

Information Management Series
**HP ALLBASE/4GL Developer
Reference Manual Vol 2**

For MPE/iX Systems



**HP Part No. 30601-64204
Printed in U.S.A.**

E0592

FINAL TRIM SIZE : 7.0 in x 8.5 in

NOTICE

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

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, 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 FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY.

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.

SWT/SOFTWARE TECHNOLOGY CENTER
8000 FOOTHILLS BOULEVARD
ROSEVILLE, CA 95678, U.S.A.

© 1986, 1987, 1988, 1989, 1990,1992 by HEWLETT-PACKARD COMPANY

Printing History

New editions of this manual incorporate all material updated since the previous edition. Edition numbers and publication dates are listed below.

First Edition	September 1988
Second Edition	September 1989
Third Edition	February 1990
Fourth Edition	May 1992

List of Effective Pages

This list of effective pages gives the date of the most recent version of each page in this manual.

All pages	May 1992
-----------	----------

Preface

MPE/iX

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with other HP 3000 users you will encounter references to MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Finally, you may encounter references to MPE V, which is the operating system for the HP 3000s not based on the PA-RISC (precision architecture-reduced instruction set computing) architecture. MPE V software can be run

on the PA-RISC (Series 900) HP 3000s in what is known as compatibility mode.

About This Manual

This manual is a comprehensive reference manual for application developers who are using HP ALLBASE/4GL on HP 3000 computer systems running under the MPE/iX operating system. It provides a detailed description of the features and the use of the HP ALLBASE/4GL developer application. It does not describe the procedures and techniques for designing HP ALLBASE/4GL applications, nor does it attempt to describe the basic data processing concepts required for designing and implementing HP ALLBASE/4GL applications. The manual assumes you are familiar with the basic concepts and terminology of HP ALLBASE/4GL.

Before you attempt to use this manual, read the *HP ALLBASE/4GL Developer Self-Paced Training Guide*.

Refer to the *HP ALLBASE/4GL Developer Administration Manual* for details of the administrator application.

Manual Structure

This manual is divided into two volumes containing 14 chapters and 6 appendixes, a glossary, and an index. Volume 1 contains chapters 1 to 10, and volume 2 contains chapters 11 to 14, the appendixes, and the glossary. Both volumes contain a comprehensive index.

Chapter 1 and chapter 2 provide an overview of the structure of HP ALLBASE/4GL and the use of HP ALLBASE/4GL menus and screens. Chapter 3 describes the rules for naming and referencing items in HP ALLBASE/4GL. Chapters 4 to 13 are reference chapters that provide a comprehensive description of the facilities of the HP ALLBASE/4GL developer. Chapter 14 outlines the procedures for generating and testing applications.

The appendixes contain a system collating sequence table, a table of ASCII values, a list of manager errors, details about using HP ALLBASE/4GL versions, details of the MPE/iX environment for HP ALLBASE/4GL, and information about application portability.

The glossary defines the meaning of some terms that are specific to HP ALLBASE/4GL.

A quick reference guide is included as a separately bound supplement to this manual.

Related Publications

In addition to this manual, the following manuals are part of the HP ALLBASE/4GL developer documentation set:

- The *HP ALLBASE/4GL Developer Reference Manual Vol. 1*.
- The *HP ALLBASE/4GL Developer Self-Paced Training Guide*.
- The *HP ALLBASE/4GL Developer Administration Manual*.
- The *HP ALLBASE/4GL Developer Quick Reference Manual*.
- The *HP ALLBASE/4GL Software Update Notice*.

Conventions

The syntax conventions used in this manual are summarized below.

Notation	Description
COMMAND	In syntax statements, command names are shown in bold UPPERCASE letters.
*ARGUMENT	Command arguments are prefixed with the symbol * and are shown in UPPERCASE letters.
<i>Italics</i>	Words in Italics in syntax statements denote a parameter that you must replace with a suitable name. For example: FILE *CLOSE <i>file_name</i>
[]	In syntax statements, square brackets enclose elements that are optional. Several elements stacked inside square brackets means that you may select any one of these elements. For example: $\left[\begin{array}{l} A \\ B \end{array} \right] \text{You can select A, B, or neither.}$
{ }	In syntax statements, braces enclose elements that you must include in the command. Several elements stacked inside braces means that you must choose one of the elements. For example: $\left\{ \begin{array}{l} A \\ B \\ C \end{array} \right\} \text{You must select A or B or C.}$
□	When necessary for clarity, the symbol □ is used in syntax expressions to indicate a required blank or an exact number of blanks.

underlining

When necessary for clarity, underlining indicates user input. For example:

`MPE/iX: hello name.account`

shading

Shaded text represents inverse video text on the terminal screen.

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

...

In syntax expressions, an ellipsis indicates that the preceding element may be repeated.

FINAL TRIM SIZE : 7.0 in x 8.5 in

Contents

11. Logic	
HP ALLBASE/4GL Logic	11-1
Processes	11-1
Functions	11-2
SQL Logic Blocks	11-3
Decision Tables	11-3
Logic Generation	11-6
Logic Menu Screens	11-7
Logic Menu	11-7
Screen Image	11-7
Menu Selections	11-8
Function Keys	11-9
Function Details Screen	11-10
Screen Image	11-10
Field Descriptions	11-10
Function Keys	11-11
Additional Information	11-12
Function Header Screen	11-13
Screen Image	11-13
Field Descriptions	11-13
Function Keys	11-13
Process Details Screen	11-15
Screen Image	11-15
Field Descriptions	11-15
Function Keys	11-16
Additional Information	11-17
Process Header Screen	11-18
Screen Image	11-18
Field Descriptions	11-18
Function Keys	11-18

SQL Logic Block Details Screen	11-20
Screen Image	11-20
Field Descriptions	11-20
Function Keys	11-21
Additional Information	11-21
Generating SQL Logic Blocks	11-21
SQL Logic Block Header Screen	11-22
Screen Image	11-22
Field Descriptions	11-22
Function Keys	11-23
Decision Table Header Screen	11-24
Screen Image	11-24
Field Descriptions	11-24
Function Keys	11-25
Additional Information	11-25
Decision Table Questions Screen	11-26
Screen Image	11-26
Field Descriptions	11-26
Function Keys	11-28
Additional Information	11-28
Decision Table Actions Screen	11-30
Screen Image	11-30
Field Descriptions	11-30
Function Keys	11-31
Decision Table Relationships Screen	11-32
Screen Image	11-32
Field Descriptions	11-32
Function Keys	11-33
Additional Information	11-34

12. Logic Commands

Naming Rules	12-1
Command Name Synonyms	12-2
Disabled Commands	12-2
Logic Command Windows	12-3
Logic Command Length	12-4
Using Literals	12-4
Using Dates	12-4

