASTER	M	6	211	3	0.0	0	0.0

CHECK SET (DETAIL)

Detail Set Name	Entries	Capacity	Pct Full	Highest Entry Used	Delete Chain Count
SALES	17	504	3	17	0
INVENTORY	5	510	1	5	0

CHECK PATH

Set Name	Type	Max Chain	Avg Chain	Std Dev	Pct Far Ptrs	Avg Blocks	Packing Ratio
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
PRODUCT	M						
SYNONYM CHAINS		2	1.07	0.25	0	1.00	N/A
SALES	D						
ACCOUNT	( S )	17231	2874.67	7033.14	6	1.17	6.86
STOCK#	( P )	5	2.13	1.64	6	1.13	0.89
SUP-MASTER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
INVENTORY	D						
STOCK#		1	1.00	0.00	0	1.00	1.00
SUPPLIER	( P )	5	5.00	0.00	0	1.00	1.00

Starting data set chained reads.

## CHECK BASE

### CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

---

### ROOT FILE

Root file inconsistency in area other than path sequence. This cannot be fixed by DBChange (CHK 220).

CUSTOMER M

Userlabel entry count does not match synonym chain count (CHK 570).

SALES D

ACCOUNT (Path 1; linked to master CUSTOMER, path 1)

Path Chain inconsistencies detected (CHK 400).

Sum of chain counts in the related master set does not match bitmap entry count for this set (CHK 580).

STOCK# (Path 2; linked to master PRODUCT, path 1)

Path Chain inconsistencies detected (CHK 400).

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The output for the CHECK BASE command using the *Quick* parameter is the same as the standard CHECK BASE command; however, a summary version of the standard CHECK INFORMATION output is displayed. Problems are identified, but the specific location and records that need fixing are not isolated.

## CHECK PATH

Checks paths for broken chains and incorrect pointer linkages and reports any problems.

### Syntax

$$\text{CHE[CK]PAT[H] } \left\{ \begin{array}{l} @ \\ \textit{MasterSetName} \\ \textit{DetailSetName} \left\{ \begin{array}{l} ( @ ) \\ (\textit{SearchItemName}) \end{array} \right\} \end{array} \right\}$$

[ *Quick* ]

### Parameters

@	indicates that you want to check all paths in all data sets in the database.
<i>MasterSetName</i>	is the name of the master data set whose synonym chains should be checked.
<i>DetailSetName</i>	is the name of the detail data set whose paths should be checked. The following parameters can be specified with <i>DetailSetName</i> :
(@)	indicates that you want to check all the paths in a detail data set.
( <i>SearchItemName</i> )	is the name of the search item defining the path to be checked.
<i>Quick</i>	is a fast method of checking paths to validate pointer integrity. If you specify the <i>Quick</i> parameter, paths are checked using a serial read. Using this option, the command reports if the paths have problems, but does not indicate the location.

### Description

#### Note



Before using the CHECK PATH command, read chapter 8, “Maintaining the Database.”

Use this command to check paths for broken chains or incorrect pointer linkages. For a master data set, the CHECK PATH command follows and examines synonym chain pointers. For a detail data set, the CHECK PATH command follows and examines detail chain pointers and the chain head pointers in the associated master data sets.

The path checking process begins only after you issue a PERFORM COMMANDS.

## CHECK PATH

**Example 1** In the example below, all the paths in the ORDERS database are checked using the CHECK PATH command without the *Quick* parameter. A description of the output follows the example.

```
>base orders
```

```
>check path @
```

```
Check Path issued for 5 set(s).
```

```
>perform commands
```

```
PERFORM COMMANDS now or create job file [job/y/N]? y
```

```
Please wait ...
```

```
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985  
FRI, MAY 25, 1990, 11:12 AM
```

```
Database has been modified since last DBSTORE (DBA 220).
```

```
Continue [y/N] ? y
```

```
Starting data set serial reads.
```

```
.....
```

```
Tabulating statistics.
```

### CHECK PATH

Set Name	Type	Max	Avg	Std	Pct	Avg	Packing
Search Item	(PS)	Chain	Chain	Dev	Far Ptrs	Blocks	Ratio
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
PRODUCT	M						
SYNONYM CHAINS		2	1.07	0.25	0	1.00	N/A
SALES	D						
ACCOUNT	( S )	17231	2874.67	7033.14	6	1.17	6.86
STOCK#	( P )	5	2.13	1.64	6	1.13	0.89
SUP-MASTER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
INVENTORY	D						
STOCK#		1	1.00	0.00	0	1.00	1.00
SUPPLIER	( P )	5	5.00	0.00	0	1.00	1.00

Starting data set chained reads.

.....

## CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	
-----	
CUSTOMER	M
Userlabel entry count does not match synonym chain count (CHK 570).	
Record 6 is not linked into the proper synonym chain (CHK 360).	
SALES	D
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)
Path Chain inconsistencies detected (CHK 400).	
Sum of chain counts in the related master set does not	
match bitmap entry count for this set (CHK 580).	
Chainhead record 113 chain count mismatches entries on chain (CHK 450).	
Record 3 contains bad backward pointer (CHK 440).	
Record 14 contains bad backward pointer (CHK 440).	
Record 15 is not linked into the proper chain (CHK 460).	
Record 16 contains bad forward pointer (CHK 430).	
STOCK#	(Path 2; linked to master PRODUCT, path 1)
Path Chain inconsistencies detected (CHK 400).	
Chainhead record 267 contains bad forward pointer (CHK 410).	
Chainhead record 267 contains bad backward pointer (CHK 420).	
Chainhead record 267 chain count mismatches entries on chain (CHK 450).	
Record 15 is not linked into the proper chain (CHK 460).	

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

Following is a description of the CHECK PATH output:

Data Set Name is the name of data set whose paths are being checked.

Search Item Name is the name of the search item. Because master data sets do not contain search items, but rather key items, the text SYNONYM CHAINS appear in the field instead. For detail data sets, the search item name appears in the field.

Type is the data set type. Valid data set types are as follows:

M is for manual master data sets.

A is for automatic master data sets.

D is for detail data sets.

## CHECK PATH

Other codes that appear in this column are as follows:

P        designates primary paths.

S        designates sort items.

Max Chain	is the maximum number of entries in the longest synonym or detail chain.
Avg Chain	is the average number of entries per chain.
Std Dev	is the standard deviation, which indicates the accuracy of the Avg Chain statistic. The closer to 0.00 this number is, the more accurate the Avg Chain statistic is.
Pct Far Ptrs	is percent far pointers. This number is the percentage of forward pointers that point outside the current block.
Avg Blocks	is the average number of blocks per chain.
Packing Ratio	is the efficiency of the path. This is calculated by the optimal number of blocks per chain divided by the actual number of blocks per chain.

**Example 2** In the example below, all the paths in the detail data set INVENTORY are checked using the *Quick* parameter. A description of the output follows the example.

```
>base orders

>check path inventory ( @ ) quick
CHECK PATH issued for 1 set(s).

>perform commands
PERFORM COMMANDS now or create job file [job/y/N]? y
Please wait ...
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
FRI, MAY 25, 1990, 11:14 AM

Database has been modified since last DBSTORE (DBA 220).
Continue [y/N] ?y

Starting data set serial reads.
...
Tabulating statistics.

CHECK PATH
```

Pct

**CHECK PATH**

Set Name	Type	Max	Avg	Std	Far	Avg	Packing
Search Item	(PS)	Chain	Chain	Dev	Ptrs	Blocks	Ratio
INVENTORY	D						
STOCK#		1	1.00	0.00	0	1.00	1.00
SUPPLIER	(P )	5	5.00	0.00	0	1.00	1.00

Starting data set chained reads.

## CHECK PATH

### CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

---

No problems were detected by CHECK.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

The output for the CHECK PATH command using the *Quick* parameter is the same as the standard CHECK PATH output; however, a summary version of the standard CHECK INFORMATION output is displayed. Problems are identified, but the specific location and records that need fixing are not isolated.

---

**CHECK SET**

Checks data sets for internal or pointer inconsistency and reports any problems.

**Syntax**       $\text{CHE[CK]SET } \left\{ \begin{array}{l} \textit{DataSetName} \\ @ \end{array} \right\} [\textit{Quick}]$

**Parameters**

<i>DataSetName</i>	is the name of the data set which should be checked.
@	indicates that all data sets in the database should be checked.
<i>Quick</i>	is a fast method of checking data sets to validate pointer integrity. If you specify the <i>Quick</i> parameter, the data sets are examined using a serial read. Using this option, the command reports if the data sets have problems, but it does not indicate the location.

**Description****Note**

Before using the CHECK SET command, read chapter 8, "Maintaining the Database."

---

Use this command to check data sets for internal or pointer inconsistency. CHECK SET only checks for internal or pointer inconsistency within the data set(s) specified. To check chain head pointers in the associated master data set(s), use the CHECK PATH command.

The database checking process begins only after you issue a PERFORM COMMANDS.

**Example 1**      In the example below, the detail data set SALES in the ORDERS database is checked using the CHECK SET command without the *Quick* parameter. A description of the output follows the example.

```
>base orders
>check set sales
CHECK SET issued for 1 set(s).

>perform commands
PERFORM COMMANDS now or create job file [job/y/N]? y
Please wait ...
```

HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985  
FRI, MAY 25, 1990, 12:57 PM

## CHECK SET

Database has been modified since last DBSTORE (DBA 220).

Continue [y/N] ? y

Starting data set serial reads.

Tabulating statistics.

### CHECK SET (DETAIL)

Detail Set Name	Entries	Capacity	Pct Full	Highest Entry Used	Delete Chain Count
SALES	17	504	3	17	0

### CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

---

SALES	D
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)
	Path Chain inconsistencies detected (CHK 400).

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

Following is a description of the CHECK SET output for a detail data set:

Detail Set Name	is the name of the detail data set being checked.
Entries	is the number of entries the data set contains.
Capacity	is the capacity of the data set.
Pct Full	is the percentage of the data set capacity, rounded to the nearest whole percentage, that is not available.
Highest Entry Used	is the record number of the highest entry ever used in the data set.
Delete Chain Count	is the number of records in the delete chain.

**Example 2** In the example below, the master data set CUSTOMER in the ORDERS database is checked using the CHECK SET command without the *Quick* parameter. A description of the output follows the example.

```
>base orders
```

```
>check set customer
```

```
CHECK SET issued for 1 set(s).
```

```
>perform commands
```

```
PERFORM COMMANDS now or create job file [job/y/N]? y
```

```
Please wait ...
```

```
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
FRI, MAY 25, 1990, 12:57 PM
```

```
Database has been modified since last DBSTORE (DBA 220).
```

```
Continue [y/N] ? y
```

```
Starting data set serial reads.
```

```
Tabulating statistics.
```

```
CHECK SET (MASTER)
```

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0

```
CHECK INFORMATION
```

```
Set Name          Type
```

```
  Search Item
```

```
  Message(s)
```

```
-----
CUSTOMER
```

```
      M
```

```
      Userlabel entry count does not match synonym chain count (CHK 570).
```

```
SALES
```

```
      D
```

```
      ACCOUNT          (Path 1; linked to master CUSTOMER, path 1)
```

```
      Sum of chain counts in the related master set does not
```

```
      match bitmap entry count for this set (CHK 580).
```

```
HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985
```

```
>
```

## CHECK SET

Following is a description of the CHECK SET output for a master data set:

Master Set Name	is the name of the master data set being checked.
Type	is the type of master data set being checked. A is for automatic master; M is for manual master.
Entries	is the number of entries the data set contains.
Capacity	is the capacity of the data set.
Pct Full	is the percentage of the data set capacity, rounded to the nearest whole percentage, that is not available.
Pct Sec	is the percentage of secondary entries.
Longest Cluster	is the longest number of TurboIMAGE/XL blocks that must be read to find an open slot to place a synonym.
Average Cluster	is the average number of TurboIMAGE/XL blocks that must be read to find an open slot to place a synonym.

The output for the CHECK SET command using the *Quick* parameter is the same as the standard CHECK SET command; however, a summary version of the standard CHECK INFORMATION output is displayed. Problems are identified, but the specific location and records that need fixing are not isolated.

**CONTROL**

Specifies the setting of several options controlling the way DBChange operates. Automatically adjusts data set capacities.

**Syntax**      CON[ TROL ]  
                   [ LIST  
                   NOLIST  
                   PERCENTFULL *DataSet* *MinFull* *MaxFull* [ *NewFull* ]

<b>Parameters</b>	LIST	writes all commands, their parameters, and any messages to \$STDLIST or the designated output device according to the file equation that is set for DBCOUT. This parameter is the default.
	NOLIST	displays messages and program output only. Commands and their parameters are not written to \$STDLIST. For security reasons, CONTROL NOLIST can be used before the PASSWORD command to prevent passwords from being displayed on batch output.
	PERCENTFULL	indicates how full a data set can get before DBChange Plus flags it as needing a capacity change. The following options can be used with the PERCENTFULL parameter: <ul style="list-style-type: none"> <li><i>DataSet</i>      identifies the name of the data set(s) to be flagged for expansion. The <i>DataSet</i> can be specified in one of the following ways: <ul style="list-style-type: none"> <li><i>DataSetName</i>    is the name of a specific data set to which the CONTROL PERCENTFULL command applies.</li> <li>@MASTERS      indicates that the CONTROL PERCENTFULL command applies to all master data sets in the database.</li> <li>@DETAILS      indicates that the CONTROL PERCENTFULL command applies to all detail</li> </ul> </li> </ul>

## CONTROL

data sets in the database.

@ indicates that the CONTROL PERCENTFULL command applies to all data sets in the database.

*MinFull* is the percent of the data set capacity at which the specified data set(s) is flagged for reduction. If the data set capacity falls below the *MinFull* amount, DBChange Plus immediately prints a warning. Zero is a valid value.

*MaxFull* is the percent of the data set capacity at which the specified data set(s) is flagged for expansion. If the data set exceeds the *MaxFull* value, a warning message is printed.

*NewFull* is the desired percent of capacity for the specified data set(s).

If the data set capacity falls below the *MinFull* parameter or exceeds the *MaxFull* parameter, the data set capacity within the current change file is automatically adjusted to the *NewFull* value. DBChange avoids two's complement values and prime numbers for master data sets and rounds to the nearest block for detail data sets.

If the capacity is expanded, the specified data set file is enlarged. Conversely, if the data set capacity is decreased, the specified data set file is reduced. When reducing detail data set capacity, a serial repack is used.

If this parameter is omitted, the percent of capacity is set at 60 percent for master data sets and 70 percent for detail data sets.

**Description**

Use this command to control the following DBChange operations:

- To suppress commands and their parameters to \$STDLIST during DBChange command execution, whether interactively or batch.
- To determine minimum and maximum data set capacity for reduction or expansion.
- To determine user access to the database while checking is in progress.

The CONTROL command can be issued multiple times. However, each command overrides the previous command. Therefore, you should first specify the most inclusive group of sets being controlled, then the next most inclusive group, and so on. For example, if you want all detail data sets in the ORDERS database to be flagged for expansion at 60 percent of capacity, except the INVENTORY data set, which should be flagged at 80 percent of capacity, use the following command sequence:

```
>control percentfull @ details 0 60
>control percentfull inventory 0 80
```

Because the second CONTROL statement overrides the first, every detail data set, including the INVENTORY data set, is flagged for expansion at 60 percent of capacity. Only the INVENTORY data set is flagged for expansion at 80 percent of capacity. Note that the CONTROL PERCENTFULL parameter used with the *NewFull* option overrides any previous CHANGE CAPACITY commands stored in the change file.

**Note**

Because the most recent DBChange Plus command always takes precedence over the previous command, you should issue the CONTROL command immediately after issuing the BASE command.

To verify data set capacity in the current change file, use the REVIEW SETS command.

**Example**

```
>base orders
>control nolist
>control percentfull @ details 55 85 65

>
```

In this example, the CONTROL command is used to control the ORDERS database. Only messages and program output are sent to \$STDLIST. The percent of the data set capacity at which all the detail data sets are flagged for reduction is 55 percent. The percent of the data set capacity at which all the detail data sets are flagged for expansion is 85 percent. The desired percent of capacity is 65 percent.

---

## COPY

Copies a database.

**Syntax**      `COPY OldBaseName [ NewBaseName] [(REPLACE)]`

**Parameters**

*OldBaseName*      is the name of the existing database to be copied. The *OldBaseName* can be fully qualified using the MPE XL file name conventions of *Name.Group.Account*.

*NewBaseName*      is the name of the new copy of the database. The *NewBaseName* can only be qualified with the MPE XL *Group* name. If this parameter is omitted, the *OldBaseName* is used, and the database is placed in the logon group.account. The same file capability in the target group is required.

(REPLACE)      allows you to overwrite an existing database with the same name as *NewBaseName*. In interactive mode, if this parameter is omitted, you are prompted with a REPLACE confirmation. In batch mode, you are not prompted.

**Description**      Use this command to copy a database. The COPY command copies a database from any account and places it in any group within the current logon account; therefore, you must log on to the system as the database creator in the account where the database copy will reside.

The COPY command cannot be issued while a change file is open, that is, after a BASE command is issued. The copying process must be done independently from all other DBChange Plus restructuring.

**Security**      To successfully copy a database, you must have the following MPE XL security capabilities:

**Table 10-3. Required MPE XL Security Capabilities**

Copying Ability	MPE XL Security
Copying any database from any account	SM or [OP+PM]
Copying any database from within the logon account	AM
Copying any database created by user	Creator

When copying a database from another account with only OP+PM capabilities, you must first pass MPE XL security or have the database released.

### Example

```
:hello kelly.testacct.testgrp
:dbcplus
```

```
HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985
FRI, JUN 15, 1990, 3:00 PM
```

```
>copy orders.shipping.database
```

```
copy ORDERS.SHIPPING.DATABASE ==> ORDERS.TESTGRP.TESTACCT [y/N]? y
```

```
Database ORDERS.TESTGRP.TESTACCT already exists. Replace [y/N]? y
```

```
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
```

```
Data set DATE-MASTER : successfully copied.
```

```
Data set CUSTOMER : successfully copied.
```

```
Data set SUP-MASTER : successfully copied.
```

```
HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985
```

```
>
```

In this example, the database ORDERS is copied from the SHIPPING account and DATABASE group to the current logon group and account (TESTGRP.TESTACCT). You are prompted to proceed with the copy and whether or not to replace the existing ORDERS database. DBAPLUS is automatically invoked to complete the copying process, and the DBChange prompt is returned.

---

## DELETE FIELD

Deletes a field from a manual master or detail data set.

**Syntax**            `D[ELETE] F[IELD] DataSetName FieldName`

**Parameters**    *DataSetName*            is the name of the data set from which the field is to be deleted.  
*FieldName*                is the name of the field to be deleted.

**Description**    Use this command to delete a field from a manual master or detail data set.

---

### Caution



The deletion of a field will cause loss of data.

---

If you are deleting a search item, you must delete the path before you can delete the field. Refer to “DELETE PATH” later in this chapter.

When you delete a field, the ITEMS part of the database schema and other data sets are not affected.

If you are deleting a sort item, you must delete the sort item before deleting the field. Refer to “DELETE SORT” later in this chapter.

**Example**            `>delete field product description`

In this example, the field DESCRIPTION is deleted from the PRODUCT data set.

Before the deletion, the data set PRODUCT data set entry in the schema looks like this:

```
NAME:        PRODUCT            ,M(13,14/12,18);
ENTRY:       STOCK#            (2)                    ,
              DESCRIPTION                                    ; <---- field to be deleted
CAPACITY: 307(16);
```

After the deletion, the data set entry in the schema looks like this:

```
NAME:        PRODUCT            ,M(13,14/12,18);
ENTRY:       STOCK#            (2)                    ; <---- field deleted after STOCK#
CAPACITY: 307(16);
```

**DELETE ITEM**

Deletes a data item and its associated fields and paths from the database.