Master Titles	12-4
BACKGRND Command	12-5
Formats	12-5
Parameters	12-5
Description	12-5
HP4BG Variable	12-6
Background Process Information File	12-7
Background Process Environment	12-7
Limitations	12-7
CALC Command	12-8
Format	12-8
Parameters	12-8
Description	12-10
CALL Command	12-12
Formats	12-12
Parameters	12-12
Description	12-15
CHECK Command	12-19
Formats	12-19
Parameters	12-19
Description	12-21
CLEAR Command	12-23
Formats	12-23
Parameters	12-23
Description	12-24
DATE Command	12-26
Formats	12-26
Parameters	12-26
DECISION Command	12-30
Formats	12-30
Parameters	12-30
DEFINE Command	12-31
Format	12-31
Parameters	12-31
Description	12-31
DISPLAY Command	12-33
Formats	12-33
Parameters	12-33

Description	12-34
DM Command	12-36
DM IMAGE *CLOSE Command	12-36
Format	12-36
Parameters	12-36
Description	12-37
DM IMAGE *LOCK Command	12-37
Formats	12-37
Parameters	12-38
Description	12-40
Locking Strategy	12-40
Implicit Locking	12-41
DM IMAGE *MODE Command	12-43
Formats	12-43
Parameters	12-43
Description	12-44
Choosing an Access Mode	12-46
Using the MODE command with DM IMAGE *MODE	12-46
DM IMAGE *UNLOCK Command	12-48
Format	12-48
Parameters	12-48
Description	12-49
ENTER Command	12-50
Formats	12-50
Parameters	12-50
Description	12-50
EXIT Command	12-52
Format	12-52
Description	12-52
EXTERNAL Command	12-53
Formats	12-53
Parameters	12-53
Description	12-54
FIELD Command	12-64
Formats	12-64
Parameters	12-64
Description	12-65
FILE Command (KSAM)	12-68

Contents-4

Formats	12-68
Parameters	12-69
Description	12-71
File Operations	12-72
File Errors	12-75
KSAM File Locking	12-76
FILE Command (Serial)	12-80
Formats	12-80
Parameters	12-81
Description	12-82
Serial File Creation	12-83
Temporary Serial Files	12-83
Dynamic File Naming	12-84
Error Handling	12-84
File Errors	12-84
File Record Buffers	12-85
Serial File Locking	12-85
Transaction Logging	12-85
File Operations	12-85
FILE Command (HP ALLBASE/SQL)	12-88
Formats	12-88
Parameters	12-88
Description	12-89
Error Handling	12-90
File Operations	12-92
Using HP ALLBASE/SQL Data	12-93
FILE Command (HP TurboIMAGE/iX)	12-94
Formats	12-94
Parameters	12-95
Description	12-98
File Operations	12-100
Data Set Locking	12-106
Implicit File Locking	12-107
IF Command	12-111
Formats	12-111
Parameters	12-112
Description	12-114
IFLOOP Command	12-117

Format	12-117
Parameters	12-117
Description	12-121
KEYS Command	12-123
Formats	12-123
Parameters	12-123
Description	12-123
LENGTH Command	12-125
Format	12-125
Parameters	12-125
Description	12-126
LINK Command	12-127
Formats	12-127
Parameters	12-127
Description	12-129
LINKLOOP Command	12-130
Formats	12-130
Parameters	12-130
Description	12-132
MATH Command	12-134
Format	12-134
Parameters	12-134
Description	12-137
MATHLOOP Command	12-138
Formats	12-138
Parameters	12-138
Description	12-141
MESSAGE Command	12-143
Formats	12-143
Parameters	12-143
Description	12-143
MODE Command	12-145
Formats	12-145
Parameters	12-145
Description	12-145
MOVE Command	12-149
Format	12-149
Parameters	12-149

Contents-6

Description	12-150
MOVELOOP Command	12-152
Format	12-152
Parameters	12-152
Description	12-154
NOTE Command	12-156
Format	12-156
Parameters	12-156
Description	12-156
OFF Command	12-157
Formats	12-157
Parameters	12-157
Description	12-157
ON Command	12-158
Formats	12-158
Parameters	12-158
Description	12-158
PRINT Command	12-160
Format	12-160
Parameters	12-160
Description	12-160
PROCEED Command	12-161
Formats	12-161
Parameters	12-161
Description	12-161
REPORT Command	12-163
Formats	12-163
Parameters	12-163
Description	12-163
SCREEN Command	12-165
Formats	12-165
Parameters	12-165
Description	12-165
SCROLL Command	12-167
Format	12-167
Parameters	12-167
Description	12-167
SELECT Command	12-169

Format	12-169
Parameters	12-169
Description	12-170
SERIES Command	12-172
Format	12-172
Parameters	12-172
Description	12-173
SHOW Command	12-175
Formats	12-175
Parameters	12-175
Description	12-175
SQL Command	12-178
Formats	12-178
Parameters	12-178
Description	12-179
TIE Command	12-181
Format	12-181
Parameters	12-181
Description	12-181
TOP Command	12-183
Format	12-183
Description	12-183
TRANSACT Command	12-184
Formats	12-184
Parameters	12-184
Description	12-184
Limitations	12-186
UPDATE Command	12-188
Format	12-188
Description	12-188
VALIDATE Command	12-189
Formats	12-189
Parameters	12-189
Description	12-191
VISIT Command	12-192
Formats	12-192
Parameters	12-192
Description	12-192

Contents-8

WINDOW Command	12-193
Formats	12-193
Parameters	12-193
Description	12-193
ZIP Command	12-196
Format	12-196
Description	12-196

13. Utilities

Utilities Menu	13-1
Screen Image	13-2
Menu Selections	13-2
Function Keys	13-3
Catalog Printing Screen	13-4
Screen Image	13-4
Field Descriptions	13-4
Function Keys	13-5
Additional Information	13-5
Catalog Copying Screen	13-7
Screen Image	13-7
Field Descriptions	13-7
Function Keys	13-8
Additional Information	13-8
Catalog Deletions Screen	13-9
Screen Image	13-9
Field Descriptions	13-9
Function Keys	13-9
Additional Information	13-10
Catalog Display Screen	13-11
Screen Image	13-11
Function Keys	13-12
Additional Information	13-12
HP ALLBASE/SQL Table and Select List Display	13-12
Additional Utilities	13-13
HP4STOA and HP4ATOS	13-13
System Variables	13-14
Keywords File	13-14
Running HP4STOA	13-14

Command Line Options	13-14
Object Types	13-16
Running HP4ATOS	13-18
Command Line Options	13-18
Exit Values	13-20
HP4REMK	13-20
HP4REOD	13-21
Checking S-files for Inconsistencies	13-22
Repairing Inconsistencies	13-24
14. Application Generation and Testing	
Application Generation	14-1
Screen Image	14-2
Menu Selections	14-2
Function Keys	14-3
Additional Information	14-3
SQL Component Generation	14-5
Generate Error Log File	14-5
Application Testing	14-6
Using Databases during Application Testing	14-6
Menu Bypass Facility	14-6
Training Mode	14-7
Using Trace Mode	14-7
Unloading Applications	14-8
A. Native Language Support	
HP ALLBASE/4GL Collating Sequence	A-1
B. HP ALLBASE/4GL Versions	
Versions	B-1
Defining Versions	B-1
Developing Versions	B-2
Limitations	B-3
Unloading and Loading Versions	B-4
Version Data File Reformat	B-4

C. Data Manager Error Messages	
MPE/iX Miscellaneous Errors	C-1
Data File and Data Set Errors	C-3
HP ALLBASE/SQL Messages	C-8
HP ALLBASE/4GL-HP ALLBASE/SQL Interface Error Messages	C-9
HP TurboIMAGE/iX Data Set Errors	C-11
D. Operating System Environment	
HP ALLBASE/4GL Files	D-2
S-Files	D-2
Copying S-Files	D-3
Program Files	D-4
MPE/iX Command Files	D-6
Application Databases	D-7
Terminal Support Files	D-8
MPE/iX Account Structure	D-8
HP4APPN[R] Group	D-9
HP4DATA[R] Group	D-9
HP4DBM[R] Group	D-9
HP4FS[R] Group	D-10
HP4S[R] Group	D-10
HP4SQL[R] Group	D-10
HP4TI[R] Group	D-11
Formal File Designators	D-11
MPE/iX Variables	D-13
S-File Size Limits	D-15
Using Remake and Reorder	D-16
Increasing the S-File Limits	D-16
Message Catalogs	D-17
Multiple HP ALLBASE/4GL Systems	D-18
Run-Only Applications	D-19
Start-up Command File	D-19
Screen Image Printing	D-21
HP4SCREEN_PRINT Variable	D-22
Hardware and MPE Clock Settings	D-22
HP ALLBASE/4GL Terminals	D-23
Terminal Support Files	D-23

Touchscreen Terminals	D-24
Terminal Initialization and Resetting	D-24
Terminal Configurations	D-25
E. HP ALLBASE/4GL Limits	
Item Limits	E-1
Other Limits	E-2
F. Application Portability	
Introduction	F-1
Portability Mechanism	F-1
Application Definition File	F-2
Database Module Files	F-3
File Structure File	F-3
Application Design Considerations	F-4
Application and Version Names	F-4
Screen Field Behaviour	F-5
Auto-Flow Fields	F-5
Justification and Case Shifting	F-5
Error Field Highlighting	F-6
Data Manager Differences	F-6
External File Names	F-6
Limits	F-7
Locking Mechanism	F-7
Data File Recovery	F-8
Reporting Environment	F-8
Using External Programs	F-9
Administrator Functionality	F-9
G. Developer Reference Chart	
Glossary	
Index	

Tables

D-1. Program Files in the Developer Environment	D-5
D-2. Program Files in the Run-Time	D-6
D-3. Command Files in the PUB.SYS Group	D-7
D-4. Command Files in the HP4GL[R].SYS Group	D-7
D-5. Terminal Support Files	D-23

FINAL TRIM SIZE : 7.0 in x 8.5 in

Logic

HP ALLBASE/4GL Logic

HP ALLBASE/4GL uses the following logic constructs:

- Processes.
- Functions.
- SQL logic blocks.
- Decision tables.

The screens used to define these logic constructs are accessed from the logic menu.

Processes

An HP ALLBASE/4GL process is similar to a program in a conventional language.

When a process starts, most parts of the HP ALLBASE/4GL environment are initialized. At the start of a process, the following actions occur:

- Any current process, screen, function, decision table, or report is terminated immediately.
- Any incomplete HP ALLBASE/SQL transactions are reversed, and all current HP ALLBASE/SQL cursors are closed.
- All HP TurboIMAGE/iX data sets are closed, and all locks are released.
- All application data files are closed.
- All file record buffers, work areas and screen buffers are cleared.

When HP ALLBASE/4GL initiates a process, the contents or state of the following items remain unchanged:

- Communication area fields other than *PROCESS.
- Scratch-pad fields.
- Switches.
- Variables.

At the conclusion of the process, control returns to the previous menu. This means that control may not return to the entity that initiated the process. For example, if HP ALLBASE/4GL executes the process as a decision table action, control does not return to the decision table at the conclusion of the process.

Creating a process involves two stages. You complete the process header screen first, and then define the logic block for the process using the process details screen.

A process logic block consists of up to 99 steps, where each step contains one or more logic commands. If you need a process containing more than 99 steps, you can use logic commands to invoke other functions, decision tables, reports or screens from within a process. Chapter 12 describes the HP ALLBASE/4GL logic commands.

Functions

An HP ALLBASE/4GL function is similar to a subroutine in a conventional language.

When HP ALLBASE/4GL finishes executing a function, it resumes running the application at the point where the function was initiated.

Functions may be nested. That is, one function may call another function. The maximum depth of nesting is 99, and recursion is permitted.

Creating a function also involves completing a function header screen and a function detail screen. The function logic block defined in the function detail screen is similar to a process logic block.

11-2 Logic

SQL Logic Blocks

SQL logic blocks are the mechanism for passing SQL commands to HP ALLBASE/SQL from HP ALLBASE/4GL. An SQL logic block consists of a header and the logic block details. The header defines the name of the SQL block and some documentation, while the block details contain the SQL commands. An SQL logic block is invoked by a logic command (the SQL command) in a process or function.

Refer to chapter 9 for more information about SQL logic blocks.

Decision Tables

HP ALLBASE/4GL decision tables allow you to define a series of interrelated decisions and the actions to be taken as a result of those decisions. Decision tables can be regarded as being equivalent to multiple levels of IF-THEN-ELSE statements in a conventional language.

A decision table consists of three major elements:

- Decision table questions.
- Decision table actions.
- Decision table relationships.

Each decision table can test up to eight different questions. The decision table questions are simple IF statements. Each IF statement can test two operands for equality or otherwise, or test a single operand or switch for its status.

The decision table actions are any actions that you can call from a menu. They are:

- B background process
- D menu or data screen
- F function
- H help screen
- P process
- R report
- X external program

The decision table relationships define up to 31 different combinations of outcomes for the decision table questions. For each combination of outcomes, the decision table relationships define the actions to be performed and the sequence in which the actions are performed.

You can think of a typical decision table as being a table in the following format.

Question	Columns					
	1	2	3	4	...	31
1	Y	-	N	N		Y
2	Y	Y	N	Y		Y
3	N	Y	-	N		-
4	N	N	Y	N		Y
5	-	Y	N	-		N
6	-	N	Y	Y		Y
7	Y	Y	-	-		-
8	N	N	Y	Y		N

11-4 Logic

Actions	Order of Execution					
	1	2	3	4	5	6
1	1	8	2	0		4
2	2	5	1	1		3
3	3	2	0	0		1
4	4	7	0	0		5
5	5	3	0	0		0
6	6	4	3	0		0
7	7	1	4	0		2
8	8	6	0	0		0

Each column in this table defines one possible combination of outcomes for the decision table questions, and the actions to be performed if this particular combination of outcomes occurs.