**Syntax**            D[ELETE] ITEM *DataItemName*

**Parameters**      *DataItemName*            is the name of the data item to be deleted.

**Description**      Use this command to delete a data item from the ITEMS part of the schema and from any data set in which it is used as a field.

**Caution**

The deletion of a data item causes loss of data and, if the data item is a search item, deletion of paths associated with the search item. Any automatic master containing the data item is also deleted. Therefore, use caution when deleting data items, especially if the data item is used as a sort or search item.

In interactive mode you are required to confirm the deletion of the data item; in batch mode no confirmation is required.

If you only want to delete a field from a specific data set, use the DELETE FIELD command. If the data item to be deleted is a sort item, you must first use the DELETE SORT command.

**Example**            >delete item zip

In this example, the data item ZIP is deleted from the database. Before the deletion, the ITEMS and SETS parts of the schema look like this:

```

ITEMS:
  ACCOUNT      , J2(0,11,12,13,14,18/);
  BINNUM       , Z2(/13);
  :
  UNIT-COST    , P8(/12);
  ZIP          , X6(12,13,14/11);           <---- to be deleted

SETS:
  NAME:        CUSTOMER      ,M(14/11,18);
  ENTRY:       ACCOUNT       (1)
  LAST-NAME
  :
  STATE
  ZIP          , <---- to be deleted
  CREDIT-RATING
  ;

```

**DELETE ITEM**

CAPACITY: 221(10);

NAME: DATE-MASTER ,A(0,11,12,13,14,18/);

ENTRY: DATE (3) ;

CAPACITY: 211(9);

NAME: PRODUCT ,M(13,14/12,18);

ENTRY: STOCK# (2) ,

DESCRIPTION ;

CAPACITY: 307(16);

⋮

NAME: SUP-MASTER ,M(13/12,18);

ENTRY: SUPPLIER (1) ,

⋮

ZIP ; <---- to be deleted

CAPACITY: 211(12);

⋮

END.

After the deletion, the ITEMS and SETS parts of the schema look like this:

ITEMS:

ACCOUNT , J2(0,11,12,13,14,18/);

BINNUM , Z2(/13);

⋮

UNIT-COST , P8(/12); <---- ZIP deleted here

SETS:

NAME: CUSTOMER ,M(14/11,18);

ENTRY: ACCOUNT (1) ,

LAST-NAME ;

⋮

STATE , <---- ZIP deleted here

CREDIT-RATING ;

CAPACITY: 221(10);

NAME: DATE-MASTER ,A(0,11,12,13,14,18/);

**DELETE ITEM**

```
ENTRY:      DATE          (3)          ;
CAPACITY:   211(19);

NAME:       PRODUCT      ,M(13,14/12,18);
ENTRY:      STOCK#       (2)          ,
            DESCRIPTION   ;
CAPACITY:   307(16);

:

NAME:       SUP-MASTER   ,M(13/12,18);
ENTRY:      SUPPLIER     (1)          ,

:

            STATE        ; <---- ZIP deleted here
CAPACITY:   211(12);

:

END.
```

---

## DELETE ITEMSECURITY

Deletes all security specifications from a data item.

**Syntax**            D[ELETE] ITEMSE[CURITY] *DataItemName*

**Parameters**    *DataItemName*            is the name of the data item whose security specifications are to be removed.

**Description**    Use this command to delete all security specifications from a data item, which results in an absent list. This command is necessary because without it, you can only delete user classes from the security list, creating a null (or empty) list (/), not an absent list. A null list allows only the database creator access to the data item. An absent list allows any user read access to the data item.

Note that access to data items may be further restricted or allowed by data set security.

**Example**            >delete itemsecurity state

In this example, all security specifications are removed from the data item STATE. This allows all users read access to the data item unless access is restricted by data set security.

Before the deletion, the schema entry for the data item STATE looks like this:

```
STATE, X2(12,13,14/11);            <---- security list to be deleted
```

After the deletion, the schema entry for the data item STATE looks like this:

```
STATE, X2;                        <---- security list deleted
```

**DELETE PASSWORD**

Deletes a password and its associated user class from the database.

**Syntax**            D[ELETE] PAS[SWORD] *Password UserClass*

**Parameters**    *Password*                    is the name of the password to be deleted from the database.  
                   *UserClass*                    is the associated user class to be deleted.

**Description**    Use this command to delete a password and its associated user class from the PASSWORDS section of the schema list.  
 In interactive mode, a confirmation message asks you to confirm the deletion of the password. In batch mode, no confirmation is required.

**Example**            >delete password ship-rec 13

In this example, the password SHIP-REC and its associated user class 13 are deleted from the PASSWORDS section of the ORDERS database.

Before the deletion, the schema looks like this:

```
BEGIN DATABASE ORDERS:
```

```
PASSWORDS:
```

```
 11 CREDIT ;
 12 BUYER ;
 13 SHIP-REC ; <---- 13 to be deleted
 14 CLERK ;
 18 DO-ALL ;
```

After the deletion, the schema looks like this:

```
BEGIN DATABASE ORDERS:
```

```
PASSWORDS:
```

```
 11 CREDIT ;
 12 BUYER ; <---- 13 deleted after 12
 14 CLERK ;
 18 DO-ALL ;
```

---

## DELETE PATH

Deletes a path from a detail data set to a master data set.

**Syntax**            D[ELETE] PAT[H] *DataSetName SearchItemName*

**Parameters**    *DataSetName*            is the name of the detail data set of the path to be deleted.

*SearchItemName*        is the name of the search item defining the path to be deleted.

**Description**    Use this command to delete a path between a detail data set and a master data set. This command also changes the search item to a non-key item. Note that you are not deleting the data item from the data set; you are only deleting the path that the search item defined.

If the deleted path was the primary path in the data set, you may want to redefine a new primary path using the CHANGE PATH command. If you do not, the primary path defaults to the first unsorted path defined for the data set.

If you delete a path that is the only path to an automatic master data set, the automatic master data set is also deleted.

**Example**            >delete path inventory stock#

In this example, the path defined by the STOCK# search item in the detail data set INVENTORY is deleted.

Before the deletion, the schema entry for the data set INVENTORY looks like this:

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (PRODUCT      ),          <----- path to be deleted
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER ),
           UNIT-COST      ,
           LASTSHIPDATE ( DATE-MASTER ),
           BINNUM         ;
CAPACITY:  450(15);
```

## DELETE PATH

After the deletion, the schema entry for the data set INVENTORY looks like this:

```
NAME:    INVENTORY      ,D(12,14/13,18);
ENTRY:   STOCK#         ,                <---- path deleted
         ONHANDQTY      ,
         SUPPLIER       ),
         UNIT-COST      (!SUP-MASTER ),
         LASTSHIPDATE  ( DATE-MASTER ),
         BINNUM         ;
CAPACITY: 450(15);
```

After the deletion, STOCK# is still a data item in the data set INVENTORY; however, it is no longer a search item.

---

## DELETE SET

Deletes a master or detail data set and all associated paths from a database.

**Syntax**            D[DELETE] SET *DataSetName*

**Parameters**      *DataSetName*            is the name of the data set to be deleted.

**Description**      Use this command to delete a data set and its associated paths from a database. The deletion of a data set causes loss of data. If the data set contains the only path to an automatic master, the path and the automatic master are also deleted. Therefore, use caution when deleting data sets. In interactive mode you are required to confirm the deletion of the data set; in batch mode no confirmation is required.

**Example**            >delete set inventory

This example deletes the detail data set INVENTORY and its related path from the database.

Before the deletion, the SETS part of the schema looks like this:

```
SETS :
NAME:      CUSTOMER      ,M(14/11,18);
ENTRY:     ACCOUNT      (1)                ,
          LAST-NAME      ,
          FIRST-NAME     ,
          INITIAL        ,
          STREET         ,
          CITY           ,
          STATE          ,
          ZIP            ,
          CREDIT-RATING  ;
CAPACITY:  221(10);

NAME:      DATE-MASTER  ,A(0,11,12,13,14,18/);
ENTRY:     DATE         (3)                ; <---- path count to change
CAPACITY:  211(9);

NAME:      PRODUCT      ,M(13,14/12,18);
ENTRY:     STOCK#       (2)                , <---- path count to change
          DESCRIPTION    ;
CAPACITY:  307(16);
```

**DELETE SET**

```

NAME:      SALES          ,D(11/14,18);
ENTRY:     ACCOUNT       ( CUSTOMER (PURCH-DATE )),
           STOCK#        (!PRODUCT          ),
           QUANTITY      ,
           PRICE         ,
           TAX           ,
           TOTAL         ,
           PURCH-DATE    ( DATE-MASTER     ),
           DELIV-DATE    ( DATE-MASTER     );
CAPACITY:  504(14);

```

```

NAME:      SUP-MASTER   ,M(13/12,18);
ENTRY:     SUPPLIER     (1)                , <---- path count to change
           STREET-ADD   ,
           CITY         ,
           STATE        ,
           ZIP          ;
CAPACITY:  211(12);

```

```

NAME:      INVENTORY    ,D(12,14/13,18);      <---- to be deleted
ENTRY:     STOCK#       (PRODUCT          ),
           ONHANDQTY    ,
           SUPPLIER     (!SUP-MASTER     ),
           UNIT-COST    ,
           LASTSHIPDATE ( DATE-MASTER     ),
           BINNUM       ;
CAPACITY:  450(15);

```

END.

## DELETE SET

After the deletion, the SETS part of the schema looks like this:

```
SETS:
  NAME:      CUSTOMER      ,M(14/11,18);
  ENTRY:     ACCOUNT      (1)                ,
            LAST-NAME      ,
            FIRST-NAME     ,
            INITIAL        ,
            STREET         ,
            CITY           ,
            STATE          ,
            ZIP            ,
            CREDIT-RATING  ;
  CAPACITY:  221(10);

  NAME:      DATE-MASTER  ,A(0,11,12,13,14,18/);
  ENTRY:     DATE        (2)                ;    <---- path count changed
  CAPACITY:  211(9);

  NAME:      PRODUCT      ,M(13,14/12,18);
  ENTRY:     STOCK#      (1)                ,    <---- path count changed
            DESCRIPTION    ;
  CAPACITY:  307(16);

  NAME:      SALES        ,D(11/14,18);
  ENTRY:     ACCOUNT      ( CUSTOMER (PURCH-DATE ) ),
            STOCK#      (!PRODUCT          ),
            QUANTITY     ,
            PRICE        ,
            TAX          ,
            TOTAL        ,
            PURCH-DATE   ( DATE-MASTER      ),
            DELIV-DATE   ( DATE-MASTER      );
  CAPACITY:  504(14);

  NAME:      SUP-MASTER  ,M(13/12,18);
  ENTRY:     SUPPLIER    ,    <---- path count changed
            STREET-ADD   ,
            CITY         ,
            STATE        ,
            ZIP          ;
  CAPACITY:  211(12);    <---- INVENTORY deleted

                                here
END.
```

**DELETE  
SETSECURITY**

Deletes all security specifications from a master or detail data set.

**Syntax**            D[DELETE] SETSECURITY *DataSetName*

**Parameters**    *DataSetName*            is the name of the data set whose security specifications are to be removed.

**Description**    Use this command to remove the security list from the data set. The absence of a security list (an absent list) allows any user read access to the data set.

Note that access to data items within the data set may be further restricted or allowed by data item security.

**Example**            >delete setsecurity product

This example removes all security specifications from the data set PRODUCT.

Before the deletion, the data set entry in the schema looks like this:

```
NAME:      PRODUCT,      M(14,13/12,18),DISC2;    <---- security list to be deleted
ENTRY:     STOCK#      (2)                ,
           DESCRIPTION                ;
CAPACITY:  300(16);
```

After the deletion, the data set entry in the schema looks like this:

```
NAME:      PRODUCT,      M,DISC2;                <---- security list deleted
ENTRY:     STOCK#      (2)                ,
           DESCRIPTION                ;
CAPACITY:  300(16);
```



---

**ERASE SET**

Erases entries in manual master and detail data sets.

**Syntax**       $ER[ASE]SET \left\{ \begin{array}{l} DetailSetName \\ ManualMasterSetName \end{array} \right\}$

**Parameters**      *DetailSetName*

is the name of the detail data set which contains entries to be erased.

For a standalone detail data set, all entries are erased. If the detail data set is linked to one or more automatic master data sets, all detail data set entries are erased, and the appropriate chain counts and chain pointers of the related automatic master data set entries are set to zero. If the deletion of a detail data set entry results in an automatic master data set entry not being linked to any detail entries, the master data set entry is deleted. If the detail data set is linked to one or more manual master data sets, all detail data set entries are erased, and the appropriate chain counts and chain pointers of the related manual master data set entries are set to zero.

Manual master data set entries are left in the master data set whether or not the master data set entries are linked to any detail data set entries.

You cannot explicitly erase entries in an automatic master data set.

Automatic master data set entries are erased via related detail data set entries.

*ManualMasterSetName*

is the name of the manual master data set which contains entries to be deleted.

For a standalone manual master data set, all entries are erased. For a manual master data set with one or more paths to detail data sets, and the detail entries in all paths are empty (zero count), all entries in the manual master data set are erased.

You cannot delete entries from a manual master data set with one or

## ERASE SET

more paths to detail data sets which contain entries (path counts are not zero).

**Description** Use the ERASE SET command to erase entries in manual master and detail data sets. In session mode, a confirmation message is displayed to confirm the erasure of the data set entries. In batch mode, the data set is erased without confirmation. If you want to cancel the specified erasure, use the CANCEL ERASE command before running the DBAPLUS program.

**Example**

```
>erase set inventory
To confirm erase of data set, enter Y for Yes. y
Erase Set issued for data set.

>
```

In this example, the ERASE SET command erases the entries in the INVENTORY detail data set.

---

**EXIT**

Stops the execution of the DBCPLUS program.

**Syntax**        E[EXIT]

**Description**    Use this command to leave the DBCPLUS program. A series of statistics are displayed about the status of the change file. These statistics are accumulated for every BASE command that is issued while the DBCPLUS program is running. They are displayed when you exit the DBCPLUS program, even if the DBAPLUS program has already applied the changes to the database.

**Example**

>exit

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

\*\*\*\*\* DBCHANGE STATISTICS\*\*\*\*\*

Number of databases prepared for rename:	0
Number of databases prepared for restructure:	1
Number of databases prepared for erase :	0
Number of databases prepared for repack:	0
Number of databases prepared for check :	0
Number of databases prepared for fix :	0

END OF PROGRAM

:

In this example, the EXIT command causes the DBCPLUS program to stop executing.

---

## FIX BASE

Examines all sets and paths in the database and attempts to correct known problems.

**Syntax**      F[IX] BA[SE]

### Description

#### Note



---

Before using the FIX BASE command, read chapter 8, “Maintaining the Database.”

---

Use this command to examine all sets and paths in the database and to correct known problems. The FIX BASE command is equivalent to issuing the FIX SET @ and FIX PATH @ commands simultaneously. The database fixing process begins only after you issue the PERFORM COMMANDS command.

The FIX BASE command requires exclusive access to the database.

#### Example

In the example below, the ORDERS database is fixed using the FIX BASE command. A description of the output follows the example.

```
>base orders

>fix base
  FIX BASE (FIX SET and FIX PATH) pending for 5 set(s).

>perform commands
  PERFORM COMMANDS now or create job file [job/y/N]? y
  Please wait ...
  HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
  FRI, MAY 25, 1990, 2:47 PM

  Database has been modified since last DBSTORE (DBA 220).
  Continue [y/N] ? y

  Checking root file.
  Starting data set serial reads.
  .....
  Tabulating statistics.
```

CHECK SET (MASTER)

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0
PRODUCT	M	32	307	10	6.3	0	0.0
SUP-MASTER	M	6	211	3	0.0	0	0.0

CHECK SET (DETAIL)

Detail Set Name	Entries	Capacity	Pct Full	Highest Entry Used	Delete Chain Count
SALES	17	504	3	17	0
INVENTORY	5	510	1	5	0

CHECK PATH

Set Name	Type	Max Chain	Avg Chain	Std Dev	Pct Far Ptrs	Avg Blocks	Packing Ratio
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
PRODUCT	M						
SYNONYM CHAINS		2	1.07	0.25	0	1.00	N/A
SALES	D						
ACCOUNT	( S )	17231	2874.67	7033.14	6	1.17	6.86
STOCK#	( P )	5	2.13	1.64	6	1.13	0.89
SUP-MASTER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
INVENTORY	D						
STOCK#		1	1.00	0.00	0	1.00	1.00
SUPPLIER	( P )	5	5.00	0.00	0	1.00	1.00

Starting data set chained reads.

...

## FIX BASE

### CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

---

CUSTOMER M  
Userlabel entry count does not match synonym chain count (CHK 570).  
Record 6 is not linked into the proper synonym chain (CHK 360).

SALES D  
ACCOUNT (Path 1; linked to master CUSTOMER, path 1)  
Path Chain inconsistencies detected (CHK 400).  
Sum of chain counts in the related master set does not  
match bitmap entry count for this set (CHK 580).  
Chainhead record 113 chain count mismatches entries on chain (CHK 450).  
Record 3 contains bad backward pointer (CHK 440).  
Record 14 contains bad backward pointer (CHK 440).  
Record 15 is not linked into the proper chain (CHK 460).  
Record 16 contains bad forward pointer (CHK 430).

STOCK# (Path 2; linked to master PRODUCT, path 1)  
Path Chain inconsistencies detected (CHK 400).  
Chainhead record 267 contains bad forward pointer (CHK 410).  
Chainhead record 267 contains bad backward pointer (CHK 420).  
Chainhead record 267 chain count mismatches entries on chain (CHK 450).  
Record 15 is not linked into the proper chain (CHK 460).

### FIX INFORMATION

---

New Key value added for dataset CUSTOMER (FIX 170). Key value:  
INTEGER(I4/J4)=>> 1482184792 / 1482184792

New Key value added for dataset PRODUCT (FIX 170). Key value:  
ASCII( 8)=>> XXXXXXXX

All problems detected have been fixed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The FIX BASE command first checks the entire database. (For a description of the CHECK BASE output, refer to "CHECK BASE" earlier in this chapter). After checking the database, any problems that are detected are fixed and described in the FIX INFORMATION section of the output. In the above example, DBChange Plus adds new key values to the data sets CUSTOMER and PRODUCT. A message is returned verifying the status of the fix process.

---

## FIX PATH

Examines paths for broken chains and incorrect pointer linkages and attempts to correct known problems.

### Syntax

$$F[IX]PAT[H] \left\{ \begin{array}{l} @ \\ MasterDataSetName \\ DetailDataSetName \left\{ \begin{array}{l} ( @ ) \\ (SearchItemName) \end{array} \right\} \end{array} \right\}$$

### Parameters

@ indicates that all paths in all data sets in the database should be examined and corrected.

*MasterDataSetName* is the name of the master data set whose synonym chains should be examined and corrected.

*DetailDataSetName* is the name of the detail data set whose paths should be examined and corrected. *DetailDataSetName* can be specified as follows:

(@) indicates that all paths in the detail data set should be examined and corrected.

(*SearchItemName*) is the name of the search item defining the path to be examined and corrected.

### Description

#### Note




---

Before using the FIX PATH command, read chapter 8, “Maintaining the Database.”

---

Use this command to examine and correct broken chain and data set structural inconsistencies, such as user label, bit map, and delete chain problems.

When used to examine a master data set, FIX PATH checks and attempts to fix the synonym chain pointers in the master data set. When used to examine a detail data set, FIX PATH checks and attempts to fix the path chain pointers in the detail data set and its associated master set(s).

The database fixing process begins only after you issue a PERFORM COMMANDS.

The FIX PATH command requires exclusive access to the database.

## FIX PATH

**Example** In the example below, the FIX PATH command is used to fix all paths in all data sets in the ORDERS database. A description of the command output follows the example.

```
>base orders
>fix path @
  FIX PATH issued for 5 set(s).