Within each column you can specify that for each question the outcome must be true (Y), false (N), or don't care (-). The numbers in the lower part of the table define the actions that the table executes, and the order of execution of the actions if that combination of outcomes occurs. The number 0 indicates that the corresponding action is not executed for that particular combination of outcomes. For example, in the above sample table, the entries in column number 3 specify that action number 2 is executed first, followed by action number 1, action number 6 and action number 7 in order. Actions number 3, number 4, number 5, and number 8 are not executed.

The order of execution of the decision table actions does not depend on the number of the action or the order of entry.

When HP ALLBASE/4GL executes a decision table, it scans the columns of the table starting from column number one. HP ALLBASE/4GL checks each column until it finds a column that matches the actual outcomes of the decision table questions. HP ALLBASE/4GL then executes the actions defined for that column, and terminates the decision table.

To define a decision table, you need to complete four screens. They are:

- The decision table header screen.
- The decision table questions screen.
- The decision table actions screen.
- The decision table relationships screen.

Each of these screens allows you to define part of the decision table. The decision table header screen allows you to define the name of the table and enter a description for documentation purposes.

The decision table questions screen and the decision table actions screen allow you to define the questions the table tests, and the actions the table can execute. However, these screens do not define the combinations of outcomes of the questions, or the sequence of execution of the actions.

The decision table relationships screen allows you to step through the table one column at a time, and define the contents of each column. For each column of the table, you can define the required answer for each question as true (Y), false (N), or don't care (-). For each column, you must also define the actions to be performed if that combination of outcomes occurs, and the order of execution of the actions. You don't need to complete all 31 columns of the table if your application doesn't need that many. However, the columns must start from number one, and must be sequentially numbered.

Logic Generation

All HP ALLBASE/4GL logic must be generated before the application can use it at run-time. Most screens in the logic menu have a **Generate** function key to allow you to generate a single component. You can also use the *Generates* menu to generate groups of logic components.

11-6 Logic

Logic Menu Screens

The following subsections describe the screens in the HP ALLBASE/4GL logic menu.

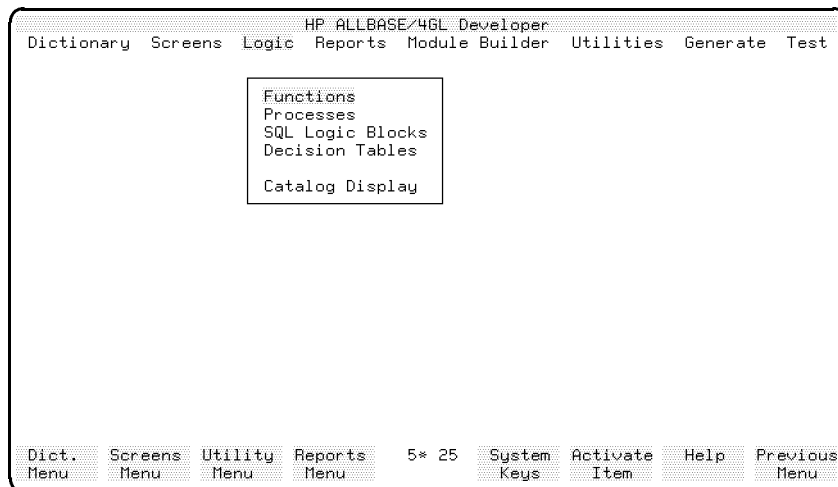
Each subsection describes screens under the following subheadings:

- Screen Image.
- Field Descriptions.
- Function Keys.
- Additional Information.

Logic Menu

The main logic menu allows you to access the screens necessary to define functions, processes, SQL logic blocks, and decision tables.

Screen Image



Menu Selections

The main logic menu allows you to access four submenus and the catalog display screen.

Functions. This selection displays the functions menu, which allows you to access the function header screen and the function details screen. The function header screen allows you to assign a name to a function and enter a description for documentation purposes. The function details screen allows you to define the logic block of a function.

Processes. This selection displays the process menu, which allows you to access the process header screen and the process details screen. The process header screen allows you to assign a name to a process and enter a description for documentation purposes. The process details screen allows you to define the logic block of a process.

SQL Logic Blocks. This selection displays the SQL logic blocks menu, which allows you to access the SQL logic block header screen and the SQL logic block details screen. The SQL logic block header screen allows you to assign a name to an SQL logic block and enter a description for documentation purposes. The SQL logic block details screen allows you to enter the SQL commands that make up the SQL logic block.

Decision Tables. This selection displays the decision table menu, which allows you to access the decision table header screen, the decision table questions screen, the decision table actions screen, and the decision table relationships screen.

The decision table header screen allows you to name a decision table and enter a description for documentation purposes.

The decision table questions screen allows you to define the questions for the decision table tests.

The decision table actions screen allows you to name the actions that the decision table may execute.

The decision table relationships screen allows you to define combinations of outcomes for the decision table questions, and the sequence of actions to be executed for each combination of outcomes.

11-8 Logic

Catalog Display. This selection displays the catalog display screen, allowing you to display the names of existing application components. Refer to chapter 13 for details.

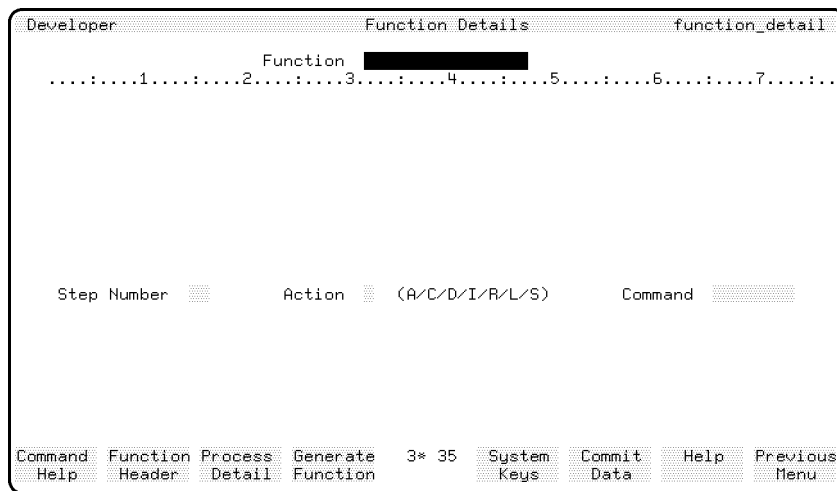
Function Keys

- f1** **Dict. Menu** Cancels the current screen and displays the dictionary menu.
- f2** **Screens Menu** Cancels the current screen and displays the screens menu.
- f3** **Utility Menu** Cancels the current screen and displays the utilities menu.
- f4** **Reports Menu** Cancels the current screen and displays the reports menu.

Function Details Screen

This screen allows you to enter the logic commands that make up a function logic block. You must complete the function header screen before you can use this screen to define the function logic block.