>perform commands
  PERFORM COMMANDS now or create job file [job/y/N]? y

  Please wait ...
  HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
  FRI, MAY 25, 1990,  3:13 PM

  Database has been modified since last DBSTORE (DBA 220).
  Continue [y/N] ? y

  Starting data set serial reads.
  .....
  Tabulating statistics.

  CHECK PATH
```

Set Name	Type	Max	Avg	Std	Pct	Avg	Packing
Search Item	(PS)	Chain	Chain	Dev	Far Ptrs	Blocks	Ratio
CUSTOMER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
PRODUCT	M						
SYNONYM CHAINS		2	1.07	0.25	0	1.00	N/A
SALES	D						
ACCOUNT	( S)	17231	2874.67	7033.14	6	1.17	6.86
STOCK#	( P )	5	2.13	1.64	6	1.13	0.89
SUP-MASTER	M						
SYNONYM CHAINS		1	1.00	0.00	0	1.00	N/A
INVENTORY	D						
STOCK#		1	1.00	0.00	0	1.00	1.00
SUPPLIER	( P )	5	5.00	0.00	0	1.00	1.00

```
Starting data set chained reads.
...
```

## CHECK INFORMATION

```

Set Name          Type
  Search Item
  Message(s)
-----
CUSTOMER          M
  Userlabel entry count does not match synonym chain count (CHK 570).
  Record 6 is not linked into the proper synonym chain (CHK 360).
SALES             D
  ACCOUNT         (Path 1; linked to master CUSTOMER, path 1)
  Path Chain inconsistencies detected (CHK 400).
  Sum of chain counts in the related master set does not
    match bitmap entry count for this set (CHK 580).
  Chainhead record 113 chain count mismatches entries on chain (CHK 450).
  Record 3 contains bad backward pointer (CHK 440).
  Record 14 contains bad backward pointer (CHK 440).
  Record 15 is not linked into the proper chain (CHK 460).
  Record 16 contains bad forward pointer (CHK 430).
  STOCK#          (Path 2; linked to master PRODUCT, path 1)
  Path Chain inconsistencies detected (CHK 400).
  Chainhead record 267 contains bad forward pointer (CHK 410).
  Chainhead record 267 contains bad backward pointer (CHK 420).
  Chainhead record 267 chain count mismatches entries on chain (CHK 450).
  Record 15 is not linked into the proper chain (CHK 460).

```

## FIX INFORMATION

```

-----
New Key value added for dataset CUSTOMER (FIX 170). Key value:
  INTEGER(I4/J4)==>> 1482184792 / 1482184792

```

```

New Key value added for dataset PRODUCT (FIX 170). Key value:
  ASCII( 8)==>> XXXXXXXX

```

All problems detected have been fixed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The FIX PATH command first checks all paths and all data sets in the database. (For a description of the CHECK PATH and CHECK SET output, refer to "CHECK BASE" and "CHECK SET" earlier in this chapter). After checking the paths and data sets, any problems that are detected are fixed and described in the FIX INFORMATION section of the output. In the above example, DBChange Plus adds new key values to the data sets CUSTOMER and PRODUCT. A message is returned verifying the status of the fix process.

---

## FIX SET

Examines data sets for internal pointer or structural inconsistency and attempts to correct known problems.

**Syntax**       $F[IX]SET \left\{ \begin{array}{l} @ \\ DataSetName \end{array} \right\}$

**Parameters**    @                    indicates that all data sets in the database should be examined and corrected.

*DataSetName*        is the name of the data set to be examined and corrected.

**Description**    Use this command to examine one or more data sets for internal pointer or structural inconsistency and to correct known problems.

**Note**             Before using the FIX SET command, read chapter 8, “Maintaining the Database.” In addition, because of the FIX SET command limitations as described below, consider using the FIX PATH command instead.

---

FIX SET examines and corrects only the internal pointer consistency of the set(s) specified. It does not correct pointer consistency across sets. For example, it does not correct a pointer from a chain head in a master data set to the first record on the chain in the associated detail data set(s). To do this, use the FIX PATH command.

FIX SET does not repair broken chains. FIX SET only repairs structures that reside entirely within the data set, such as user label, bit map, and delete chains. For information about fixing both broken chains and data set structure, refer to “FIX PATH” earlier in this chapter.

The database fixing process begins only after you issue the PERFORM COMMANDS command.

The FIX SET command requires exclusive access to the database.

**Example** In the example below, all the data sets in the ORDERS database are fixed using the FIX SET command. A description of the output follows the example.

>base orders

>fix set @

FIX SET issued for 5 set(s).

>perform commands

PERFORM COMMANDS now or create job file [job/y/N]? y

Please wait ...  
 HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985  
 TUE, MAY 29, 1990, 8:49 AM

Database has been modified since last DBSTORE (DBA 220).  
 Continue [y/N] ? y

Starting data set serial reads.

.....  
 Tabulating statistics.

CHECK SET (MASTER)

Master Set Name	Type	Entries	Capacity	Pct Full	Pct Sec	Longest Cluster	Average Cluster
CUSTOMER	M	9	221	4	11.1	0	0.0
PRODUCT	M	32	307	10	6.3	0	0.0
SUP-MASTER	M	6	211	3	0.0	0	0.0

CHECK SET (DETAIL)

Detail Set Name	Entries	Capacity	Pct Full	Highest Entry Used	Delete Chain Count
SALES	17	504	3	17	0
INVENTORY	5	510	1	5	0

## FIX SET

### CHECK INFORMATION

Set Name	Type
Search Item	
Message(s)	

---

CUSTOMER	M	Userlabel entry count does not match synonym chain count (CHK 570).
SALES	D	
ACCOUNT	(Path 1; linked to master CUSTOMER, path 1)	Path Chain inconsistencies detected (CHK 400).
		Sum of chain counts in the related master set does not match bitmap entry count for this set (CHK 580).
STOCK#	(Path 2; linked to master PRODUCT, path 1)	Path Chain inconsistencies detected (CHK 400).

### FIX INFORMATION

---

All problems detected have been fixed.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>

The FIX SET command first checks all the master and detail data sets in the database. (For a description of the CHECK SET output, refer to "CHECK SET" earlier in this chapter). After checking the data sets, any problems that are detected are described in the CHECK INFORMATION section of the output. Under FIX INFORMATION, a message is returned verifying the status of the fix process. In the above example, all problems detected are fixed.

---

**HELP**

Provides information about DBChange commands.

**Syntax**      H[ELP] [ *CommandName*]

**Parameters**    *CommandName*      is the name of a command about which you want information. The abbreviated form of the command may be used.

If this parameter is omitted, a brief description of all DBChange Plus commands is displayed.

**Description**    Use this command to display information about DBChange Plus commands.

**Example 1**

```
>help recover
```

```
RECOVER ITEM            Cancels a data item deletion in change file.  
RECOVER SET            Cancels a data set deletion in change file.
```

```
>
```

In this example, information about the RECOVER command is displayed.

## Example 2

```
>help recover item
```

RECOVER ITEM

    Cancels a data item deletion stored in the change file.

Syntax

    REC[OVER] ITEM DataItemName

Parameter

    DataItemName is the name of the data item to be recovered.

Examples

    RECOVER ITEM zip

    REC ITEM state

Related Commands

    ADD ITEM, DELETE ITEM, REVIEW ITEM.

>

    In this example, information about the RECOVER command and the ITEM keyword is displayed.



---

## PERFORM COMMANDS

Invokes the DBAPLUS program to copy, restructure, check, or maintain the database depending on the commands stored in the current change file.

**Syntax** PE[RFORM] COM[MANDS] [JOB] [ *ParmValue*]

**Parameters** JOB creates a job file ( *DatabaseNameJJ*) to enable running the DBAPLUS program at a later time.

For batch mode, this parameter is required. In session mode, if you omit this option, DBChange asks you if changes should be performed online or if a job stream file should be created.

*ParmValue* is a parameter value from 0 to 15 that provides instructions to DBAPLUS during execution. This option is required for batch mode or if the JOB parameter is used. In session mode, if you omit this option, you are prompted for any necessary Y or N responses.

PARM values can be determined by adding them together. For example, if you want both PARM=1 and PARM=4 to take effect, add them together and specify PARM=5. Table 10-4 lists the questions DBAPLUS may need answers to about parameter values.

**Description** Use PERFORM COMMANDS to invoke DBAPLUS to copy, restructure, check, or maintain the database according to the deferred commands stored in the current change file. PERFORM COMMANDS should be the last command used before the EXIT command or before the next BASE command if multiple databases are being restructured.

In session mode, DBChange asks if changes should be performed online or if a job stream file should be created. If you request a job stream file to be created, DBChange creates *FileNameJJ*, which you can later modify and stream. Note that the larger the database, the longer DBAPLUS takes to perform the restructure. Therefore, depending on the size of the database, you may want to create a batch job.

In batch mode, PERFORM COMMANDS only takes effect if the change file has no errors. If errors are encountered since the most recent BASE command, a message stating that the commands cannot be performed because previous errors were encountered is displayed.

Refer to chapter 9, “Applying Changes to the Database Structure,” for more information about the DBAPLUS program.

Table 10-4. DBAPLUS Parameters

PARM Value	You do not have enough disk space to build all necessary temporary data sets before beginning the restructure. Do you want DBAPLUS to continue with the restructure using an alternative restructuring method?	The database has been modified since the last DBSTORE. Do you wish to continue?	Your root file is not in normal DBSCHEMA format. Do you want DBAPLUS to fix it and continue with the restructure?	The change file is normally purged after successful completion of DBAPLUS. Do you want to save the change file?
1				Y
2			Y	
3			Y	Y
4		Y		
5		Y		Y
6		Y	Y	
7		Y	Y	Y
8	Y			
9	Y			Y
10	Y		Y	
11	Y		Y	Y
12	Y	Y		
13	Y	Y		Y
14	Y	Y	Y	
15	Y	Y	Y	Y

## PERFORM COMMANDS

### Example

```
>base orders

>fix base
  Fix Base (Fix Root, Fix Set and Fix Path) issued for 6 set(s).

>perform commands

  PERFORM COMMANDS now or create job file [job/y/N]? y

  Please wait ...
  HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
  THU, JAN 18, 1990, 7:15 AM

  Database has been modified since last DBSTORE (DBA 220).
  Continue [y/N] ? y

  HP36386 (X.00.02) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>
```

In this example, the BASE command is issued for the ORDERS database. The FIX BASE command is issued, followed by PERFORM COMMANDS. DBChange asks if changes should be performed online or if a job stream file should be created. Because the database has been modified since the last DBSTORE, DBAPLUS asks if you want to continue. When the database has been successfully fixed, the DBChange prompt is returned.

---

**PRINT SCHEMA**

Prints the schema of the database, applying any changes stored in the current change file.

**Syntax**           PR[INT] SC[HEMA] *FileName*

**Parameters**    *FileName*                   is the name of an MPE XL disk file to which the schema is printed. Use this parameter interactively or in batch mode. The file must adhere to MPE XL file naming conventions. (Refer to the *MPE XL General Users Reference Manual*.)

**Description**    Use this command to print the database schema. The printed schema reflects any changes stored in the current change file. The schema is printed on \$STDLIST unless output has been redirected by the OUTPUT command, or the *filename* parameter is specified in the PRINT SCHEMA command. If the *filename* parameter is specified, a 60000-line, ASCII MPE XL disk file is created, and the schema is printed to the disk file in a DBSCHEMA readable format. If the schema file does not use the 60000 lines of allotted disk space, the unused disk space is returned to the MPE XL system.

**Example 1**           >print schema salessc  
                      Schema is printed.

In this example, the schema is interactively printed to an MPE XL disk file named SALESSC.

**Example 2**

>print schema ordersz  
Old schema file ORDERSZ exists. Purge and create a new file [y/N]? y  
Schema is printed.

In this example, the schema is interactively printed to an MPE XL disk file named ORDERSZ. Because the file ORDERSZ already exists, DBChange returns a message confirming whether or not to purge and create a new file.

**Example 3**           >print schema ordersz  
                      Duplicate schema file. Cannot process PRINT SCHEMA.

In this example, the schema is printed to an MPE XL disk file named ORDERSZ in batch mode. Because the file ORDERSZ already exists, the new schema file is not created.

## PRINT SCHEMA

**Example 4**      >base orders new  
                 >output lp  
                 >print schema

In this example, the schema for the ORDERS database is directed to a printer.

**Example 5**      >base orders old  
                 >:file dbcout;dev=pp,2,3  
                 >output lp  
                 >print schema  
                 Schema is printed.  
                 >output term

In this example, the schema for the ORDERS database is directed to printer PP. The priority is two, and three copies will be printed.

---

**RECOVER ITEM**

Cancels a DELETE ITEM command stored in the change file.

**Syntax**            REC[OVER] ITEM *DataItemName*

**Parameters**      *DataItemName*            is the name of the data item to be recovered.

**Description**      Use this command to recover a data item, its type and security definitions, and all fields represented by the data item.

The recovery can only take place if DBAPLUS has not yet been run against the current change file. Once DBAPLUS restructures a database, data items can no longer be recovered. In addition, if a new data item has already been added with the same name as the deleted item, then the deleted item cannot be recovered.

If the deleted data item is a field in a data set that has been subsequently changed, the field is not recovered with this command. That is, if any changes to the data set containing the deleted field are stored in the current change file, the field must be recovered by adding it back with the ADD FIELD command. For example, suppose you change a data set capacity and then want to recover a deleted data item that was a field in that data set. Because the data set has been changed, you must add the deleted field, using the ADD FIELD command, to that data set.

**Example**            >delete item zip  
                      >change password CLERK SALESREP  
                      >recover item zip

In this example, the data item ZIP, its type and security definitions, and all fields represented by the data item are recovered, provided the data sets containing the fields have not been changed and DBAPLUS has not yet been run against the current change file.

---

## RECOVER SET

Cancels a DELETE SET command stored in the change file.

**Syntax**            `REC[OVER] SET DataSetName`

**Parameters**    *DataSetName*            is the name of the data set to be recovered.

**Description**    Use this command to recover a data set.

The recovery can only take place if DBAPLUS has not yet been run against the current change file. Once DBAPLUS restructures a database, data sets can no longer be recovered. In addition, if a new data set has already been added with the same name as the deleted data set, then the deleted data set cannot be recovered.

**Example**            >delete set customer  
                      >change password CLERK SALESREP  
                      >recover set customer

In this example, the data set CUSTOMER is recovered.

---

**REDO**

Allows a user to correct or modify the last DBChange command entered.

**Syntax**      R[EDO]

**Description**      Use this command to display the last DBChange command entered. Use the associated subcommands to correct or change the displayed command. The REDO command applies only to the last command entered and is available only in interactive mode. A warning is issued if REDO is used in batch mode.

**Subcommands**

D	deletes the character above the cursor. If D is repeated, each character above each D is deleted. The D and I subcommands can be used in conjunction, to delete characters and then insert new characters.
H	lists all the valid editing subcommands.
I	inserts one or more characters immediately preceding the character above the cursor. The D and I subcommands can be used in conjunction, to delete characters and then insert new characters.
L	lists the current command as it is currently edited.
R	replaces the characters above the cursor with new characters. If one character is entered, the character above the cursor is replaced; if two characters are entered, two characters (the character above the cursor and the character to the right) are replaced; and so on for additional characters. R is the default subcommand and is only required when the first character to be replaced is a D, H, I, L, R, U, or X.
U	restores the line being corrected to its original form.
X	executes the edited version of a command (same as a carriage return).

## REDO

### Example

```
>print shcema                <---- Incorrect command is entered
Invalid command (DBC 508).
>redo                        <---- Request to REDO command
  print shcema                <---- Command is displayed for corrections
    rch                       <---- Error is deleted and replaced by correction
  print schema                <---- Corrected command is displayed
```

In this example, the PRINT SCHEMA command is corrected using the R subcommand.

---

**RENAME BASE**

Renames the database.

**Syntax**      `REN[AME] BA[SE] NewDatabaseName`**Parameters**    *NewDatabaseName*      is the new database name. A database name must be from 1 to 6 alphanumeric characters, the first of which must be alphabetic.**Description**    Use this command to rename the database named in the previous BASE command. Note that the database is physically renamed only after DBAPLUS is run.**Example**

```
>base orders
>rename base order2
```

In this example, the database ORDERS is renamed to ORDER2. Before renaming the database, the schema listing begins like this:

```
BEGIN DATABASE ORDERS;
```

After renaming the database, the schema listing begins like this:

```
BEGIN DATABASE ORDER2;
```

---

## RENAME ITEM

Renames a data item.

**Syntax**           REN[AME] ITEM *OldItemName NewItemName*

**Parameters**    *OldItemName*           is the name of the data item to be renamed.  
*NewItemName*       is the new name of the data item. Each data item within a database must have a unique name. This name must be from 1 to 16 characters, the first of which must be alphabetic. Characters after the first must be chosen from the following set: letters A - Z, digits 0 - 9, and + \* / ? ' # % & @ inclusive. Data item names are upshifted.

**Description**    Use this command to rename a data item. The data item is renamed wherever it appears in the schema, including the data set field lists.

**Example**           >rename item lastshipdate shipdate

This example gives the data item LASTSHIPDATE the new name SHIPDATE.

Before renaming the data item, the ITEMS and SETS parts of the schema looks like this:

```
ITEMS:
ACCOUNT      , J2(0,11,12,13,14,18/);
BINNUM       , Z2(/13);
:
LASTSHIPDATE , X6(12/);           <---- item to be renamed
ONHANDQTY    , J2(14/12);
:
UNIT-COST    ,
LASTSHIPDATE ( DATE-MASTER      ), <---- field to be renamed
BINNUM       ;
CAPACITY: 450(15);

END.
```

## RENAME ITEM

After renaming the data item, the ITEMS and SETS parts of the schema look like this:

```
ITEMS:
  ACCOUNT      , J2(0,11,12,13,14,18/);
  BINNUM       , Z2(/13);
:
  SHIPDATE     , X6(12/);                <----- item renamed
:
          SHIPDATE ( DATE-MASTER      ), <----- field renamed
          BINNUM   ;
CAPACITY: 450(15);

END.
```

---

## RENAME SET

Renames a data set.

**Syntax**            `REN[AME] SET OldSetName NewSetName`

**Parameters**    *OldSetName*            is the name of the data set to be renamed.  
*NewSetName*            is the new name of the data set. Each data set within a database must have a unique name. This name must be from 1 to 16 characters, the first of which must be alphabetic. Characters after the first must be chosen from the following set: letters A - Z, digits 0 - 9, and + \* / ? ' # % & @ inclusive. Data set names are upshifted.

**Description**    Use this command to rename a data set.

**Example**            `>rename set sales local-sales`

In this example, the data set SALES is renamed to LOCAL-SALES.

Before renaming the data set, the data set entry in the schema looks like this:

```
NAME:      SALES      ,D(11/14,18);           <---- data set to be renamed
ENTRY:     ACCOUNT   ( CUSTOMER (PURCH-DATE )),
           STOCK#    (!PRODUCT                ),
           QUANTITY   ,
           PRICE      ,
           TAX        ,
           TOTAL      ,
           PURCH-DATE ( DATE-MASTER           ),
           DELIV-DATE ( DATE-MASTER           ),
CAPACITY:  504(14);
```

After renaming the data set, the data set entry in the schema looks like this:

```
NAME:      LOCAL-SALES ,D(11/14,18);          <---- data set renamed
ENTRY:     ACCOUNT   ( CUSTOMER (PURCH-DATE )),
           STOCK#    (!PRODUCT                ),
           QUANTITY   ,
           PRICE      ,
           TAX        ,
           TOTAL      ,
           PURCH-DATE ( DATE-MASTER           ),
           DELIV-DATE ( DATE-MASTER           ),
CAPACITY:  504(14);
```

**REORDER FIELD**

Reorders the field sequence in a manual master or detail data set.

**Syntax**            REO[RDER] F[IELD] *DataSetName* *FieldName* [ (*NextField*) ]

**Parameters**

<i>DataSetName</i>	is the name of the data set which contains the field to be reordered.
<i>FieldName</i>	is the name of the field to be reordered. It must be an existing field in the data set.
( <i>NextField</i> )	is the existing field that follows the newly reordered field in the field list for the data set. If this parameter is omitted, the reordered field is placed last.

**Description**    Use this command to reorder the field (data item) sequence in a manual master or detail data set field list. The field must already be an existing field in the data set. Use the *NextField* parameter to specify where you want to place the reordered field. If you want the reordered field to appear in last place, omit the *NextField* parameter.

**Example**            >reorder field inventory unit-cost binnum

This example reorders the field UNIT-COST in the INVENTORY data set. The field will appear directly before BINNUM in the field listing for the data set.

Before the reorder, the schema entry for INVENTORY looks like this:

```

NAME:      INVENTORY      ,D(12,14/13);
ENTRY:     STOCK#         (PRODUCT      ),
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER ),
           UNIT-COST      ,                <---- field to be reordered
           LASTSHIPDATE  ( DATE-MASTER ),
           BINNUM         ;
CAPACITY:  450(15);

```

After the reorder, the data set entry in the schema looks like this:

```

NAME:      INVENTORY      ,D(12,14/13);
ENTRY:     STOCK#         ( PRODUCT      ),
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER ),
           LASTSHIPDATE  ( DATE-MASTER ),
           UNIT-COST      ,                <---- field reordered
           BINNUM         ;
CAPACITY:  450(15);

```

---

## REORDER ITEM

Reorders the sequence in which the data item resides in the database.

**Syntax**            `REO[RDER] ITEM DataItemName [( NextDataItem)]`

**Parameters**    *DataItemName*            is the name of the data item to be reordered.  
*(NextDataItem)*            is the data item that follows the newly reordered data item in the schema data item list. If this parameter is omitted, the reordered data item is placed last.

**Description**    Use this command to reorder the data item sequence in the data item part of the schema. The data item must be an existing data item in the database. Use the *NextDataItem* parameter to specify where you wish to place the reordered field. If you wish the reordered data item to appear in last place, omit the *NextDataItem* parameter.

**Example**            `>reorder item store# street-add`

In this example, the data item `STORE#` is reordered to appear before the data item `STREET-ADD` in the schema data item list.

Before the reorder, the schema data item list looks like this:

```
ITEMS:
ACCOUNT           , J2(0,11,12,13,14,18/);
BINNUM            , Z2(/13);
CITY               , X12(12,13,14,/11);
CREDIT-RATING     , R2(/14);
DATE              , X6(0,11,12,13,14,18/);
DELIV-DATE        , X6(/14);
DESCRIPTION        , X20(0,11,12,13,14,18/);
FIRST-NAME         , X10(14/11);
INITIAL            , U2(14/11);
LAST-NAME          , X16(14/11);
LASTSHIPDATE      , X6(12/);
ONHANDQTY         , J2(14/12);
PRICE             , J2(14/);
PURCH-DATE        , X6(11/14);
QUANTITY          , I1(/14);
STATE             , X2(12,13,14,/11);
STOCK#            , U8(0,11,12,13,14,18/);
STREET-ADD        , S26(12,13,14/11);
SUPPLIER          , X16(12,13/);
TAX               , J2(14/);
TOTAL             , J2(11,14);
UNIT-COST         , P8(/12);
```

**REORDER ITEM**

ZIP , X6(12,13,14/11);  
STORE# , J2(11,12/14); <---- *item to be reordered*

After the reorder, the ITEMS part of the schema looks like this:

ITEMS:

ACCOUNT , J2(0,11,12,13,14,18/);  
BINNUM , Z2(/13);  
CITY , X12(12,13,14,/11);  
CREDIT-RATING , R2(/14);  
DATE , X6(0,11,12,13,14,18/);  
DELIV-DATE , X6(/14);  
DESCRIPTION , X20(0,11,12,13,14,18/);  
FIRST-NAME , X10(14/11);  
INITIAL , U2(14/11);  
LAST-NAME , X16(14/11);  
LASTSHIPDATE , X6(12/);  
ONHANDQTY , J2(14/12);  
PRICE , J2(14/);  
PURCH-DATE , X6(11/14);  
QUANTITY , I1(/14);  
STATE , X2(12,13,14,/11);  
STOCK# , U8(0,11,12,13,14,18/);  
STORE# , J2(11,12/14); <---- *item reordered*  
STREET-ADD , S26(12,13,14/11);  
SUPPLIER , X16(12,13/);  
TAX , J2(14/);  
TOTAL , J2(11,14);  
UNIT-COST , P8(/12);  
ZIP , X6(12,13,14/11);

---

## REORDER SET

Reorders the sequence in which the data set resides in the database.

**Syntax**            `REO[RDER] SET DataSetName [( NextDataSet)]`

**Parameters**

<i>DataSetName</i>	is the name of the data set to be reordered. It must be an existing data set in the database.
<i>(NextDataSet)</i>	is the existing data set that follows the newly reordered data set in the schema data set list. If this parameter is omitted, the reordered data set is placed last.

**Description**    Use this command to reorder the data set sequence in the data set part of the schema. It must be an existing data set in the database. Use the *NextDataSet* parameter to specify where you wish to place the reordered data set. If you wish the reordered data set to appear in last place, omit the *NextDataSet* parameter.

**Note**             When reordering data sets, remember that all master data sets must appear in the schema before their related detail data sets.

---

**Example**            `>reorder set sup-master sales`

In this example, the data set SUP-MASTER is reordered to appear before the data set SALES in the schema data set list.

Before the reorder, the schema data set list looks like this:

```
SETS:
  NAME:      CUSTOMER      ,M(14/11,18);
  ENTRY:     ACCOUNT      (1)
             LAST-NAME
             FIRST-NAME
             INITIAL
             STREET-ADD
             CITY
             STATE
             CREDIT-RATING
CAPACITY: 221(10);
```

**REORDER SET**

NAME: DATE-MASTER ,A(0,11,12,13,14,18/)  
 ENTRY: DATE (3)  
 CAPACITY: 211(9)

NAME: PRODUCT ,M(13,14/12,18)  
 ENTRY: STOCK# (2)  
 DESCRIPTION  
 CAPACITY: 307(16)

NAME: SALES ,D(11/14,18)  
 ENTRY: ACCOUNT ( CUSTOMER (PURCH-DATE) )  
 STOCK# (!PRODUCT )  
 QUANTITY  
 PRICE  
 TAX  
 TOTAL  
 PURCH-DATE ( DATE-MASTER )  
 DELIV-DATE ( DATE-MASTER )  
 CAPACITY: 504(14)

NAME: SUP-MASTER ,M(13/12,18); <---- data set to  
 ENTRY: SUPPLIER (1) , be reordered  
 STREET-ADD ,  
 CITY ,  
 STATE ,  
 ZIP ;  
 CAPACITY: 211(12);

NAME: INVENTORY ,D(12,14/13,18);  
 ENTRY: STOCK# (PRODUCT ) ,  
 ONHANDQTY ,  
 SUPPLIER (!SUP-MASTER ) ,  
 UNIT-COST ,  
 LASTSHIPDATE ( DATE-MASTER ) ,  
 BINNUM ;  
 CAPACITY: 450(15);

END.

## REORDER SET

After the reorder, the SETS part of the schema looks like this:

```
SETS:
  NAME:    CUSTOMER      ,M(14/11,18);
  ENTRY:   ACCOUNT      (1)
           LAST-NAME
           FIRST-NAME
           INITIAL
           STREET-ADD
           CITY
           STATE
           CREDIT-RATING
CAPACITY: 221(10);

  NAME:    DATE-MASTER  ,A(0,11,12,13,14,18/);
  ENTRY:   DATE         (2)
CAPACITY: 211(19);

  NAME:    PRODUCT      ,M(13,14/12,18);
  ENTRY:   STOCK#       (1)
           DESCRIPTION
CAPACITY: 307(16);

  NAME:    SUP-MASTER   ,M(13/12,18);           <---- data set reordered
  ENTRY:   SUPPLIER
           STREET-ADD
           CITY
           STATE
           ZIP
CAPACITY: 211(12);

  NAME:    SALES        ,D(11/14,18);
  ENTRY:   ACCOUNT      ( CUSTOMER (PURCH-DATE) ),
           STOCK#       (!PRODUCT           ),
           QUANTITY
           PRICE
           TAX
           TOTAL
           PURCH-DATE   ( DATE-MASTER       ),
           DELIV-DATE   ( DATE-MASTER       );
CAPACITY: 504(14);
```

**REORDER SET**

```
NAME:      INVENTORY      ,D(12,14/13,18);
ENTRY:     STOCK#         (PRODUCT                ),
           ONHANDQTY      ,
           SUPPLIER       (!SUP-MASTER          ),
           UNIT-COST      ,
           LASTSHIPDATE ( DATE-MASTER          ),
           BINNUM         ;
CAPACITY:  450(15);

END.
```



**Description** Use this command to rebuild a detail data set. This command eliminates deleted entry gaps and adjusts pointers.

To verify a pending REPACK SET command, use the REVIEW SETS command. If you want to cancel the data set repack, use the CANCEL REPACK command *before* running the DBAPLUS program.

Refer to chapter 8, “Maintaining the Database,” for more information about repacking data sets.

### Example 1

```
>base orders

>repack set sales serial
Repack Set issued for data set.

>perform commands

PERFORM COMMANDS now or create job file [job/y/N]? y

Please wait ...
HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985
TUE, MAY 29, 1990, 11:10 AM

Database has been modified since last DBSTORE (DBA 220).
Continue [y/N]? y

Data Set SALES      : 17 entries repacked successfully.

HP36386 (A.00.00) DBCHANGE (c) COPYRIGHT Hewlett-Packard Co. 1985

>
```

In this example, the BASE command is issued for the ORDERS database. The REPACK SET command is issued, specifying a serial repack for the data set SALES. To repack the data set interactively, the PERFORM COMMANDS command is issued. The DBCPLUS program asks if changes should be performed online or if a job file should be created. Because the database has been modified since the last DBSTORE, DBAPLUS asks if you wish to continue. When the data set has been successfully repacked, a message is displayed stating the number of entries successfully repacked. The DBChange prompt is returned.

## REPACK SET

### Example 2

>base orders

>repack set inventory chained (lastshipdate) 100 (4)  
REPACK SET issued for data set.

>perform commands

Please wait ... change file is being verified.

PERFORM COMMANDS now or create job file [job/y/N]? y

Please wait ...

HP36386 (A.00.00) DBALTER (c) COPYRIGHT Hewlett-Packard Co. 1985  
WED, MAY 30, 1990, 10:00 AM

Database has been modified since last DBSTORE (DBA 220).

Continue [y/N] ? y

PAGE 1           HEWLETT-PACKARD 30391C.02.00 TurboIMAGE/3000: DBSCHEMA  
                  WED, MAY 30, 1990, 10:00 AM (C) HEWLETT-PACKARD CO. 1987

BEGIN DATABASE ORDERS;

PASSWORDS:

11 CREDIT ;  
12 BUYER ;  
13 SHIP-REC;  
14 CLERK ;  
18 DO-ALL ;

ITEMS:

ACCOUNT       , I4(0,11,12,13,14,18/);  
BINNUM        , Z2(/13);  
:

SETS:

:

```

NAME: INVENTORY ,D(12,14/13,18);
ENTRY: STOCK# ( PRODUCT ),
      ONHANDQTY ,
      SUPPLIER (!SUP-MASTER ),
      UNIT-COST ,
      LASTSHIPDATE ( DATE-MASTER ),
      BINNUM ;
CAPACITY: 100(4); <---- capacity and blocking factor changed

```

END.

:

DATA SET NAME	TYPE	FLD CNT	PT CT	ENTR LGTH	MED REC	CAPACITY	BLK FAC	BLK LGTH	DISC SPACE
DATE-MASTER	A	1	3	3	26	365	19	496	96
CUSTOMER	M	9	1	43	54	201	10	541	112
PRODUCT	M	2	2	14	31	300	16	497	80
SUP-MASTER	M	5	1	31	42	201	12	505	80
INVENTORY	D	6	3	20	32	100	4	129	64 <---- capacity and blocking factor changed
SALES	D	8	4	21	37	500	10	371	160
TOTAL DISC SECTORS INCLUDING ROOT:									624

NUMBER OF ERROR MESSAGES: 0  
ITEM NAME COUNT: 23 DATA SET COUNT: 6  
ROOT LENGTH: 1176 BUFFER LENGTH: 541 TRAILER LENGTH: 256

ROOT FILE ORDERS CREATED.

New database schema is saved as temporary file ORDERSSC  
Temporary file created for set INVENTORY.  
Data set DATE-MASTER : No conversion necessary.  
Data set CUSTOMER : No conversion necessary.  
Data set PRODUCT : No conversion necessary.  
Data set SUP-MASTER : No conversion necessary.  
Data set INVENTORY : 5 entries expected.  
Data set INVENTORY : 5 entries repacked successfully.  
Data set SALES : No conversion necessary.

**REPACK SET**

>

## **REPACK SET**

In this example, the `BASE` command is issued for the `ORDERS` database. The `REPACK SET` command is issued, specifying a chained repack for the detail data set `INVENTORY` on the search item `LASTSHIPDATE`. The capacity is changed to 100 and the blocking factor is changed to 4.

After issuing a `PERFORM COMMANDS` to actually repack the data set, the schema is printed along with messages about the status of the repack. In this case, five entries were successfully repacked.



**Example**

```
>review blocks @
```

```
REVIEW BLOCKS:
```

Data Set Name	Type	Fld Cnt	Pth Cnt	Entr Len	Med Rec	Capacity	Blk Fac	Blk Len	Blk Max
CUSTOMER	M	9	1	43	54	221	10	541	640
DATE-MASTER	A	1	3	3	26	211	19	496	496
PRODUCT	M	2	2	14	31	307	16	498	498
SALES	D	8	4	21	37	504	14	519	640
SUP-MASTER	M	5	1	31	42	211	12	505	505
INVENTORY	D	6	3	20	32	450	15	481	481

```
>
```

In this example, all the data sets in the ORDERS database are reviewed using the REVIEW BLOCKS command.

---

## REVIEW FIELDS

Displays information about the fields (data items) in a master and detail data set.

**Syntax**            `REV[IEW] F[IELDS] DataSetName`

**Parameters**      *DataSetName*            is the name of a data set that contains fields to be reviewed.

**Description**      Use this command to review fields (data items) in a data set. For a master data set, the following information is displayed:

Data Set No.      is the sequence in which the data set appears in the SETS part of the schema.

Name              is the data set name.

Type              is the type of data set. Valid data set types are as follows:

    A              is for automatic master data set.

    M              is for manual master data set.

    D              is for detail data set.

Fld No.            is the sequence in which the field appears in the data set.

Field Name        is the name of the field.

Detail Set Name    is the name of the related detail data set.

Path's Search Item is the name of the key item by which the master data set is linked to a detail data set.

For a detail data set, the information displayed is similar to that of a master data set, except that paths are described along with the names of the related master data sets. The primary path and any sort items are also indicated. All changes stored in the the current change file are included in the REVIEW FIELDS output.

**Example**

```
>review fields product
```

```
REVIEW FIELDS (MASTER SET):
```

```
Data Set No.: 3 Name: PRODUCT Type: M
```

Fld No.	Field Name	Detail Set Name	Path's Search Item
1	STOCK#	SALES INVENTORY	STOCK# STOCK#
2	DESCRIPTION		
3	DISC-PRICE		

```
>
```

In this example, the fields in the master data set PRODUCT are reviewed. The data set PRODUCT contains three fields, STOCK#, DESCRIPTION, and DISC-PRICE. The key item, STOCK#, defines paths to two detail data sets, SALES and INVENTORY.





---

## REVIEW PATHS

Displays information about the paths in a data set.

**Syntax**            `REV[IEW] PAT[HS] DataSetName`

**Parameters**    *DataSetName*            is the name of the master or detail data set whose paths you want to review.

**Description**    Use this command in interactive and batch mode to review the paths in a data set. For detail data sets, the following information is displayed:

Data Set No.            is the sequence in which the data set appears in the SETS part of the schema.

Name                    is the name of the detail data set being reviewed.

Type                    is the type of data set. D is for a detail data set.

Search Item Name        is the name of the search item linking the detail data set to a master data set.

Master Set Name        is the name of the master data set to which the detail data set is associated.

Sort Item Name            is the name of the sort item.

Primary                    indicates the primary path.

For master data sets, the following information is displayed:

Data Set No.            is the sequence in which the data set appears in the SETS part of the schema.

Name                    is the name of the master data set being reviewed.

Type                    is the type of master data set. Valid data set types are as follows:

A                    is for automatic master data set.

M                    is for manual master data set.

Key(Search) Item        is the key item that connects the master data set to the related detail data set.

Detail Set Name        is the name of the related detail data set.

Sort Item Name            is the name of the sort item.

## **REVIEW PATHS**

All changes stored in the the current change file are included in the REVIEW PATHS output. The REVIEW PATHS command also displays status messages if a CHECK PATH or FIX PATH command is pending.

## REVIEW PATHS

### Example 1

```
>review paths inventory
```

```
REVIEW PATHS (DETAIL SET):
```

```
Data Set No.: 6 Name: INVENTORY Type: D
```

```
Search Item Name Master Set Name Sort Item Name Primary
```

```
-----
```

```
STOCK#          PRODUCT
SUPPLIER        SUP-MASTER          Y
LASTSHIPDATE    DATE-MASTER
```

```
>
```

In this example, the paths for the detail data set INVENTORY are reviewed. The data set number is 6. The search item SUPPLIER represents the primary path. No sort items are identified for any of the paths.

### Example 2

```
>review path customer
```

```
REVIEW PATHS (MASTER SET):
```

```
Data Set No.: 2 Name: CUSTOMER Type: M
```

```
Key(Search) Item Detail Set Name Sort Item Name
```

```
-----
```

```
ACCOUNT          SALES          PURCH-DATE
```

```
>
```

In this example, the paths for the master data set CUSTOMER are reviewed. The data set number is 2. The key item is ACCOUNT, the related detail data set is SALES, and the sort item is PURCH-DATE.



## REVIEW SETS

### Example

```
>review sets inventory
```

```
REVIEW SETS:
```

Set					Blk	Blk	Dev	
No.	Data Set Name	Type	Capacity	Fac	Max	Class	Security	
-----								
6	INVENTORY	D	450	15	481	DISK1	(12,14/18)	

```
>
```

In this example, the detail data set INVENTORY is reviewed. It is data set number 6 on device class identified as DISK1 with a capacity of 450 and a blocking factor of 15. The maximum block size is 481. User class 18 has write (and implied read) access and user classes 12 and 14 have read-only access to the data set.

---

**XEQ**

Executes DBChange commands from a file instead of the terminal or batch files.

**Syntax**        *X[EQ] FileName*

**Parameters**    *FileName*                    is the name of an ASCII file containing commands and parameters.

**Description**    Use this command to execute DBChange commands from a file instead of the standard input device. All commands that are accepted by DBChange Plus are allowed in an XEQ file. To create an XEQ file, use a text editor.

When an XEQ command is entered, DBChange reads the specified file and executes the commands until it reaches the end-of-file. The command file must contain all data needed. When the end-of-file is reached, control returns to the original command input device (in session mode, the terminal; or in batch mode, the batch file).

In session mode, during the execution of an XEQ file, the DBCPLUS program displays the contents of the file and any messages on the screen. You are prompted if any changes in the file may have unexpected results or may cause loss of data. If an error occurs when the database is being opened, the DBCPLUS program closes the XEQ file and prompts for another command. If an error occurs when executing a DBChange Plus command, error messages are displayed on the screen. You can then correct the XEQ file and enter the XEQ command again. If the XEQ file contains a PERFORM COMMANDS command, you are prompted for confirmation before the database is restructured.

In batch mode, during the execution of an XEQ file, if an error occurs when the database is being opened, DBChange closes the XEQ file and terminates the job. If an error occurs while executing a DBChange command, the error message is sent to DBCOUT, and XEQ continues processing subsequent commands.

## XEQ

### Example

```
>xeq xeqfile
><<This command file creates a new change file for the ORDERS database >>

><< and stores the addition of the data item REGION in the change file >>

>

>base orders                <----- DBChange command

>delete item date          <----- DBChange command

Deleting this data item will cause loss of data, delete [y/N]? y
Deletion of data item accepted.

>add item region j 2 (11,12/14) <----- DBChange command

Addition of data item accepted.

>

> XEQ file is at end-of-file.

>
```

This example interactively executes the file XEQFILE. Note that, because the deletion of the data item DATE will cause loss of data, DBChange prompts for confirmation.

## DBChange Plus Messages

---

This appendix lists the DBChange Plus error and information messages. In the case of error messages, the cause of the error and user action to resolve the error are supplied for each message.

### Note



All messages are arranged in numerical order. Duplicate message numbers exist. In those cases, refer to the code in parentheses following the message to determine the correct message. For example, (DBC 500) is a DBCPLUS message, and (DBA 500) is a DBAPLUS message.

---

0	MESSAGE	Cannot open message catalog, DBCMSG.SYS (FSERR <i>n</i> ) (DBC 0).
	MEANING	The DBChange Plus message catalog, DBC000.PUB.SYS, is corrupt or does not exist.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
5	MESSAGE	Unable to issue file equate for DBCHGF (CIERR <i>n</i> ) (DBC 5).
	MEANING	The MPE intrinsic COMMAND failed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where CIERR <i>n</i> is the error number returned from the COMMAND intrinsic.

---

6	MESSAGE	Cannot read the change file (FSERR <i>n</i> ) (DBC 6).
	MEANING	The MPE intrinsic FREADDIR failed while reading the change file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

---

7	MESSAGE	Cannot write to the change file (FSERR <i>n</i> ) (DBC 7).
	MEANING	The MPE intrinsic FWRITEDIR failed while writing to the change file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
8	MESSAGE	Cannot open the change file (FSERR <i>n</i> ) (DBC 8).
	MEANING	The MPE intrinsic FOPEN failed while opening the change file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
9	MESSAGE	Cannot open the root file (FSERR <i>n</i> ) (DBC 9).
	MEANING	The MPE intrinsic FOPEN failed while opening the database root file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
10	MESSAGE	Cannot close the change file (FSERR <i>n</i> ) (DBC 10).
	MEANING	The MPE intrinsic FCLOSE failed while closing the change file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		

11	MESSAGE	Cannot close the root file (FSERR <i>n</i> ) (DBC 11).
	MEANING	The MPE intrinsic FCLOSE failed while closing the database root file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
12	MESSAGE	Cannot read the root file (FSERR <i>n</i> ) (DBC 12).
	MEANING	The MPE intrinsic FREAD or FREADDIR failed while reading the database root file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
13	MESSAGE	Cannot open the database (IMAGERR <i>n</i> ) (FSERR <i>n</i> ) (DBC 13).
	MEANING	The TurboIMAGE/XL intrinsic DBOPEN failed while opening the database.
	ACTION	Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> , where IMAGERR <i>n</i> is the error number returned in the status array. Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic. An asterisk (*) means no file system error occurred.
<hr/>		

14	MESSAGE	Cannot close the database (IMAGERR <i>n</i> ) (FSERR <i>n</i> ) (DBC 14).
	MEANING	The TurboIMAGE/XL intrinsic DBCLOSE failed while closing the database.
	ACTION	Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> , where IMAGERR <i>n</i> is the error number returned in the status array. Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic. An asterisk (*) means no file system error occurred.
<hr/>		
15	MESSAGE	Internal error, change file not open (DBC 15).
	MEANING	The change file was closed unexpectedly.
	ACTION	Notify the system manager.
<hr/>		
16	MESSAGE	DBINFO intrinsic failed (IMAGERR <i>n</i> ) (FSERR <i>n</i> ) (DBC 16).
	MEANING	The TurboIMAGE/XL intrinsic DBINFO failed.
	ACTION	Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> , where IMAGERR <i>n</i> is the error number returned in the status array. Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic. An asterisk (*) means no file system error occurred.

17	MESSAGE	The change file is full (DBC 17).
	MEANING	The DBCPLUS program failed to find any empty records in the change file for the new change.
	ACTION	Run the DBAPLUS program to restructure the database with the current changes. Then run the DBCPLUS program to create a new change file for the remaining changes.

**Caution**




---

Review the changes in the change file before running the DBAPLUS program. Make sure that the changes will leave the database in a consistent state.

---



---

18	MESSAGE	FFILEINFO intrinsic failed (FSERR <i>n</i> ) (DBC 18).
	MEANING	The MPE intrinsic FFILEINFO failed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

19	MESSAGE	Cannot create temporary DBCHGF file (FSERR <i>n</i> ) (DBC 19).
	MEANING	The change file could not be built.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

20	MESSAGE	Cannot purge temporary DBCHGF file (FSERR <i>n</i> ) (DBC 20).
	MEANING	The change file could not be purged.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

21	MESSAGE	Invalid control record in change file (DBC 21).
	MEANING	The database name, or the group and account stored in an existing change file does not match the logon group and account or the database name.
	ACTION	Verify that you have logged on correctly and entered the correct database name.
<hr/>		
22	MESSAGE	Internal item information is inconsistent (DBC 22).
	MEANING	The change file contains an inconsistent or invalid item name or number.
	ACTION	Notify the system manager.
<hr/>		
23	MESSAGE	Internal set information is inconsistent (DBC 23).
	MEANING	The change file contains an inconsistent or invalid data set name or number.
	ACTION	Notify the system manager.
<hr/>		
24	MESSAGE	Internal error, cannot find set information record (DBC 24).
	MEANING	The set information record for the master data set associated with the sort item could not be found in the change file.
	ACTION	Notify the system manager.
<hr/>		
25	MESSAGE	Internal error, no search key defined for master (DBC 25).
	MEANING	No key item is defined for the master data set in the set information record in the change file.
	ACTION	Notify the system manager.
<hr/>		

26           MESSAGE           Insufficient stack space available  
                                  to initialize change file (DBC 26).  
  
                  MEANING           The stack could not be expanded enough  
                                  to hold the root file information. Set and  
                                  item information must be read from the  
                                  root file to initialize tables stored in the  
                                  change file.  
  
                  ACTION           Notify the system manager.

---

27           MESSAGE           Cannot initiate DBAPLUS (create  
                                  failed) (DBC 27).  
  
                  MEANING           The MPE intrinsic CREATE failed.  
  
                  ACTION           Notify the system manager.

---

28           MESSAGE           Cannot initiate DBAPLUS (activate  
                                  failed) (DBC 28).  
  
                  MEANING           The MPE intrinsic ACTIVATE failed.  
  
                  ACTION           Notify the system manager.

---

51           MESSAGE           Enter the item/set name or @ to  
                                  review all items/sets (DBC 51).  
  
                  MEANING           The item or set name cannot be blank.  
  
                  ACTION           To change a set or item, enter the  
                                  name. To review sets or items, use the  
                                  REVIEW SETS or REVIEW ITEMS  
                                  command.

---

52	MESSAGE	This is not a TurboIMAGE database (DBC 52).
	MEANING	The database is an IMAGE/3000 database. DBChange Plus can only be used on TurboIMAGE/XL databases.
	ACTION	Use DBCONV.PUB.SYS to convert the IMAGE/3000 database to a TurboIMAGE/XL database. Run DBChange Plus again.
<hr/>		
56	MESSAGE	The master data set is not in the database (DBC 56).
	MEANING	The master data set has not been defined for this database.
	ACTION	Enter the name of an existing master data set.
<hr/>		
57	MESSAGE	Enter either user class or password to delete a password (DBC 57).
	MEANING	Neither the user class nor the password were specified. You must enter either the user class or password or both when deleting a password.
	ACTION	Enter the user class, password, or both.
<hr/>		
58	MESSAGE	The detail data set is not in the database (DBC 58).
	MEANING	The detail data set has not been defined for this database.
	ACTION	Enter the name of an existing detail data set.
<hr/>		

59           MESSAGE           Enter new database name if copying  
                                  to logon group and account (DBC 59) .

                  MEANING           When making a copy of the database  
                                  that will reside in the logon group and  
                                  account, you must specify a different  
                                  name for the database copy.

                  ACTION           Specify a new name for the database  
                                  copy.

---

61           MESSAGE           Job stream file *FileName* cannot be  
                                  opened (FSERR *n*) (DBC 61) .

                  MEANING           Could not open the job stream file  
                                  *FileName* that would run the DBAPLUS  
                                  program in batch mode.

                  ACTION           Refer to the *MPE Intrinsic Reference  
                                  Manual*, where FSERR *n* is the file  
                                  system error number returned from the  
                                  FCHECK intrinsic.

---

62           MESSAGE           The user class does not match the  
                                  password entered (DBC 62) .

                  MEANING           The specified user class is not associated  
                                  with the specified password.

                  ACTION           Enter a user class that corresponds to  
                                  the password you want to delete.

---

63           MESSAGE           This user class already exists (DBC  
                                  63) .

                  MEANING           The specified user class number already  
                                  exists.

                  ACTION           Specify a new and unique user class  
                                  number that does not already exist.

---

64	MESSAGE	The user class does not exist (DBC 64) .
	MEANING	The specified user class does not exist.
	ACTION	Specify an existing user class.
65	MESSAGE	Database copy must be done independently of any other changes (DBC 65) .
	MEANING	When copying a database, you cannot specify any other DBChange Plus function, except HELP or EXIT. If you wish to copy the database to another name, you must enter the new name at the same time you enter the user, group, and account for the copy.
	ACTION	Run the DBAPLUS program to copy the database; then, after the copy is finished, enter any other changes you wants.
66	MESSAGE	Duplicate data item name (DBC 66) .
	MEANING	A data item cannot be renamed with the name of an existing data item in the database.
	ACTION	Specify a new data item name. To review existing data items, use the REVIEW ITEMS command.
67	MESSAGE	No more data items can be added; max. of 1023 has been reached (DBC 67) .
	MEANING	The TurboIMAGE/XL limit of 1023 data items per database has been reached.
	ACTION	To add this data item, you must first delete a data item from the database.

68	MESSAGE	No more data sets can be added; max. of 199 has been reached (DBC 68).
	MEANING	The TurboIMAGE/XL limit of 199 data sets per database has been reached.
	ACTION	To add this data set, you must first delete a data set from the database.
69	MESSAGE	Search item must be a non-compound item (DBC 69).
	MEANING	The subitem count of a search item cannot be changed because a search item cannot be a compound data item. A search item must have a subitem count of 1.
	ACTION	Make sure the item whose subitem count you are changing is not a search item.
70	MESSAGE	New item type cannot be converted from current item type (DBC 70).
	MEANING	When changing data item attributes, the new data item type must be able to convert from the existing data item type.
	ACTION	Refer to Table 4-2 for information on available type conversions between different item types.
71	MESSAGE	This is not a field in this data set (DBC 71).
	MEANING	The specified field is not an item in this data set.
	ACTION	Specify a valid field. To review the fields in the data set, use the REVIEW FIELDS command.

72	MESSAGE	No more fields can be added; max. of 255 has been reached (DBC 72).
	MEANING	The TurboIMAGE/XL limit of 255 items (fields) per data set has been reached.
	ACTION	To add this field, you must first delete a field from the data set.
<hr/>		
73	MESSAGE	Only one primary path can be defined (DBC 73).
	MEANING	More than one primary path for this data set was specified.
	ACTION	Specify only one primary path.
<hr/>		
74	MESSAGE	Duplicate search item in detail set (DBC 74).
	MEANING	The search item in a detail data set was defined as the key item for two or more master data sets.
	ACTION	Define one unique key item for each master data set.
<hr/>		
75	MESSAGE	A sort item must be type U, X or K (DBC 75).
	MEANING	An item type other than U, X, or K was specified for a sort item. Only U, X, or K type is valid for sort items.
	ACTION	Specify a sort item of type U, X, or K.
<hr/>		

76           MESSAGE       Run DBAPLUS from group.account  
                              where database will reside (DBC 76).

              MEANING       When copying a database, you must  
                              logon to the group and account where  
                              the copy will reside and run the  
                              DBAPLUS program from that group  
                              and account.

              ACTION        Logon to the group and account  
                              where the copy will reside and run the  
                              DBAPLUS program.

---

77           MESSAGE       This field already exists in this  
                              data set (DBC 77).

              MEANING       The item to be added already exists in  
                              this data set.

              ACTION        Specify a new data item name.

---

78           MESSAGE       This is not a master set (DBC 78).

              MEANING       A detail data set name was specified, but  
                              a master data set name is required.

              ACTION        Specify a master data set name.

---

79           MESSAGE       This is not a detail data set (DBC  
                              79).

              MEANING       A master data set name was specified,  
                              but a detail data set name is required.

              ACTION        Specify a detail data set name.

---

80           MESSAGE       Enter a Y to redefine primary path  
                              (DBC 80).

              MEANING       A character other than Y was entered.

              ACTION        Enter Y if you want to redefine the  
                              primary path.

---

81	MESSAGE	Job stream file <i>FileName</i> cannot be created (FSERR <i>n</i> ) (DBC 81).
	MEANING	Could not create the job stream file <i>FileName</i> that runs the DBAPLUS program in batch mode.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from FCHECK intrinsic.
<hr/>		
82	MESSAGE	This is not a data item in the database (DBC 82).
	MEANING	The specified data item does not exist in the database or change file.
	ACTION	Specify the name of an existing data item.
<hr/>		
83	MESSAGE	This is not a data set in the database (DBC 83).
	MEANING	The specified data set does not exist in the database or change file.
	ACTION	Specify the name of an existing data set.
<hr/>		
84	MESSAGE	This item already exists (DBC 84).
	MEANING	The data item to be added already exists in the database or change file.
	ACTION	Specify a new data item name.
<hr/>		
85	MESSAGE	This set already exists (DBC 85).
	MEANING	The data set to be added already exists in the database or change file.
	ACTION	Specify a new data set name.
<hr/>		

86	MESSAGE	The device class is not configured on the system (DBC 86).
	MEANING	The device class entered for the data set is not configured on the system.
	ACTION	Enter the name of a configured device class that is valid for a disk file.
<hr/>		
87	MESSAGE	Duplicate data set name (DBC 87).
	MEANING	A data set cannot be renamed with the name of an existing data set.
	ACTION	Specify a new data set name. To review existing data sets, use the REVIEW SETS command.
<hr/>		
88	MESSAGE	This name cannot be blank (DBC 88).
	MEANING	A required name was not specified.
	ACTION	Specify a name.
<hr/>		
89	MESSAGE	You must logon as the database creator (DBC 89).
	MEANING	To perform any DBChange Plus functions, you must logon as the database creator.
	ACTION	Logon as the database creator.
<hr/>		
90	MESSAGE	The search item of a manual master cannot be changed (DBC 90).
	MEANING	Changing the key item of a manual master data set is not supported.
	ACTION	To add or delete paths to a manual master data set, change the associated detail data set path.
<hr/>		

91	MESSAGE	This is a deleted data item (DBC 91).
	MEANING	The specified data item has been previously deleted.
	ACTION	Specify the name of an existing data item.
92	MESSAGE	This is a deleted data set (DBC 92).
	MEANING	The specified data set has been previously deleted.
	ACTION	Specify the name of an existing data set.
93	MESSAGE	Write to job stream file <i>FileName</i> failed (FSERR <i>n</i> ) (DBC 93).
	MEANING	Could not write to the job stream file <i>FileName</i> that runs the DBAPLUS program in batch mode.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.
94	MESSAGE	An automatic master search item cannot be changed (DBC 94).
	MEANING	The key item in an automatic master data set cannot be explicitly changed.
	ACTION	To change an automatic master data set key item, first delete the automatic master data set then re-add it with a different search item.

95           MESSAGE           Master data set cannot be moved  
                                  after its related details (DBC 95).  
  
                  MEANING           When reordering data sets, a master  
                                  data set must be moved before any of its  
                                  associated detail data sets.  
  
                  ACTION           Reorder the master data set so that it is  
                                  defined before all of its associated detail  
                                  data sets.

---

96           MESSAGE           New capacity cannot be less than  
                                  number of existing entries (DBC 96).  
  
                  MEANING           The specified new capacity is less than  
                                  the number of existing entries in the  
                                  data set.  
  
                  ACTION           Specify a capacity that is equal to or  
                                  greater than the number of existing  
                                  entries.

---

97           MESSAGE           New name does not follow TurboIMAGE  
                                  naming conventions (DBC 97).  
  
                  MEANING           A data item or data set cannot be added  
                                  or renamed with an invalid name. A  
                                  data item or data set name must be  
                                  from 1 to 16 characters, and the first  
                                  character must be alphabetic.  
  
                  ACTION           Specify a valid data item or data set  
                                  name.

---

98           MESSAGE           A search item name is required for a  
                                  master set (DBC 98).  
  
                  MEANING           A new master data set cannot be added  
                                  without specifying the required key item.  
  
                  ACTION           Specify a key item when adding the  
                                  master data set.

---

99	MESSAGE	Detail data set cannot be moved before its related masters (DBC 99).
	MEANING	When reordering data sets, a detail data set must be moved after any of its associated master data sets.
	ACTION	Reorder the detail data set so that it is defined after all of its associated master data sets.
<hr/>		
100	MESSAGE	The search item must be a field in this data set (DBC 100).
	MEANING	The specified search item is not a field in this detail data set.
	ACTION	Add the search item to the field list in the detail data set.
<hr/>		
101	MESSAGE	No more paths can be added, max. of 16 has been reached (DBC 101).
	MEANING	The TurboIMAGE/XL limit of 16 paths per data set has been reached.
	ACTION	To add this path, you must first delete a path from the data set.

---

**DBChange Plus  
Messages  
(continued)**

102	MESSAGE	Detail search item (type) does not match master key (DBC 102).
	MEANING	A search item specified in the detail data set has a different data type than the key item in the related master data set.
	ACTION	Specify the same data type for the search and key items that define the path from the detail data set to the master data set.

103        MESSAGE        Detail search item (length) does not match master key (DBC 103).

           MEANING        A search item specified in the detail data set has a different length than the key item in the related master data set.

           ACTION         Specify the same length for the search and key items that define the path from the detail data set to the master data set.

---

104        MESSAGE        This is not a search item in the data set (DBC 104).

           MEANING        The specified search item is not a search item in the detail data set.

           ACTION         Specify the name of an existing search item.

---

105        MESSAGE        This is a search item, delete the path first (DBC 105).

           MEANING        A search item cannot be deleted from a data set.

           ACTION         Delete the path to the data set first, then delete the item.

---

107        MESSAGE        Sub-item length for item type I or J must be 1, 2, or 4 (DBC 107).

           MEANING        The specified length is invalid for item type I or J.

           ACTION         Specify a subitem length of 1, 2, or 4.

---

108        MESSAGE        Sub-item length for item type R must be 2 or 4 (DBC 108).

           MEANING        The specified length is invalid for item type R.

           ACTION         Specify a subitem length of either 2 or 4.

---

109	MESSAGE	Sub-item length for item type K must be 1 or 2 (DBC 109).
	MEANING	The specified length is invalid for item type K.
	ACTION	Specify a subitem length of 1 or 2.
110	MESSAGE	Product of sub-item count and length must be even (DBC 110).
	MEANING	When adding or changing a data item of type U or X, the product of the subitem count and length must be an even number.
	ACTION	Specify a subitem count and length for which the product is even.
111	MESSAGE	Data item length cannot exceed 2047 words (DBC 111).
	MEANING	The product of the subitem count and subitem length is greater than 2047 words.
	ACTION	Correct the length and count.
112	MESSAGE	Product of sub-item count and length must be divisible by 4 (DBC 112).
	MEANING	When adding or changing a data item type of P, the product of the subitem count and length must be divisible by 4.
	ACTION	Correct the length and count.
113	MESSAGE	No more items to display (DBC 113).
	MEANING	You have reached the beginning or end of the data item list.
	ACTION	No action necessary.

114        MESSAGE        The master set has to be defined  
                         before the detail (DBC 114) .  
  
                 MEANING        A detail data set was added with paths  
                         to a master data set that is defined later  
                         in the schema.  
  
                 ACTION        Place the new detail data set after  
                         all associated master data sets in the  
                         schema.

---

115        MESSAGE        No more sets to display (DBC 115) .  
  
                 MEANING        You have reached the beginning or end  
                         of the data set list.  
  
                 ACTION        No action necessary.

---

117        MESSAGE        No more fields to display (DBC 117) .  
  
                 MEANING        You have reached the beginning or end  
                         of the field list.  
  
                 ACTION        No action necessary.

---

119        MESSAGE        A value must be entered in this field  
                         (DBC 119) .  
  
                 MEANING        A required value is missing.  
  
                 ACTION        Specify the required value.

---

120        MESSAGE        No changes have been entered in the  
                         change file (DBC 120) .  
  
                 MEANING        The change file contains no changes for  
                         DBAPLUS to process.  
  
                 ACTION        Enter changes before invoking the  
                         DBAPLUS program.

---

---

122	MESSAGE	Only 'R', 'W', 'N', or blank is valid (DBC 122).
	MEANING	A character other than R, W, N, or blank was specified.
	ACTION	Enter R for read access, W for read and write access, N for no access, or leave blank.

---

---

123	MESSAGE	Cannot open the schema file (FSERR <i>n</i> ) (DBC 123).
	MEANING	The MPE intrinsic FOPEN failed while opening the schema file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

---

124	MESSAGE	Cannot write to the schema file (FSERR <i>n</i> ) (DBC 124).
	MEANING	The MPE intrinsic FWRITE failed while writing to the schema file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

---

125	MESSAGE	Check file <i>FileName</i> exists; it will be overwritten (DBA 125).
	MEANING	The file <i>FileName</i> used to hold the events from a prior CHECK will be purged.
	ACTION	If the file <i>FileName</i> is still needed, rename it.

---

---

125	MESSAGE	Duplicate password (DBC 125).
	MEANING	The specified password already exists for this database.
	ACTION	Specify a new and unique password.
<hr/>		
126	MESSAGE	Change file version <i>n</i> not match DBChange version <i>n</i> (DBC 126).
	MEANING	The change file was either created with an old version of DBChange or with a newer version than is currently executing. The change file may have been created by either DBChange/V or DBChange/CM, not by DBChange Plus.
	ACTION	Verify that the version of DBChange running on your system is the correct one. Then purge the change file and create a new one.
<hr/>		
127	MESSAGE	Database is opened exclusively by another user (DBC 127).
	MEANING	The database has been opened exclusively by another user, during which time no other user can access the database.
	ACTION	Use DBChange Plus at a later time.
<hr/>		
128	MESSAGE	Security violation (DBC 128).
	MEANING	The database cannot be opened due to MPE file system security.
	ACTION	Verify that you have access to the database.
<hr/>		

129	MESSAGE	The new name is the same as the old name (DBC 129).
	MEANING	The name of the database has not been changed.
	ACTION	Enter a different name when renaming the database.
<hr/>		
130	MESSAGE	This file already exists, please specify a new name (DBC 130).
	MEANING	A file already exists with the same database name specified for the database copy or rename.
	ACTION	Enter a different name.
<hr/>		
131	MESSAGE	Password contains invalid characters (DBC 131).
	MEANING	Invalid characters appear in the password.
	ACTION	Re-enter the password using only valid characters. Passwords must be from 1 to 8 characters. Either uppercase or lowercase characters are allowed. Passwords cannot contain blanks, carriage returns, slashes (/ or \), or semicolons (;).
<hr/>		
132	MESSAGE	The password does not exist (DBC 132).
	MEANING	The specified password does not exist in the database.
	ACTION	Enter an existing password.
<hr/>		

133	MESSAGE	Sort item is the same as the search item (DBC 133).