Screen Image



Field Descriptions

Function. Enter the name of a function defined on the function header screen.

Step Number. Enter the number of the step to be added, modified or deleted. Step numbers must be in the range 1 to 99.

The top half of the screen displays up to 10 steps for the function. This display is always oriented around the step number entered. Only the first 76 characters of each step are displayed.

Action. The following actions may be performed on the step corresponding to the step number:

11-10 Logic

- A add a new step.
- C change an existing step.
- D delete a step.
- I insert a new step before this one.
- R renumber steps.
- L list all steps.
- S swap display pages. (HP ALLBASE/4GL displays steps in groups of 10 from 1 to 99 — that is, 1 to 10, 11 to 20, 21 to 30 and so on, in the display area of the screen.)

If the current step is a new step, the action defaults to A. Otherwise the action defaults to C.

Command. Enter the name of the logic command to be performed by this step. If the step already exists, HP ALLBASE/4GL displays the command name automatically and skips this field.

If the command uses a window, HP ALLBASE/4GL displays it automatically.

If you leave this field blank, HP ALLBASE/4GL displays a general purpose unformatted window to allow free format entry of the command.

Function Keys

- f1** **Command Help** Displays a series of help screens containing a summary of the HP ALLBASE/4GL logic commands.
- f2** **Function Header** Cancels the current screen and displays the function header screen.
- f3** **Process Detail** Cancels the current screen and displays the process details screen.
- f4** **Generate Function** Generates the function.

Additional Information

If you enter the number of an existing step that uses a window, HP ALLBASE/4GL checks that the contents of the fields will fit into the window's fields. If so, HP ALLBASE/4GL displays the window with the command formatted into the appropriate fields. If not, HP ALLBASE/4GL displays the command on the general purpose window.

You must always press the **Commit Data** function key after you delete, insert or change a command in the function. HP ALLBASE/4GL does not update the appropriate internal file until you commit the screen data.

The D action blanks out the step from the screen. However the step is not deleted from the actual logic block until you press the **Commit Data** function key.

When you add or delete a step, HP ALLBASE/4GL renumbers the remaining steps in the logic block. HP ALLBASE/4GL also attempts to correct references to the renumbered steps in logic commands such as ENTER. However, after you add or delete a step, you should always check that steps have been renumbered correctly, and references to renumbered steps are correct.

You must generate a function before it can be executed by the application at run time.

Function Header Screen

This screen allows you to name a function and enter a description for documentation purposes.

Screen Image

The screenshot shows a terminal window titled "Function Header" with the following fields and controls:

- Function Name:** A text input field containing a redacted name.
- Description:** A large text area for entering a description.
- Last Modification:** A field for the date and time of the last modification.
- Date:** A field for the date.
- Time:** A field for the time.
- Command Menu:** A row of buttons at the bottom: Command Help, Function Detail, Process Menu, SQL Blk Menu, 5* 33, System Keys, Commit Data, Help, and Previous Menu.

Field Descriptions

Function Name. Enter the name of the function (up to 16 characters).

Description. Documentation fields. Enter a description of the function for documentation purposes. HP ALLBASE/4GL automatically updates the date and time fields to show the date and time of the last modification.

Function Keys

- (f1) **Command Help** Displays a series of help screens containing a summary of the HP ALLBASE/4GL logic commands.
- (f2) **Function Detail** Cancels the current screen and displays the function detail screen.

- f3** **Process Menu** Cancels the current screen and displays the processes menu.
- f4** **SQL Blk Menu** Cancels the current screen and displays the SQL logic block menu.

11-14 Logic

Process Details Screen

This screen allows you to define the logic commands that make up a process logic block. You must complete the process header screen before you can use this screen to define the process logic block.

Screen Image

The screenshot shows a terminal window titled "Process Details" with a sub-header "process_detail". The main content area displays a table with three columns: "Step Number", "Action", and "Command". The "Action" column contains the text "(A/C/D/I/R/L/S)". Above the table, there is a "Process" label followed by a blacked-out box. Below the table, there is a footer area with several menu options: "Command Help", "Process Header", "Function Detail", "Generate Process", "3* 32", "System Keys", "Commit Data", "Help", and "Previous Menu".

Step Number	Action	Command
	(A/C/D/I/R/L/S)	

Field Descriptions

Process. Enter the name of a process defined on the process header screen.

Step Number. Enter the number of the step to be added, modified or deleted. Step numbers must be in the range 1 to 99.

The top half of the screen displays up to 10 steps for the process. This display is always oriented around the step number entered. Only the first 76 characters of each step are displayed.

Action. The following actions may be performed on the step corresponding to the step number you enter:

- A add a new step.
- C change an existing step.
- D delete a step.
- I insert a new step before this one.
- R renumber steps.
- L list all steps.
- S swap display pages. (HP ALLBASE/4GL displays steps in groups of 10 from 1 to 99 — that is, 1 to 10, 11 to 20, 21 to 30 and so on, in the display area of the screen.)

If the current step is a new step, the action defaults to A. Otherwise, the action defaults to C.

Command. Enter the name of the logic command to be performed by this step. If the step already exists, HP ALLBASE/4GL displays the command name automatically and skips this field.

If the command uses a window, HP ALLBASE/4GL displays it automatically.

If you leave this field blank, HP ALLBASE/4GL displays a general purpose unformatted window, to allow free format entry of the command.

Function Keys

- f1** **Command Help** Displays a series of help screen containing a summary of the HP ALLBASE/4GL logic commands.
- f2** **Process Header** Cancels the current screen and displays the process header screen.
- f3** **Function Detail** Cancels the current screen and displays the function details screen.
- f4** **Generate Process** Generates the process.

11-16 Logic

Additional Information

If you enter the number of an existing step that uses a window, HP ALLBASE/4GL checks that the contents of the fields will fit into the window's fields. If so, HP ALLBASE/4GL displays the window with the command formatted into the appropriate fields. If not, HP ALLBASE/4GL displays the command on the general purpose window.

You must always press the **Commit Data** function key after you delete, insert or change a command in the process. The system does not update the appropriate HP ALLBASE/4GL file until you commit the screen data.

The D- action blanks out the step from the screen. However the step is not deleted from the actual logic block until you press the **Commit Data** function key.

When you add or delete a step, HP ALLBASE/4GL renumbers the remaining steps in the logic block. HP ALLBASE/4GL also attempts to correct references to the renumbered steps in logic commands such as ENTER. However, after you add or delete a step, you should always check that steps have been renumbered correctly, and references to renumbered steps are correct.

You must generate a process before it can be used by the application at run time.

Process Header Screen

This screen allows you to name a process and enter a description for documentation purposes.

Screen Image

The screenshot shows a terminal window titled "Developer Process Header process_header". The main area contains three input fields: "Process Name" with a blacked-out value, "Description" with a shaded area, and "Last Modification:" with sub-fields for "Date" and "Time". At the bottom, a command menu lists: Command Help, Process Detail, Function Menu, SQL Blk Menu, 5* 34, System Keys, Commit Data, Help, and Previous Menu.