	MEANING	The sort item specified is the same data item as the search item that defines the path.
	ACTION	Enter a sort item that is different from the search item.
<hr/>		
134	MESSAGE	Capacity/block factor should not exceed 8388607 (DBC 134).
	MEANING	The maximum size of the data set was exceeded. This is a TurboIMAGE/XL limitation.
	ACTION	Reduce the capacity or increase the blocking factor.
<hr/>		
140	MESSAGE	PASCAL 'Range' error detected (DBA 140).
	MEANING	While converting a data item to a new type, a value in the original data set was either too large or too small for the new type specified for the data item. The data was truncated.
	ACTION	The values in the original data set should be examined before converting a data item type to a smaller subitem length.
<hr/>		
200	MESSAGE	Set capacity must be between 1 and 2147483647 (DBC 200).
	MEANING	The capacity of the data set was not between 1 and $2^{31} - 1$ (2,147,483,647).
	ACTION	Enter a capacity within these limits. If the set is a master data set, the capacity should be a prime number.
<hr/>		

201	MESSAGE	Block factor must be between 0 and 255 (DBC 201).
	MEANING	The blocking factor of the data set was not between 0 and 255 inclusive. If the blocking factor is zero, DBSCHEMA calculates the most optimal blocking factor for the set and then reblocks it.
	ACTION	Enter a blocking factor within these limits.
<hr/>		
202	MESSAGE	Invalid device class name (DBC 202).
	MEANING	The device class name entered does not follow standard naming conventions for a device class. A logical device number is not a valid device class name.
	ACTION	Enter a valid device class name.
<hr/>		
203	MESSAGE	Sub-item count must be between 1 and 255 (DBC 203).
	MEANING	The subitem count of the data item was not between 1 and 255, inclusive.
	ACTION	Enter a subitem count within these limits.
<hr/>		
204	MESSAGE	Type must be I, J, K, P, R, U, X, or Z (DBC 204).
	MEANING	A data item type was entered that was not I, J, K, P, R, U, X, or Z.
	ACTION	Enter a valid data item type. Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> for a list of types accepted by TurboIMAGE/XL.
<hr/>		

205	MESSAGE	Sub-item length must be between 1 and 255 (DBC 205).
	MEANING	The subitem length was not between 1 and 255, inclusive.
	ACTION	Enter a subitem length within these limits.
<hr/>		
206	MESSAGE	Set type must be A, M, or D (DBC 206).
	MEANING	A data set type was entered that was not A for automatic master data set, M for manual master data set, or D for detail data set.
	ACTION	Enter a valid data set type.
<hr/>		
207	MESSAGE	User class must be between 1 and 63 (DBC 207).
	MEANING	A user class was entered that was not between 1 and 63, inclusive.
	ACTION	Enter a user class within these limits.
<hr/>		
220	MESSAGE	Database has been modified since last DBSTORE (DBA 220).
	MEANING	The DBAPLUS program was run on a database that has had modifications (i.e. adds, updates, deletes) since the last time it was stored with DBSTORE.
	ACTION	Make a new backup copy of the database using DBSTORE.

226	MESSAGE	Fatal error checking root file (Internal Error <i>n</i> ) (DBA 226).
	MEANING	Damage has occurred to the root file which prohibits file checks from completing.
	ACTION	Restore a backup copy of the root file making sure it matches the structure of the current database and recheck the root file.
<hr/>		
230	MESSAGE	Invalid change file structure (DBA 230).
	MEANING	The change file contains inconsistent data item and data set information. Given the item or set number, the corresponding item or set name could not be found in the change file.
	ACTION	Run the DBCPLUS program again, and create a new change file. If the database is corrupt, purge the database and restore it from a backup copy.
<hr/>		
240	MESSAGE	Change file version <i>n</i> not match DBAlter version <i>n</i> (DBA 240).
	MEANING	The change file was created on a different version of DBChange Plus than is currently being run.
	ACTION	Verify that the version of the DBAPLUS program (DBAlter) is the correct one for your system. Run the DBCPLUS program again, and create a new change file.

250	MESSAGE	Cannot close database <i>DatabaseName</i> (IMAGERR <i>n</i> ) (DBA 250).
	MEANING	The TurboIMAGE/XL intrinsic DBCLOSE failed while trying to close the database <i>DatabaseName</i> .
	ACTION	Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> , where IMAGERR <i>n</i> is the error number returned in the status array. If the error was caused by a file system error, message 410 is printed instead.
<hr/>		
252	MESSAGE	Cannot close the check file <i>FileName</i> (FSERR <i>n</i> ) (DBA 252).
	MEANING	The file <i>FileName</i> cannot be closed and saved as a permanent file.
	ACTION	Refer to the <i>MPE Ininsics Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
255	MESSAGE	Cannot close or purge the change file (DBA 255).
	MEANING	You do not have the file system write capability to close or purge the change file.
	ACTION	Modify the file system access capability where the change file resides. Then purge the change file by running the DBCPLUS program with the PURGEFC option in the BASE command.
<hr/>		

260	MESSAGE	Cannot close the root file (DBA 260).
	MEANING	The MPE intrinsic FCLOSE failed while closing the root file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
270	MESSAGE	Cannot close the data set <i>DataSetName</i> (DBA 270).
	MEANING	An error occurred while closing the data set <i>DataSetName</i> . The MPE intrinsic FCLOSE failed. In addition to 270, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
280	MESSAGE	Restructuring has not been completed due to fatal error (DBA 280).
	MEANING	A fatal error occurred during a critical stage of restructuring the database.
	ACTION	If the database was modified, restore a backup copy of the database from the DBSTORE tape and try to restructure it again. If the error occurs again, notify the system manager.
<hr/>		
290	MESSAGE	Copying has not been completed due to a fatal error (DBA 290).
	MEANING	A fatal error occurred while copying the database.
	ACTION	Try to copy the database again. If the error occurs again, notify the system manager.
<hr/>		

310        MESSAGE        Cannot create the new root file (DBA 310).

           MEANING        The new database schema contains errors that caused DBSCHEMA to fail.

           ACTION         Check the new schema for errors. Run the DBCPLUS program again with the same change file and correct the errors.

---

320        MESSAGE        Cannot create the data set *DataSetName* (DBA 320).

           MEANING        One of the following MPE intrinsics failed while creating the data set *DataSetName*: FOPEN, FCLOSE, FWRITE, FWRITEDIR, FWRITELABEL. In addition to 320, message 410 is also displayed.

           ACTION         Refer to the *MPE Intrinsics Reference Manual* where *n* is the file system error number returned from the FCHECK intrinsic.

---

330        MESSAGE        Cannot create the hashing database TEMPDB (DBA 330).

           MEANING        The MPE intrinsic FOPEN failed while creating the temporary database TEMPDB used in the conversion process. In addition to 330, message 410 is also displayed.

           ACTION         Refer to the *MPE Intrinsics Reference Manual* where *n* is the file system error number returned from the FCHECK intrinsic.

---

---

340	MESSAGE	Database has been corrupted (DBA 340).
	MEANING	The database is in an inconsistent state due to a fatal error encountered during the restructuring process.
	ACTION	Restore a backup copy of the database from the DBSTORE tape, and try the changes again. If the error occurs again, notify the system manager.

---

---

345	MESSAGE	Set <i>DataSetName</i> using block factor <i>n</i> exceeds blockmax <i>m</i> (DBA 345).
	MEANING	The blocking factor <i>n</i> for data set <i>DataSetName</i> is more than the specified maximum block size (blockmax) of <i>m</i> .
	ACTION	Change either the block factor or the maximum block size.

---

---

350	MESSAGE	Set <i>DataSetName</i> block size <i>n</i> exceeds 2560 words (DBA 350).
	MEANING	The block size <i>n</i> of the data set <i>DataSetName</i> exceeds 2560 words.
	ACTION	Run the DBCPLUS program again and change the blocking factor.

---

---

355	MESSAGE	Set <i>DataSetName</i> entry length <i>n</i> exceeds 2047 words (DBA 355).
	MEANING	The entry length of the data set <i>DataSetName</i> exceeds 2047 words.
	ACTION	Delete some fields from the data set.

---

---

360	MESSAGE	Cannot enable library traps (DBA 360).
	MEANING	The MPE intrinsic XLIBTRAP failed.
	ACTION	Notify the system manager.

---

365	MESSAGE	CHECK/FIX event file damaged at record <i>n</i> (DBA 365).
	MEANING	The file <i>DatabaseNameRX</i> does not meet DBChange Plus internal verification requirements.
	ACTION	Store the database and contact your HP representative.

---

366	MESSAGE	CHECK/FIX event file misordered at record <i>n</i> (DBA 366).
	MEANING	The file <i>DatabaseNameRX</i> does not meet DBChange Plus internal verification requirements.
	ACTION	Store the database and contact your HP representative.

---

370	MESSAGE	Root condition is <i>value</i> (DBA 370).
	MEANING	Each <i>value</i> of the condition word is listed below with a meaning and action for each.

---

		JB
	MEANING	The database has not been created; only the root file exists.
	ACTION	Run DBUTIL to finish creating the database.

---

		RM
	MEANING	A system failure occurred while output deferred was enabled.
	ACTION	Follow the recovery procedures outlined in the <i>TurboIMAGE/XL DBMS Reference Manual</i> .

---

---

**DBChange Plus  
Messages  
(continued)**

---

370	MESSAGE	Root condition is <i>value</i> (DBA 370).
	MEANING	Each <i>value</i> of the condition word is listed below with a meaning and action for each.
		<hr/>
		ME
	MEANING	DBUTIL was interrupted while creating the database.
	ACTION	Purge the database and re-create it.
		<hr/>
		MC
	MEANING	DBUTIL was interrupted while erasing the database.
	ACTION	Purge the database and restore a backup copy.
		<hr/>
		IL
	MEANING	Intrinsic Level Recovery (ILR) was interrupted and did not complete.
	ACTION	Run QUERY and open the database. DBOPEN will invoke ILR.
		<hr/>
		IE
	MEANING	DBUTIL was interrupted while the database was being enabled for Intrinsic Level Recovery (ILR).
	ACTION	Run DBUTIL to finish enabling the database for ILR.

---

ID

MEANING DBUTIL was interrupted while the database was being disabled for Intrinsic Level Recovery (ILR).

ACTION Run DBUTIL to finish disabling the database for ILR.

---

CN

MEANING The database is being converted from an IMAGE database to a TurboIMAGE/XL database. The database conversion process *cannot be continued*.

ACTION Purge the database and restore a backup copy. Run DBCONV again.

---

CA

MEANING The database is being converted from an IMAGE database to a TurboIMAGE/XL database. The database conversion process *can be continued*.

ACTION Run DBCONV to finish converting the database to TurboIMAGE/XL.

---

MV

MEANING DBUTIL was interrupted while a data set was being moved to a different logical device.

ACTION Purge the database and restore a backup copy.

---

380	MESSAGE	DBCHGF file equate can only contain actual file designator (DBA 380).
	MEANING	The file equation for DBCHGF contains options which are not allowed. Only the actual file name can be specified.
	ACTION	Reenter the file equation and run the DBAPLUS program again.
<hr/>		
390	MESSAGE	FFILEINFO failed on data set <i>DataSetName</i> (DBA 390).
	MEANING	The MPE intrinsic FFILEINFO failed when issued against the data set <i>DataSetName</i> . In addition to 390, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
400	MESSAGE	Internal error: file <i>FileName</i> is not open (DBA 400).
	MEANING	The DBAPLUS program expected the file <i>FileName</i> to be open and it is not.
	ACTION	Notify the system manager.
<hr/>		
410	MESSAGE	File system error <i>n</i> (DBA 410).
	MEANING	An MPE file system intrinsic did not complete.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		

413	MESSAGE	Damage requires manual intervention (DBA 413).
	MEANING	Database damage exists that cannot be repaired by DBChange Plus. In addition to 413, one of the following messages is also displayed: <ul style="list-style-type: none"> <li>■ Capacity and blocking factor recorded in root file are inconsistent.</li> <li>■ Data set media length recorded in the root file is inconsistent.</li> <li>■ Root file inconsistency detected by DBChange (CHK 220).</li> </ul>
	ACTION	Depending on the message displayed, take the following action: <ul style="list-style-type: none"> <li>■ If root file damage exists, restore a valid copy of the root file from the last database backup, then issue the CHECK command. Note that a restructure must <i>not</i> have occurred since the last backup.</li> <li>■ If MPE file characteristics do not match root file specifications, the root file and the affected data sets can be resynchronized. Before resynchronizing, be sure to store the database. After resynchronizing, issue the CHECK BASE command, then review the data in the database.</li> </ul>

---

415	MESSAGE	Cannot allocate <i>n</i> bytes of heap space (DBA 415).
	MEANING	Native mode heap is not large enough to accommodate DBChange Plus space requirements.
	ACTION	Copy the terminal screen and notify the system manager.

---

416	MESSAGE	Cannot return <i>n</i> bytes of heap space (DBA 416).
	MEANING	An internal error occurred while deallocating memory.
	ACTION	Copy the terminal screen and notify the system manager.
<hr/>		
420	MESSAGE	TurboIMAGE error <i>n</i> (DBA 420).
	MEANING	The TurboIMAGE/XL intrinsic DBCONTROL or DBPUT failed.
	ACTION	Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> , where <i>n</i> is the error number returned in the status array.
<hr/>		
430	MESSAGE	Invalid data item number found in the change file (DBA 430).
	MEANING	An invalid data item number was found in the change file. The change file is corrupt.
	ACTION	Run the DBCPLUS program again, and create a new change file. If the database is corrupt, purge it and restore a backup copy.
<hr/>		
440	MESSAGE	Invalid data set number found in change file (DBA 440).
	MEANING	An invalid data set number was found in the change file. The change file is corrupt.
	ACTION	Run the DBCPLUS program again, and create a new change file. If the database is corrupt, purge it and restore a backup copy.
<hr/>		

450        MESSAGE        Data item *ItemNumber* does not exist  
                              in the change file (DBA 450).

                 MEANING        The change file contains inconsistent  
                              data item information. The data item  
                              number *ItemNumber* was not found in  
                              the change file.

                 ACTION        Run the DBCPLUS program again, and  
                              create a new change file. If the database  
                              is corrupt, purge it and restore a backup  
                              copy.

---

460        MESSAGE        Cannot get extra data segment needed  
                              for copy/restructure (DBA 460).

                 MEANING        An extra data segment is used when  
                              copying a database and for certain  
                              restructuring functions. The system  
                              could not allocate one for the DBAPLUS  
                              program.

                 ACTION        Notify the system manager.

---

480        MESSAGE        Cannot obtain buffers for database  
                              *DatabaseName* (DBA 480).

                 MEANING        The minimum number of buffers  
                              necessary to perform the restructure  
                              could not be obtained.

                 ACTION        Refer to message 420.

---

490        MESSAGE        Cannot enable output deferred for  
                              database *DatabaseName* (DBA 490).

                 MEANING        The TurboIMAGE/XL intrinsic  
                              DBCNTROL failed to enable output  
                              deferred for the database *DatabaseName*.

                 ACTION        Refer to the *TurboIMAGE/XL DBMS*  
                              *Reference Manual*, where *n* is the error  
                              number returned in the status array.

---

499	MESSAGE	Native Language Support subsystem error <i>n</i> (DBA 499) .
	MEANING	Native Language Support is not functioning.
	ACTION	Notify the system manager.
<hr/>		
500	MESSAGE	Cannot create schema file <i>FileName</i> (DBA 500) .
	MEANING	The MPE intrinsic FOPEN or FCLOSE failed while opening or closing the schema file. In addition to 500, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
500	MESSAGE	Cannot open input file DBCIN (FSERR <i>n</i> ) (DBC 500) .
	MEANING	The MPE intrinsic FOPEN or HPFEQINFO failed while opening the input file or the DBCIN equate file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic. Verify that the input file exists, and that the file name is correct. Rerun the DBCPLUS program.
<hr/>		

501	MESSAGE	Detail set <i>DataSetName</i> still has <i>n</i> entries (DBA 501).
	MEANING	The ERASE SET command was issued for a manual master data set that has one or more paths linked to detail data sets. Related detail data set entries still exist.
	ACTION	Manual master data set entries cannot be erased until all related detail data set entries are erased. Issue the ERASE SET command for each related detail data set before issuing the ERASE SET command for the manual master data set.

---

501	MESSAGE	Cannot open output file DBCOUT (FSERR <i>n</i> ) (DBC 501).
	MEANING	The MPE intrinsic FOPEN or HPFEQINFO failed while opening the output file or the DBCOUT equate file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic. Verify that the output file exists, and that the file name is correct. Rerun the DBCPLUS program.

---

---

502	MESSAGE	Cannot erase manual set <i>DataSetName</i> due to detail entries (DBA 502).
	MEANING	The ERASE SET command was issued for a manual master data set that has one or more paths linked to detail data sets. Related detail data set entries still exist.
	ACTION	Manual master data set entries cannot be erased until all related detail data set entries are erased. Issue the ERASE SET command for each related detail data set before issuing the ERASE SET command for the manual master data set.

---

---

502	MESSAGE	Cannot write to the command output file (FSERR <i>n</i> ) (DBC 502).
	MEANING	The MPE intrinsic FWRITE failed while writing to the output file or the DBCOUT file equate.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR # is the file system error number returned from the FCHECK intrinsic. Verify that the output file, as specified, exists, and that the filename is correct. Rerun the DBCPLUS program.

---

---

503	MESSAGE	Cannot allocate internal buffers for processing master set (DBA 503).
	MEANING	The DBAPLUS program allocates up to four internal buffers to process erased entries on a master data set. This message is displayed if four buffers are not enough to process the erased entries.
	ACTION	Notify your Hewlett-Packard representative.

---

---

503	MESSAGE	LP or TERM must be used for OUTPUT command (DBC 503).
	MEANING	A parameter other than LP or TERM was specified for the OUTPUT command.
	ACTION	Enter either OUTPUT LP or OUTPUT TERM.

---

---

504	MESSAGE	Path search item is invalid, cannot REPACK SET <i>DataSetName</i> (DBA 504).
	MEANING	An incorrect search item name was specified in the REPACK SET command. The path may have been deleted after the REPACK SET command was used.
	ACTION	Specify the correct path to be repacked.

---

---

504	MESSAGE	Read input error (DBC 504).
	MEANING	An error occurred while reading the input from the terminal, the DBCIN file equate, or the XEQ file.
	ACTION	Verify that the input is correctly entered, and that the terminal, DBCIN file, or XEQ file is in ASCII format.

---

---

505	MESSAGE	Cannot use a newly added path to REPACK SET <i>DataSetName</i> (DBA 505).
	MEANING	A new path was added to the data set and then specified as the path for the REPACK SET command. No chain information exists for the new path, so the data set cannot be repacked.
	ACTION	Specify the correct path to be repacked with the REPACK SET command.

---

505	MESSAGE	Command exceeds 3 lines or 216 characters (DBC 505).
	MEANING	Too many lines or characters were used when entering a command.
	ACTION	Reenter the command using fewer than 3 lines or 216 characters.
<hr/>		
506	MESSAGE	Cannot HPFOPEN internal map file for set <i>DataSetName</i> (HPFERR <i>n</i> ) (DBA 506).
	MEANING	The MPE intrinsic HPFOPEN failed while opening an internal file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where HPFERR <i>n</i> is the error number returned from the HPFOPEN intrinsic.
<hr/>		
506	MESSAGE	EXIT command not entered. End of file reached. EXIT assumed (DBC 506).
	MEANING	No EXIT command was entered in a batch job or within an XEQ file.
	ACTION	If EXIT is desired, no action necessary; otherwise, enter the desired DBCPLUS commands, and run the DBCPLUS program again.
<hr/>		
507	MESSAGE	Message catalog read error (DBC 507).
	MEANING	An error occurred while reading a command from the DBChange Plus message catalog DBCMSGC.
	ACTION	Verify that the correct DMCMSGC message catalog is installed.
<hr/>		

508	MESSAGE	Invalid command (DBC 508).
	MEANING	An incorrect command was entered.
	ACTION	Enter a valid command or use the correct command abbreviation. Use HELP to find the correct command.
<hr/>		
509	MESSAGE	Command valid only in session mode (DBC 509).
	MEANING	A command that is valid only during a session (interactive) mode was entered in a job stream file.
	ACTION	Delete this command from the job stream file.
<hr/>		
510	MESSAGE	Cannot open database <i>DatabaseName</i> (IMAGERR <i>n</i> ) (DBA 510).
	MEANING	The TurboIMAGE/XL intrinsic DBOPEN failed while trying to open the database <i>DatabaseName</i> .
	ACTION	Refer to the <i>TurboIMAGE/XL DBMS Reference Manual</i> , where IMAGERR <i>n</i> is the error number returned in the status array. If the error was caused by a file system error, message 410 is printed instead.
<hr/>		
510	MESSAGE	Incorrect command parameter or missing parameter (DBC 510).
	MEANING	A parameter was entered incorrectly for the command, or a required parameter is missing.
	ACTION	Check the syntax of the command entered, and reenter the command using the correct parameter. Use HELP to verify the command syntax.
<hr/>		

511	MESSAGE	Correct BASE command is required before continuing (DBC 511).
	MEANING	The BASE command was not entered before processing the database. The only commands that can be issued before the BASE command are HELP, OUTPUT, REDO, and XEQ. The BASE command is required after issuing the PERFORM COMMANDS command.
	ACTION	Enter the appropriate BASE command.
<hr/>		
512	MESSAGE	Fatal error occurred. DBChange cannot continue (DBC 512).
	MEANING	This message prints immediately after message 506 or fatal error messages.
	ACTION	Correct the error(s) indicated by the messages.
<hr/>		
513	MESSAGE	Old change file does not exist (DBC 513).
	MEANING	In the BASE command, either OLD was entered, or it tried to default to an existing old change file. No old change file exists for the database.
	ACTION	Correct the BASE command by specifying NEW, then rerun the DBCPLUS program.

514           MESSAGE           Old change file cannot be renamed to be saved (FSERR *n*) (DBC 514).

                  MEANING           The MPE intrinsic FCLOSE failed while trying to save the old change file with a name other than *FileNameCO* or *FileNameCF*.

                  ACTION           Refer to the *MPE Intrinsic Reference Manual*, where FSERR *n* is the file system error number returned from the FCHECK intrinsic. Verify that there is sufficient disk space to save the change file and that no other CO or CF change file exists. Rerun the DBCPLUS program.

---

515           MESSAGE           PERFORM COMMANDS parmvalue required in batch and jobfile (DBC 515).

                  MEANING           A value was not entered in the batch job or in session mode when the JOB parameter was specified.

                  ACTION           Use a parameter value between 1 and 15 for the PERFORM COMMANDS command.

---

516           MESSAGE           Parm value must be between 1 and 15 (DBC 516).

                  MEANING           An incorrect value was entered for the PERFORM COMMANDS command.

                  ACTION           Enter a parameter value between 1 and 15 for the PERFORM COMMANDS command.

---

517	MESSAGE	Cannot PERFORM COMMANDS due to errors (DBC 517).
	MEANING	Errors were detected in the DBCPLUS run for the database. The PERFORM COMMANDS command cannot be executed.
	ACTION	Correct the errors, then run the PERFORM COMMANDS command.
<hr/>		
518	MESSAGE	Use CHANGE PRIMARYPATH or DELETE PATH then ADD PATH (DBC 518).
	MEANING	CHANGE PATH command was entered. No CHANGE PATH command exists.
	ACTION	Use other suggested commands.
<hr/>		
520	MESSAGE	Cannot open the change file <i>FileName</i> (DBA 520).
	MEANING	The MPE intrinsic FOPEN failed when opening the change file. In addition to 520, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
520	MESSAGE	Use CHANGE BLOCKFACTOR, CHANGE BLOCKMAX, CHANGE CAPACITY, CHANGE DEVICECLASS, CHANGE SETSECURITY or REORDER SET (DBC 520).
	MEANING	An invalid command was entered. No CHANGE SET command exists.
	ACTION	Use other suggested commands.
<hr/>		

521	MESSAGE	No recover command for field, password, path or sort (DBC 521).
	MEANING	No RECOVER command exists for the command keywords FIELD, PASSWORD, PATH, and SORT.
	ACTION	Use the ADD command to re-add fields, passwords, paths, or sort items after they have been deleted.
<hr/>		
522	MESSAGE	Invalid XEQ file name (DBC 522).
	MEANING	An incorrect XEQ file name was entered.
	ACTION	Enter the correct XEQ file name.
<hr/>		
523	MESSAGE	Cannot open the XEQ file (FSERR <i>n</i> ) (DBC 523).
	MEANING	The MPE intrinsic FOPEN failed while opening the XEQ file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic. Check that the XEQ file name is correct and that the file exists in the logon group and account.
<hr/>		
524	MESSAGE	XEQ file must be in ASCII format (DBC 524).
	MEANING	The XEQ file is not in ASCII format.
	ACTION	Enter a correct ASCII file for the XEQ file.
<hr/>		

525	MESSAGE	File system error <i>n</i> opening file <i>FileName</i> (DBA 525).
	MEANING	The MPE intrinsic FOPEN failed while opening the file <i>FileName</i> .
	ACTION	Refer to the <i>MPE Ininsics Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
525	MESSAGE	Another XEQ file is already opened. Nested XEQ file not allowed (DBC 525).
	MEANING	A second XEQ command was issued. Only one XEQ file can be processed at a time.
	ACTION	Remove the second XEQ command.
<hr/>		
526	MESSAGE	CONTROL Y internal error (DBC 526).
	MEANING	An error occurred while issuing CONTROL Y in the DBCPLUS program.
	ACTION	Check your terminal configuration or that of the MPE operating system.
<hr/>		
529	MESSAGE	Security access code must be 'R', 'W' or 'N' (DBC 529).
	MEANING	The access code entered is not R, W, or N.
	ACTION	Enter R for read, W for write, or N for no access.
<hr/>		

---

530	MESSAGE	Cannot open the root file (DBA 530).
	MEANING	The MPE intrinsic FOPEN failed while opening the root file. In addition of 530, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

---

530	MESSAGE	Blockmax must be between 128 and 2048 (DBC 530).
	MEANING	The maximum block size entered is not between 128 and 2048.
	ACTION	Enter a maximum block size between 128 and 2048.

---

---

531	MESSAGE	Set range must be in ascending schema order (DBC 531).
	MEANING	The set range entered for CHANGE BLOCKMAX is not in ascending schema order.
	ACTION	Verify the order of the data sets by using the REVIEW SET command. Specify the correct ascending data set range in the CHANGE BLOCKMAX command.

---

---

532	MESSAGE	For set <i>DataSetName</i> using block factor <i>n</i> exceeds blockmax <i>m</i> (DBC 532).
	MEANING	The blocking factor <i>n</i> exceeds the specified blockmax <i>m</i> for the data set <i>DataSetName</i> .
	ACTION	Either decrease the blocking factor or increase the maximum block size for the data set.

---

---

533	MESSAGE	For set <i>DataSetName</i> the block size <i>n</i> exceeds 2560 words (DBC 533).
	MEANING	The blocking factor for the data set <i>DataSetName</i> is too large causing the block size to exceed 2560 words.