Field Descriptions

Process Name. Enter the name of the process (up to 16 characters).

Description. Documentation fields. Enter a description of the process for documentation purposes. HP ALLBASE/4GL automatically updates the date and time fields to show the date and time of the last modification.

Function Keys

- (f1) **Command Help** Displays a series of help screens containing a summary of the HP ALLBASE/4GL logic commands.
- (f2) **Process Detail** Cancels the current screen and displays the process details screen.

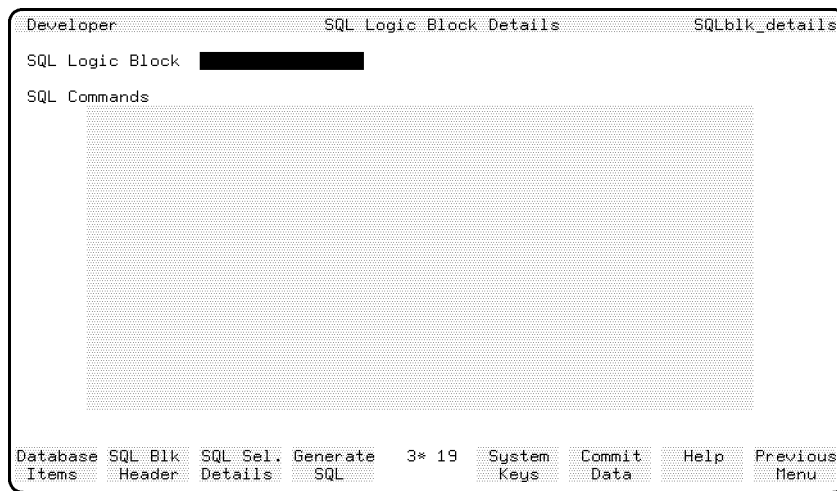
11-18 Logic

- f3** **Function Menu** Cancels the current screen and displays the functions menu.
- f4** **SQL Blk Menu** Cancels the current screen and displays the SQL logic block menu.

SQL Logic Block Details Screen

This screen allows you to enter the SQL commands that make up an SQL logic block.

Screen Image



Field Descriptions

SQL Logic Block. Enter the name of an SQL logic block that has been defined on the SQL logic block header screen.

SQL Commands. This is a free format text entry area that allows you to enter one or more valid SQL commands. Terminate each command with a semicolon.

If the SQL block contains a SELECT command, this must be the only command in the SQL logic block. If the SQL block does not contain a SELECT command, it can contain up to eight SQL commands that generate into stored database sections. The SQL logic block may contain additional SQL commands that do **not** generate into stored database sections.

11-20 Logic

The SQL commands that do not generate into stored database sections are:

- BEGIN WORK.
- COMMIT WORK.
- ROLLBACK WORK.
- SAVEPOINT.
- SQLEXPLAIN.

There are some restrictions on the commands you can use in SQL logic blocks. Refer to chapter 9 for details of these restrictions.

Function Keys

- f1** Database Items Cancels the current screen and displays the dictionary database items menu.
- f2** SQL Blk Header Cancels the current screen and displays the SQL logic block header screen.
- f3** SQL Sel. Details Cancels the current screen and displays the SQL logic block details screen.
- f4** Generate SQL Generates the SQL logic block.

Additional Information

You can include a note or a comment in the SQL logic block with the following statement:

NOTE [text of note ...]

Refer to the *HP ALLBASE/SQL Reference Manual* for further information about the use and syntax of HP ALLBASE/SQL commands.

Generating SQL Logic Blocks

You must generate an SQL logic block before the application can use it. To generate an SQL logic block, press the **Generate SQL** function key or select the *SQL Logic Blocks* option on the generates menu selected from the main menu.

SQL Logic Block Header Screen

This screen allows you to name an SQL logic block and enter a description for documentation purposes.

Screen Image

Developer: SQL Logic Block Header SQLblk_header

SQL Logic Block Name [REDACTED]

Description

Last Modification: Date Time

Database SQL Blk SQL Sel. 5* 33 System Commit Help Previous
Items Detail Header Keys Data Menu

Field Descriptions

SQL Logic Block Name. Enter the name of the SQL logic block (up to 16 characters).

Description. Documentation fields. Enter a description of the SQL logic block for documentation purposes. HP ALLBASE/4GL automatically updates the date and time fields to show the date and time of the last modification.

Function Keys

- f1** **Database Items** Cancels the current screen and displays the dictionary data base items menu.
- f2** **SQL Blk Detail** Cancels the current screen and displays the SQL logic block details screen.
- f3** **SQL Sel. Header** Cancels the current screen and displays the SQL select list header screen.

Decision Table Header Screen

This screen allows you to name a decision table and enter a description for documentation purposes.

Screen Image

Developer: Decision Table Header dtable_header

Decision Table Name [REDACTED] Secured (Y/N)

Description

Last Modification: Date Time

Question Actions R'ships Generate D.Table 5* 30 System Keys Commit Data Help Previous Menu

Field Descriptions

Decision Table Name. Enter the name of the decision table (up to 16 characters).

Secured. If you wish to protect this decision table from unauthorized modification, enter **Y** after you have completed this screen; otherwise enter **N**.

If you enter **Y**, this decision table can only be modified by a developer who has signed on to the application under the correct development security code.

Description. Documentation fields. Enter a description of the decision table for documentation purposes. HP ALLBASE/4GL automatically updates the date and time fields to show the date and time of the last modification.

Function Keys

- (f1) **Questions** Cancels the current screen and displays the decision table questions screen.
- (f2) **Actions** Cancels the current screen and displays the decision table actions screen.
- (f3) **R'ships** Cancels the current screen and displays the decision table relationships screen.
- (f4) **Generate D.Table** Generates the decision table.

Additional Information

You must name a new decision table using this screen before you can specify any other details for the decision table.

Decision Table Questions Screen

This screen allows you to define the questions that the decision table tests.

Screen Image

The screenshot shows a window titled "Decision Table Questions" with a developer toolbar at the top. The toolbar includes buttons for "Header", "Actions", "R'ships", "Generate D.Table", "4* 25", "System Keys", "Commit Data", "Help", and "Previous Menu". The main area contains a table with the following structure:

Decision Table	Secured	(Y/N)
Questions		
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>
IF Data name 1: Data name 2:	IF Test	<input type="text"/>

Field Descriptions

Decision Table Name. Enter the name of a decision table that has been defined on the decision table header screen.

Secured. Display-only field indicating the status of the *Secured* field on the decision table header screen.

Data name 1. You must complete this field for each decision table question.

Enter the name of the first operand for the logic IF test. The entry in this field can be one of the following:

- Whole number.
- Literal.
- Numeric or alphanumeric constant.

11-26 Logic

- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area field name.
- Switch name or number.

Data name 2. Enter the name of the second operand for the logic IF test. This field is only required if the particular decision table question uses a relational operator to perform the IF test. The acceptable entries for this field are the same as the entries for the *Data name 1* field, except that you cannot use a switch name or number in this field.

IF Test. You must complete this field for each decision table question.

Enter the IF test to be applied to the operands in the *Data name 1* field (and optionally the *Data name 2* field). The IF test can test the relationship between the two operands according to a relational operator, or test the content of a single operand (Data name 1), or test the status of a switch.

If you are testing the relationship between two different operands, the entry in this field must be one of the following relational operators:

=	equal to
< >	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

The edit codes for the operands used with relational operators determine the nature of the relational test. If you are comparing two numeric values, all relation tests are performed according to strict numerical order. If you are comparing two alphanumeric fields or comparing a field with a literal, the

comparison is performed on a character-by-character basis according to the system collating sequence. (Refer to appendix A.)

Note

If you are using a relational operator to test the relationship between two operands, make sure the two operands both have the same edit code, or contain the same type of data.

If you are testing the contents of the operand in the *Data name 1* field, the entry in this field must be one of *BLANK, *ALPHA, *NUMERIC, or *NULL. The *NULL contents test is only valid for testing the contents of a file record field for an HP ALLBASE/SQL table or select list.

If you are testing the status of a switch, the entry in this field must be either *ON or *OFF.

Function Keys

- f1** **Header** Cancels the current screen and displays the decision table header screen.
- f2** **Actions** Cancels the current screen and displays the decision table actions screen.
- f3** **R'ships** Cancels the current screen and displays the decision table relationships screen.
- f4** **Generate D.Table** Generates the decision table.

Additional Information

The screen allows you to enter up to eight questions for the table. Each question must be expressed in the same format as a logic IF command. That is, one of the following forms:

operand relational-operator *operand*
operand *BLANK
operand *ALPHA
operand *NUMERIC
operand *NULL
switch-name *ON
switch-name *OFF

The entry in the *Data name 1* field corresponds to the first of the operands,
The entry in the *IF Test* field must be one of: relational-operator, *BLANK,
*ALPHA, *NUMERIC, *NULL, *ON, *OFF.

An entry in the *Data name 2* field is only required if the IF test uses a
relational operator, and corresponds to the second operand.

Refer to chapter 12 for details of the logic IF command.

Decision Table Actions Screen

This screen allows you to define the possible actions that the decision table may execute.

Screen Image

Developer: Decision Table Actions dtableactions

Decision Table [REDACTED] Secured (Y/N)

Action Number	Action (Prefixes: B-/D-/F-/H-/P-/R-/X-)
1st	[REDACTED]
2nd	[REDACTED]
3rd	[REDACTED]
4th	[REDACTED]
5th	[REDACTED]
6th	[REDACTED]
7th	[REDACTED]
8th	[REDACTED]

Header Question R'ships Generate D.Table 4* 30 System Keys Commit Data Help Previous Menu

Field Descriptions

Decision Table Name. Enter the name of a decision table defined on the decision table header screen.

Secured. Display-only field indicating the status of the *Secured* field on the decision table header screen.

Action Number and Action. These fields allow you to define up to eight actions for the decision table. The sequence of entry does not define the sequence of execution. You must define the sequence of execution in the decision table relationships screen.

You must define each action with a code letter (uppercase) specifying the type of action, followed by a hyphen, and then the name of the item to be executed. For example, *D-get_name* specifies the screen *get_name*.

11-30 Logic

The actions you can specify are:

B	background process
D	menu or data screen
F	function
H	help screen
P	process
R	report
X	external program

Function Keys

- f1** **Header** Cancels the current screen and displays the decision table header screen.
- f2** **Questions** Cancels the current screen and displays the decision table questions screen.
- f3** **R'ships** Cancels the current screen and displays the decision table relationships screen.
- f4** **Generate D.Table** Generates the decision table.

Decision Table Relationships Screen

This screen allows you to define the columns of the decision table. Each decision table column defines one combination of possible outcomes for the decision table questions, and the sequence of actions to be performed if that particular combination of outcomes occurs. A decision table can have up to 31 columns.

You must complete the entire screen to define each column of the decision table.

Screen Image

The screenshot shows a terminal-style interface for the 'Decision Table Relationships' screen. At the top, it displays 'Developer', 'Decision Table Relationships', and 'dtablelships'. Below this, there are several fields: 'Decision Table' (with a redacted name), 'Secured (Y/N)', 'Column Number', and 'Amendment Mode (A/D)'. The main part of the screen is a table for defining columns. The table has four columns: 'Question Number', 'Response (Y/N/-)', 'Possible Action', and 'Order of Execution (1-8 or 0 to ignore)'. The rows are numbered 1 through 8, with 'Response' and 'Order of Execution' columns containing dots. Below the table is a 'Column Definition' field. At the bottom, there is a menu bar with options: 'Header', 'Question', 'Actions', 'Generate D.Table', '4* 25', 'System Keys', 'Commit Data', 'Help', and 'Previous Menu'.

Field Descriptions

Decision Table Name. Enter the name of a decision table whose questions and actions have already been defined on the appropriate screens.

Secured. Display-only field indicating the status of the *Secured* field on the decision table header screen.

Column Number. Enter a number in the range 1 to 31, specifying the column to be defined or modified.

11-32 Logic

Amendment Mode. May be either:

- A add a new column or amend an existing column.
- D delete an existing column.

The next set of fields defines the interpretation of each question required for this column and the sequence of actions to be executed if the outcomes for the questions match the outcomes specified for the column.

Question. For each decision table question, specify one of the following options in the response area:

- Y the question must be true.
- N the question must be false.
- the question has no bearing on this column (that is, *don't care*).

Possible Action. These fields display the specified decision table actions for reference purposes.

Order of Execution. In these fields, you can specify the sequence in which HP ALLBASE/4GL executes the decision table actions. For any given decision table column, HP ALLBASE/4GL only executes the decision table actions if the results of all the decision table questions match the results you specify for that column. Refer to Additional Information.

To specify the sequence in which HP ALLBASE/4GL executes the actions, enter a number from 1 to 8 against each action. Use each number once only.

Enter a zero against an action if you don't want the system to execute that action.

Column Definition. Documentation field. Enter a description (up to 20 characters) to identify the purpose of the column.

Function Keys

- f1** **Header** Cancels the current screen and displays the decision table header screen.
- f2** **Questions** Cancels the current screen and displays the decision table questions screen.

- f3 **Actions** Cancels the current screen and displays the decision table actions screen.
- f4 **Generate D.Table** Generates the decision table.

Additional Information

When HP ALLBASE/4GL executes a decision table, it scans the columns of the decision table, starting with column number one. HP ALLBASE/4GL tests each column in turn until it detects a column that matches the actual outcomes of the decision table questions.