	ACTION	Decrease the block factor for the data set.
<hr/>		
534	MESSAGE	For set <i>DataSetName</i> entry length <i>n</i> exceeds 2047 words. (DBC 534).
	MEANING	The entry length for the data set <i>DataSetName</i> contains too many items or items that are too long for the set causing the entry length to exceed 2047 words.
	ACTION	Check the items in the data set using the REVIEW ITEMS command. Either delete some items from the data set or decrease the length of some data items.
<hr/>		
535	MESSAGE	Cannot open the root file for exclusive access (DBA 535).
	MEANING	The MPE intrinsic FOPEN failed. In addition to 535, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
535	MESSAGE	Item name is required for this command (DBC 535).
	MEANING	The data item name is required but was not entered.
	ACTION	Use HELP to check the command syntax, and enter the data item name.
<hr/>		

536	MESSAGE	Set name is required for this command (DBC 536).
	MEANING	The data set name is required but was not entered.
	ACTION	Use HELP to check the command syntax, and enter the data set name.
<hr/>		
537	MESSAGE	A value must be entered for a required parameter (DBC 537).
	MEANING	A required value is missing from the command.
	ACTION	Use HELP to check the command syntax, and enter the correct command.
<hr/>		
538	MESSAGE	Error occurred when issuing the MPE command (CIERR <i>n</i> ) (DBC 538).
	MEANING	An invalid MPE command was entered after the colon (:).
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where CIERR <i>n</i> is the error number returned from the HPCICOMMAND intrinsic. Correct the problem and enter a valid command.
<hr/>		
539	MESSAGE	DBCIN file equate error in HPFEQINFO (HPQERR <i>n</i> ) (DBC 539).
	MEANING	The DBCIN file specified in the file equation is not a valid existing permanent file.
	ACTION	Enter a correct command file for the DBCIN equation.
<hr/>		

540	MESSAGE	Cannot open the data set <i>DataSetName</i> (DBA 540).
	MEANING	The MPE intrinsic FOPEN failed to open the data set. In addition to 540, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
540	MESSAGE	DBCOUT file equate error in HPFEQINFO (HPQERR <i>n</i> ) (DBC 540).
	MEANING	The DBCOUT file specified in the file equation is not \$STDLIST, a temporary disk file, a permanent disk file, or a print file.
	ACTION	Enter a correct DBCOUT file equation.
<hr/>		
541	MESSAGE	Password cannot exceed 8 characters (DBC 541).
	MEANING	More than eight characters were entered for a password.
	ACTION	Enter up to eight characters for a password.
<hr/>		
542	MESSAGE	MaxFull or NewFull percent must be 1 to 100 (DBC 542).
	MEANING	An incorrect percent number was entered for the <i>MaxFull</i> or <i>NewFull</i> parameter in the CONTROL command.
	ACTION	Enter a number between 1 and 100 inclusive for <i>MaxFull</i> or <i>NewFull</i> percent.
<hr/>		

---

544	MESSAGE	Cannot open data set file <i>FileName</i> (FSERR <i>n</i> ) (DBC544).
	MEANING	The MPE intrinsic FOPEN failed while opening the data set file <i>FileName</i> .
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

---

545	MESSAGE	File system error 46: out of disc space (DBA 545).
	MEANING	Not enough disk space is available to create a temporary file, most likely a data set file, needed for the restructure.
	ACTION	Obtain more disk space, and rerun the DBAPLUS program.

---

---

545	MESSAGE	Cannot read file label of data set file <i>FileName</i> (FSERR <i>n</i> ) (DBC 545).
	MEANING	The MPE intrinsic FREADLABEL failed while attempting to read the file label for the data set file <i>FileName</i> .
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

---

546	MESSAGE	Cannot decrease capacity. Use REPACK SET (DBC 546).
	MEANING	The CHANGE CAPACITY command was used to decrease the capacity of a detail data set whose delete chain count is greater than zero. The requested new capacity is below the high-water mark.
	ACTION	Use the REPACK SET command instead of the CHANGE CAPACITY command.

---

547	MESSAGE	No fields defined for data set <i>DataSetName</i> (DBC 547).
	MEANING	A data set was added without any fields added, or all fields in the data set were deleted.
	ACTION	Add at least one field to the data set or delete the data set.

---

548	MESSAGE	No paths defined for auto master set <i>DataSetName</i> (DBC 548).
	MEANING	An automatic master data set was added without any paths defined.
	ACTION	Add at least one path for the automatic master data set, or delete the data set.

---



---

**DBChange Plus  
Messages  
(continued)**

549	MESSAGE	No key item defined for master set <i>DataSetName</i> (DBC 549).
	MEANING	A master data set was added without a key item, or the key item was deleted.
	ACTION	Delete the data set; or delete the data set, and then add the set with the key item defined.

---

550	MESSAGE	Pascal error <i>n</i> (DBA 550).
	MEANING	A Pascal library procedure did not complete.
	ACTION	Refer to the <i>Pascal/3000 Reference Manual</i> , where <i>n</i> is the Pascal error number.

---

550	MESSAGE	Set <i>DataSetName</i> entry length <i>n</i> less than 2 words (DBC 550).
	MEANING	The detail data set <i>DataSetName</i> has no paths and an entry length less than two words (MPE/V words or MPE/XL halfwords), which is not allowed by DBSCHEMA.
	ACTION	Add more fields to the data set, or increase the item length of the existing fields.

---

551	MESSAGE	Cannot CHECK/FIX this new set until added by DBAlter (DBC 551).
	MEANING	The CHECK or FIX command cannot be issued for a data set that is not actually in the database, that is, not yet processed by the DBAPLUS program (DBAlter).
	ACTION	Run the DBAPLUS program to actually add the data set, then use the CHECK or FIX command.

---

552	MESSAGE	Invalid search item specified for CHECK/FIX PATH (DBC 552).
	MEANING	The item name entered as a search item for the CHECK or FIX command is not a valid search item.
	ACTION	Enter a correct search item name.

---

553	MESSAGE	Cannot erase automatic master data set (DBC 553).
	MEANING	The ERASE SET command was issued for an automatic master data set. The data in an automatic master data set cannot be explicitly erased.
	ACTION	An automatic master data set entry is erased once the detail data set entries are erased. Use the ERASE SET command to erase the related detail data sets linked to the automatic master data set.
<hr/>		
554	MESSAGE	Detail set <i>DataSetName</i> still has <i>n</i> entries (DBC 554).
	MEANING	The ERASE SET command was issued for a manual master data set that has one or more paths linked to detail data sets where related detail data set entries still exist.
	ACTION	Issue ERASE SET for each of the detail data set entries with paths linked to the manual master data set, then issue ERASE SET for the manual master data set.
<hr/>		
555	MESSAGE	Cannot read the change file (DBA 555).
	MEANING	The MPE intrinsic FREAD or FREADDIR failed while reading the change file. In addition to 555, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Ininsics Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		

---

555	MESSAGE	Cannot erase manual set with existing related detail entries (DBC 555).
	MEANING	The ERASE SET command was issued for a manual master data set that has one or more paths linked to detail data sets where related detail data entries still exist.
	ACTION	Issue ERASE SET for each of the detail data set entries with paths linked to the manual master data set, then issue ERASE SET for the manual master data set.

---

---

556	MESSAGE	Cannot erase data set until added by DBAlter (DBC 556).
	MEANING	ERASE SET was issued for a data set that was not yet actually added to the database, that is, processed by the DBAPLUS program (DBAlter).
	ACTION	Verify that the data set name is correct and exists in the database.

---

---

557	MESSAGE	Cannot read the check file <i>FileName</i> (FSERR <i>n</i> ) (DBA 557).
	MEANING	The MPE intrinsic FREAD or FREADDIR failed while reading the check file.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where FSERR <i>n</i> is the file system error number returned from the FCHECK intrinsic.

---

557	MESSAGE	ERASE SET has already been issued for the data set (DBC 557).
	MEANING	ERASE SET was issued a second time for the same data set.
	ACTION	Erase the data set entries and reissue the ERASE SET command.
<hr/>		
558	MESSAGE	Data set has not been set to erase. Cannot CANCEL ERASE (DBC 558).
	MEANING	CANCEL ERASE was issued for a data set that was not flagged to be erased by ERASE SET.
	ACTION	No action necessary if you do not intend to erase the data set entries.
<hr/>		
559	MESSAGE	CTL_ERASE_SETS less than zero. Cannot CANCEL ERASE (DBC 559).
	MEANING	CANCEL ERASE was issued for a data set. CTL_ERASE_SETS field in the change file control record, which reflects the number of data set entries to be erased, has a number less than zero.
	ACTION	Verify that the change file has been correctly restored from a previous DBCPLUS run. If not, purge the change file and rerun the DBCPLUS program.
<hr/>		
560	MESSAGE	Cannot read the root file (DBA 560).
	MEANING	The MPE intrinsic FREADDIR failed while trying to read the root file. In addition to 560, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		

560	MESSAGE	Data set has not been set to repack. Cannot CANCEL REPACK (DBC 560).
	MEANING	CANCEL REPACK was issued for a data set without issuing a previous REPACK SET command.
	ACTION	No action necessary if you do not want to repack the data set.
<hr/>		
561	MESSAGE	CTL_REPACK_SETS less than zero. Cannot CANCEL REPACK (DBC 561).
	MEANING	CANCEL REPACK was issued for a data set. The CTL_REPACK_SETS field in the change file control record, which reflects the number of data sets to be repacked, has a number less than zero.
	ACTION	Verify that the change file has been correctly restored from a previous DBCPLUS run. If not, purge the change file and rerun the DBCPLUS program.
<hr/>		
562	MESSAGE	Cannot repack a master data set (DBC 562).
	MEANING	REPACK SET was issued for a master data set.
	ACTION	REPACK SET can only be issued for a detail data set.
<hr/>		
563	MESSAGE	Cannot repack data set until added by DBAlter (DBC 563).
	MEANING	REPACK SET was issued for a data set that was not yet actually added to the database, that is, processed by the DBAPLUS program (DBAlter).
	ACTION	Verify that the data set name is correct and exists in the database.
<hr/>		

564	MESSAGE	REPACK SET has already been issued for the data set (DBC 564).
	MEANING	REPACK SET was issued a second time for the same data set.
	ACTION	If the repack option is to be changed for the data set, use CANCEL REPACK then REPACK SET with the correct repack option. Otherwise, no action is necessary.

---

565	MESSAGE	REPACK SET option must be serial or chained (DBC 565).
	MEANING	REPACK SET was issued with the incorrect repack option or the repack option was incorrectly placed in the command syntax.
	ACTION	Verify the syntax of the REPACK SET command by using HELP. Use serial or chained as the repack option.

---

566	MESSAGE	Invalid search item specified for REPACK SET (DBC 566).
	MEANING	An incorrect or misplaced search item was used in the REPACK SET command.
	ACTION	Verify that the correct search item name was used. Verify the syntax of the REPACK SET command by using HELP.

---

567	MESSAGE	Path search item is invalid, cannot REPACK SET <i>DataSetName</i> (DBC 567).
	MEANING	The path specified by the search item used in the REPACK SET command does not exist. The path may have been deleted after the REPACK SET command was used.
	ACTION	Verify that the path still exists by using the REVIEW PATHS command. If not, use the CANCEL REPACK and REPACK SET commands for another path.
<hr/>		
568	MESSAGE	Cannot use a newly added path to REPACK SET <i>DataSetName</i> (DBC 568).
	MEANING	A new path was added to the data set <i>DataSetName</i> , and the path is being specified for a REPACK SET command with the chained option. No chain information exists for the new path, so it cannot be used to repack the data set.
	ACTION	Use the CANCEL REPACK and REPACK SET commands for another path.
<hr/>		
569	MESSAGE	Duplicate schema file. Cannot process PRINT SCHEMA (DBC 569).
	MEANING	A file with the same name already exists.
	ACTION	Use another unique file name, or purge the existing file and run the DBCPLUS program again with the PRINT SCHEMA command.

570           MESSAGE        Cannot read the root file label (DBA  
570) .

              MEANING        The MPE intrinsic FREADLABEL  
failed while trying to read the file label  
of the root file. In addition to 570,  
message 410 is also displayed.

              ACTION         Refer to the *MPE Intrinsic Reference  
Manual*, where *n* is the file system error  
number returned from the FCHECK  
intrinsic.

---

570           MESSAGE        Cannot close the schema file (FSERR  
*n*) (DBC 570) .

              MEANING        The MPE intrinsic FCLOSE failed while  
closing the schema file.

              ACTION         Refer to the *MPE Intrinsic Reference  
Manual*, where FSERR *n* is the file  
system number returned from the  
FCHECK intrinsic.

---

571           MESSAGE        LOGON to the account where  
*NewBaseName* will reside (DBC 571) .

              MEANING        When copying a database, you must be  
logged on to the account where the new  
database copy will reside.

              ACTION         Logon to the correct account.

---

572           MESSAGE        SM or (OP and PM) capabilities  
required to copy across accounts  
(DBC 572) .

              MEANING        You do not have the required MPE  
capabilities to copy a database across  
accounts.

              ACTION         Acquire SM or (OP and PM)  
capabilities.

---

573           MESSAGE           AM capability or CREATOR access  
required to copy within account  
(DBC 573).

                  MEANING           You do not have the required MPE  
capabilities to copy a database within an  
account.

                  ACTION           Acquire AM capability or CREATOR  
access.

---

574           MESSAGE           SF (save file) capability required  
to copy a database (DBC 574).

                  MEANING           You do not have the required MPE  
capability to copy a database.

                  ACTION           Acquire SF capability.

---

575           MESSAGE           AM capability or CREATOR access  
required to replace a database  
(DBC 575).

                  MEANING           You do not the required MPE capability  
to replace a database.

                  ACTION           Acquire AM capability or CREATOR  
access.

---

576           MESSAGE           Percentfull must be  $0 < MinFull < NewFull < MaxFull < 100$  (DBC 576).

                  MEANING           The values entered for the CONTROL  
PERCENTFULL command are  
incorrect.

                  ACTION           Re-enter the CONTROL  
PERCENTFULL parameter values.  
*MinFull* must be 0 to 100; *NewFull* and  
*MaxFull* must be 1 to 100; *MaxFull* must  
be greater than *NewFull*; and *NewFull*  
must be greater than *MinFull*.

---

577           MESSAGE       Data set *DataSetName* has capacity  
less than one (DBC 577) .

              MEANING       The capacity for the data set  
*DataSetName* in the current database  
is less than one. CONTROL  
PERCENTFULL cannot be calculated  
for the set.

              ACTION        Change the capacity using the CHANGE  
CAPACITY command.

---

578           MESSAGE       Cannot calculate PERCENTFULL for a  
new data set (DBC 578) .

              MEANING       The data set specified in the CONTROL  
PERCENTFULL command is not yet  
added to the database.

              ACTION        Either specify a data set that already  
exists, or add the specified data set  
by running the DBAPLUS program,  
and then entering the CONTROL  
PERCENTFULL command again.

---

579           MESSAGE       MinFull percent must be 0 to 100 (DBC  
579) .

              MEANING       The *MinFull* parameter was  
incorrectly entered in the CONTROL  
PERCENTFULL command.

              ACTION        Re-enter the *MinFull* parameter using a  
value of 0 to 100.

---

580           MESSAGE           Cannot read the data set  
                                  *DataSetName* (DBA 580) .

                  MEANING           The MPE intrinsic FREADDIR failed  
                                  while reading the data set *DataSetName*.  
                                  In addition to 580, message 410 is also  
                                  displayed.

                  ACTION           Refer to the *MPE Intrinsic Reference*  
                                  *Manual*, where *n* is the file system error  
                                  number returned from the FCHECK  
                                  intrinsic.

---

580           MESSAGE           *NewBaseName* already exists and  
                                  REPLACE not requested (DBC 580) .

                  MEANING           Unless the REPLACE parameter is used  
                                  with the COPY command, an existing  
                                  database cannot be overwritten.

                  ACTION           Either give the database copy a new  
                                  name, or use the REPLACE parameter.

---

590           MESSAGE           Cannot read the file label of data  
                                  set *DataSetName* (DBA 590) .

                  MEANING           The MPE intrinsic FREADLABEL  
                                  failed while reading the file label of data  
                                  set *DataSetName*. In addition to 590,  
                                  message 410 is also displayed.

                  ACTION           Refer to message 410.

---

600           MESSAGE           Cannot rename set *DataSetName1* to  
                                  *DataSetName2* (DBA 600) .

                  MEANING           The MPE intrinsic FRENAME  
                                  failed while renaming the data  
                                  set *DataSetName1* to the name  
                                  *DataSetName2*. In addition to 600,  
                                  message 410 is also displayed.

                  ACTION           Refer to message 410.

---

601	MESSAGE	Invalid characters after delete (D) command (DBC 601).
	MEANING	An invalid character or too many characters were entered when using the REDO command with the delete option.
	ACTION	Only D or I (for insert) is allowed after the first D in the REDO command. Redo the command line.
<hr/>		
602	MESSAGE	More than 80 characters on the line (DBC 602).
	MEANING	Too many characters were inserted when using the REDO command.
	ACTION	Only 80 characters are allowed on a command line. Redo the command line using 80 or fewer characters.
<hr/>		
603	MESSAGE	New command exceeds maximum 216 characters allowed (DBC 603).
	MEANING	Too many characters were inserted when using the REDO command.
	ACTION	Only 216 characters per command are allowed. Redo the command using 216 or fewer characters.
<hr/>		
610	MESSAGE	Cannot rename set <i>DataSetName1</i> to TEMPDB <i>DataSetName2</i> (DBA 610).
	MEANING	The MPE intrinsic FRENAME failed while renaming the data set <i>DataSetName1</i> to <i>DataSetName2</i> in the database TEMPDB used in the conversion process. In addition to 610, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.

615        MESSAGE        Cannot reset condition word or root  
file label (DBA 615).

          MEANING        The condition word in the root file label  
could not be reset to a value that would  
allow subsequent access to the database.

          ACTION         Run DBUTIL and purge the database.  
Restore a backup copy.

---

620        MESSAGE        Cannot restructure database while  
recovery in progress (DBA 620).

          MEANING        Database recovery has not been  
completed.

          ACTION         Refer to the *TurboIMAGE/XL DBMS  
Reference Manual* for information on  
DBRECOV STOP-RESTART.

---

630        MESSAGE        Database is corrupt due to prior  
termination of DBAlter (DBA 630).

          MEANING        The database is corrupted because the  
restructuring process was not completed  
during a previous run of the DBAPLUS  
program (DBAlter).

          ACTION         Run the DBAlter program again to  
finish the restructuring process.

---

640        MESSAGE        Database is not in a consistent  
state (DBA 640).

          MEANING        A database operation was started and  
has not completed. In addition to 640,  
message 370 is also displayed.

          ACTION         Finish the database operation before  
running the DBAPLUS program.  
Refer to message 370 for additional  
instructions.

---

645	MESSAGE	Database contains errors and no "FIX" requested (DBA 645).
	MEANING	The database contains errors. The DBCPLUS JCW is set to 210 so that a batch job can detect the damaged database.
	ACTION	Use the FIX command to correct errors.
<hr/>		
650	MESSAGE	Security violation (DBA 650).
	MEANING	Either the logon user, group, or account do not match what is in the change file or you do not have MPE access to the change file.
	ACTION	Verify that you have logged on correctly and have access to the change file.
<hr/>		
660	MESSAGE	<i>N</i> existing records of data set <i>M</i> exceed the capacity of <i>cap</i> (DBA 660).
	MEANING	The new capacity is less than the number of existing entries in the data set.
	ACTION	Run the DBChange program again and specify a capacity that is greater than the number of existing entries. If the database is corrupt, purge it, and restore a backup copy.
<hr/>		
663	MESSAGE	Data set <i>DataSetName</i> is full (DBA 663).
	MEANING	The key value cannot be added to the master data set <i>DataSetName</i> because the data set is full.
	ACTION	Change the data set capacity and reissue the FIX command.
<hr/>		

670           MESSAGE           Data set *DataSetNum* does not exist  
                                  in the change file (DBA 670).

                  MEANING           The change file contains inconsistent  
                                  data set information. The data set  
                                  number *DataSetNum* was not found in  
                                  the change file.

                  ACTION           Run the DBCPLUS program again and  
                                  create a new change file. If the database  
                                  is corrupt, purge it, and restore a backup  
                                  copy.

---

675           MESSAGE           Error returned from SORT subsystem  
                                  (SORTERR *n*) (DBA 675).

                  MEANING           An error occurred in the SORT  
                                  subsystem.

                  ACTION           Refer to the *MPE XL Utilities*  
                                  *SORT-MERGE/3000 Reference Manual*  
                                  where SORTERR *n* is the error number  
                                  returned from the SORT ERROR  
                                  messages.

---

680           MESSAGE           Not enough stack space to perform  
                                  operation (DBA 680).

                  MEANING           The stack size could not be increased  
                                  to copy the database or to create a  
                                  data set. The Pascal library procedure  
                                  GETHEAP or RTNHEAP failed.

                  ACTION           Notify the system manager.

---

683           MESSAGE           Check file full. Further events  
                                  will not be recorded (DBA 683).

                  MEANING           All events resulting from the previously  
                                  issued CHECK command cannot be  
                                  recorded and displayed.

                  ACTION           Use CHECK PATH/FIX PATH  
                                  commands instead of CHECK  
                                  BASE/FIX BASE commands.

---

685	MESSAGE	Unknown event number encountered in check file (DBA 685).
	MEANING	Either the file <i>DatabaseNameRX</i> is damaged or a DBChange Plus error exists.
	ACTION	Contact your HP representative.
<hr/>		
687	MESSAGE	Cannot write to the check file <i>FileNameRX</i> (FSERR <i>n</i> ) (DBA 687).
	MEANING	The MPE intrinsic FWRITE failed while writing to the check file <i>FileNameRX</i> .
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
690	MESSAGE	Cannot write to new root file (DBA 690).
	MEANING	The MPE intrinsic FWRITE failed while attempting to write to the new root file. In addition to 690, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
700	MESSAGE	Cannot write to the root file label (DBA 700).
	MEANING	The MPE intrinsic FWRITELABEL failed while writing to the file label of either the new or old root file. In addition to 700, message 410 is also displayed.
	ACTION	Refer to message 410.
<hr/>		

710        MESSAGE        Cannot write to the data set  
                              *DataSetName* (DAB 710) .

                 MEANING        The MPE intrinsic FWRITE failed while  
                              writing to the data set *DataSetName*.  
                              In addition to 710, message 410 is also  
                              displayed.

                 ACTION        Refer to the *MPE Intrinsic Reference*  
                              *Manual*, where *n* is the file system error  
                              number returned from the FCHECK  
                              intrinsic.

---

720        MESSAGE        Cannot write to file label of data  
                              set *DataSetName* (DBA 720) .

                 MEANING        The MPE intrinsic FWRITELABEL  
                              failed while writing to the file label of  
                              the data set *DataSetName*. In addition  
                              to 720, message 410 is also displayed.

                 ACTION        Refer to the *MPE Intrinsic Reference*  
                              *Manual*, where *n* is the file system error  
                              number returned from the FCHECK  
                              intrinsic.

---

730        MESSAGE        Cannot write to hashing database  
                              TEMPDB (DBA 730) .

                 MEANING        The MPE intrinsic FWRITEDIR failed  
                              while writing to the temporary database  
                              TEMPDB during the restructuring  
                              process. In addition to 730, message 410  
                              is also displayed.

                 ACTION        Refer to message 410.

---

740        MESSAGE        MPE intrinsic SWITCHDB failure (DBA  
                              740) .

                 MEANING        The MPE intrinsic SWITCHDB failed  
                              while copying or restructuring the  
                              database.

                 ACTION        Notify the system manager.

---

750	MESSAGE	Cannot write to change file (DBA 750).
	MEANING	The MPE intrinsic FWRITEDIR failed while writing to the change file. In addition to 750, message 410 is also displayed.
	ACTION	Refer to the <i>MPE Intrinsic Reference Manual</i> , where <i>n</i> is the file system error number returned from the FCHECK intrinsic.
<hr/>		
900	MESSAGE	Data set <i>DataSetName</i> : expected <i>N</i> entries, processed only <i>M</i> entries (DBA 900).
	MEANING	An error occurred while writing to the data set <i>DataSetName</i> records. <i>N</i> is the number of records in the set. <i>M</i> is the number of records processed successfully.
	ACTION	Informational message only. No action necessary.
<hr/>		
910	MESSAGE	<i>N</i> detail records were lost in altering <i>DataSetName</i> (DBA 910).
	MEANING	An error occurred while writing to the data set <i>DataSetName</i> records. <i>N</i> is the number of records not written.
	ACTION	Informational message only. No action necessary.
<hr/>		
920	MESSAGE	<i>N</i> master records were lost in altering <i>DataSetName</i> (DBA 920).
	MEANING	An error occurred while writing to the data set <i>DataSetName</i> records. <i>N</i> is the number of records <i>not</i> written.
	ACTION	Informational message only. No action necessary.
<hr/>		

## ORDERS Schema

---

The ORDERS database schema that is used for examples throughout this guide is reproduced below before any DBChange Plus modifications.

```
BEGIN DATA BASE ORDERS;
```

```
PASSWORDS:
```

```
 11 CREDIT ;
 12 BUYER ;
 13 SHIP-REC;
 14 CLERK ;
 18 DO-ALL ;
```

```
ITEMS:
```

```
ACCOUNT          , I4(11,12,13,14,18/);
BINNUM           , Z2(/13);
CITY             , X12(12,13,14/11);
CREDIT-RATING   , R2(/14);
DATE            , X6(11,12,13,14,18/);
DELIV-DATE      , X6(/14);
DESCRIPTION     , X20(11,12,13,14,18/);
FIRST-NAME      , X10(14/11);
INITIAL         , U2(14/11);
LAST-NAME       , X16(14/11);
LASTSHIPDATE    , X6(12/);
ONHANDQTY      , J2(14/12);
PRICE           , J2(14/);
PURCH-DATE     , X6(11/14);
QUANTITY       , I1(/14);
STATE          , X2(12,13,14/11);
STOCK#         , U8(11,12,14,18/);
STREET-ADD     , X26(12,13,14/11);
SUPPLIER       , X16(12,13/);
TAX            , J2(14/);
TOTAL          , J2(11,14/);
UNIT-COST      , P8(/12);
ZIP            , X6(12,13,14/11);
```

SETS:

```

NAME:  DATE-MASTER  ,A;
ENTRY:  DATE        (3)
CAPACITY: 365(19);

```

\$CONTROL BLOCKMAX = 640

```

NAME:  CUSTOMER    ,M(14/11,18);
ENTRY:  ACCOUNT    (1)
        LAST-NAME
        FIRST-NAME
        INITIAL
        STREET-ADD
        CITY
        STATE
        ZIP
        CREDIT-RATING
CAPACITY: 201(10);

```

\$CONTROL BLOCKMAX = 512

```

NAME:  PRODUCT     ,M(13,14/12,18);
ENTRY:  STOCK#     (2)
        DESCRIPTION
CAPACITY: 300(16);

```

```

NAME:  SUP-MASTER  ,M(13/12,18);
ENTRY:  SUPPLIER   (1)
        STREET-ADD
        CITY
        STATE
        ZIP
CAPACITY: 201(12);

```

```

NAME:  INVENTORY   ,D(12,14/13,18);
ENTRY:  STOCK#     ( PRODUCT
        ONHANDQTY
        SUPPLIER    (!SUP-MASTER
        UNIT-COST
        LASTSHIPDATE ( DATE-MASTER
        BINNUM
CAPACITY: 450(15);

```

```
NAME: SALES ,D(11/14,18);
ENTRY: ACCOUNT ( CUSTOMER (PURCH-DATE )),
        STOCK# (!PRODUCT ),
        QUANTITY ,
        PRICE ,
        TAX ,
        TOTAL ,
        PURCH-DATE ( DATE-MASTER ),
        DELIV-DATE ( DATE-MASTER );
CAPACITY: 500(10);
```

END.