HP ALLBASE/4GL executes the actions specified for this decision table column in the sequence you have specified, and then terminates processing of the decision table. This means that if you have defined two or more decision table columns that may be satisfied by a particular combination of outcomes, only the lowest numbered column will be executed. This situation may occur if you have specified the *don't care* result for a question in one column, and also specified a more restrictive set of conditions by specifying Y or N for the same question in a different column.

Note that if one of the actions executed by a decision table is a process or a menu, the decision table is terminated immediately.

Logic Commands

This chapter describes the HP ALLBASE/4GL logic commands. The commands are listed in alphabetical order.

Naming Rules

In certain command descriptions, the descriptions refer to *screen field name* and *scratch-pad field name* rather than a *screen field reference* or a *scratch-pad field reference*. Wherever these are specified, the reference to the screen or scratch-pad field must be by name. Absolute or relative addressing is not allowed.

A screen field name takes the form:

S-dictionary_name.screen_name

where *dictionary_name* is the dictionary field specification name assigned to that screen field, and *screen_name* is the name of the screen.

Note



The screen name qualifier is optional with the CLEAR, FIELD, SHOW, TIE, and MOVE commands. If you do specify a screen name as a qualifier for a screen field name in these commands, it must be the current screen name.

A scratch pad field name takes the form:

P-field_name

Whenever this section refers to the following items, you can use appropriate substring or subscript references:

- variable or calculated items.

- constants.
- screen fields.
- scratch-pad fields.
- file record fields.
- work area fields.
- communication area fields.

Chapter 3 describes subscript or substring references.

Enter all user defined names exactly as they are defined. For example, a variable defined as *Cost_MtoD* would be referenced as *V-Cost_MtoD*.

Command Name Synonyms

The system administrator can assign a synonym for each logic command name. You can use either the synonym or the command name. For example, MESSAGE may be assigned the synonym MSG.

Each command can only have one synonym.

Disabled Commands

The system administrator can disable any logic command. If you try to use a disabled command, you will not be able to generate the function or process successfully.

If the system administrator disables a logic command, any existing generated logic that uses the command will still operate.

12-2 Logic Commands

Logic Command Windows

The function and process detail screens allow you to enter logic commands in a window on the screen. For some commands, HP ALLBASE/4GL presents a formatted window that shows the command options and guides you with the syntax of the command. These commands are:

- CHECK
- FILE
- IF
- IFLOOP
- LINK
- LINKLOOP
- MATH
- MATHLOOP
- MOVELOOP
- SELECT
- VALIDATE

HP ALLBASE/4GL presents the appropriate window for these commands when you enter the command name in the *Command* field on the function detail screen or the process detail screen.

In all cases, you can leave the *Command* field blank. If you do, HP ALLBASE/4GL presents an unformatted data entry window on the function or process detail screen. You can enter any logic command in this window, however you must use the correct syntax for the command.

HP ALLBASE/4GL converts all recognized command names and command arguments to upper case when you press the **Commit Data** function key on the function or process details screen.

Logic Command Length

The maximum length of an HP ALLBASE/4GL logic step is 400 characters.

The maximum length of the internal code generated by HP ALLBASE/4GL for each step is 255 characters. In some circumstances, the internal length of the step can be exceeded even though the logic step that you have entered does not exceed 400 characters. This usually occurs when your logic step contains long literal text strings. To shorten the internal length of the step, use a variable to define the long string.

Using Literals

Some logic commands allow you to use literals in the command. You must start and finish the literal with quote characters (").

Using Dates

HP ALLBASE/4GL stores and uses dates internally in the form YY/MM/DD, where YY is the year, MM is the month, and DD is the day. If you need to use a literal to express a date in a logic command, you must express the date in the format YY/MM/DD.

Master Titles

Some logic commands allow you to use master titles. When HP ALLBASE/4GL encounters a reference to a master title, it initially searches for an application title with the same name. If such an application title does not exist, HP ALLBASE/4GL uses the named master title.

You cannot use direct references to application titles in any logic command.

12-4 Logic Commands

BACKGRND Command

The BACKGRND command executes a process as a background task.

Formats

$$\text{BACKGRND } \left\{ \begin{array}{l} \textit{process_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

process_name

The name of an HP ALLBASE/4GL process.

data_ref

May be one of the following whose content is a valid HP ALLBASE/4GL process name:

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

Description

The BACKGRND command executes a process as a background task, releasing the terminal for data entry or inquiries while HP ALLBASE/4GL executes the process.

You must specify a valid process name or a data_ref whose content is a valid process name.

Once HP ALLBASE/4GL has started executing the background process, the background process cannot accept further data input from the user.

HP ALLBASE/4GL executes a background process by creating and then streaming a job file. The job file is then purged immediately. HP ALLBASE/4GL can execute more than one background process at any given

time and more than one background process can share the same name since each background process becomes a separate MPE/iX job.

During normal application execution, HP ALLBASE/4GL writes any messages issued by the background process to \$STDLIST. When trace mode is in use under application testing mode, messages from background processes are logged to a file *Mnnnnxxx* in the current group and account. In this file name, *nnnn* is a four digit number indicating the MPE/iX session number for the parent process (that is, the HP ALLBASE/4GL session that initiated the background process). The term *xxx* is a three digit number to distinguish background processes belonging to the same parent process.

The process terminates if any message issued by the background process is an abort type message. All other messages, including query messages, are treated as *message* type messages. The background process does not wait for a response to query messages.

Unless the background process aborts due to an error condition, it runs to completion even if the user terminates the HP ALLBASE/4GL session and logs out of MPE/iX.

HP4BG Variable

To run a background process, HP ALLBASE/4GL must create and then stream a suitable job file. To allow HP ALLBASE/4GL to create this file, the user must set the MPE/iX variable HP4BG containing the correct user, group and account passwords. The variable must contain a valid logon sequence, but must not contain the job keyword. For example, the HP4BG variable could be set as follows:

```
setvar HP4BG "jobname, username/password.account/password"
```

If you don't specify a job name in this variable, the name of the job defaults to the user's HP ALLBASE/4GL sign-on name.

12-6 Logic Commands

Background Process Information File

HP ALLBASE/4GL creates a disk file to pass information from the current foreground process to the background process. This file is called *HP4nnnnn*, where *nnnnn* is a five digit number. The information file is purged when the background process stream job starts.

Background Process Environment

The background process stream job sets a number of variables and JCWs to allow the background HP ALLBASE/4GL process to run. The variables and JCWs set by the background process stream job are:

- HP4BG
- HP4SPATH
- HP4DBMPATH
- HP4DATAPATH
- NLUSERLANG
- NLDATALANG

If the background process, or any external programs called by the background process require further variables or JCWs, these must be set in a logon UDC.

Limitations

The background process takes a snapshot copy of the HP ALLBASE/4GL environment and stores it in the *HP4nnnnn* file when it starts. The background process cannot use the copied contents of file record buffers or screen field buffers as these are cleared when the process starts. You cannot pass any further values to a background process, and the background process can only return values to the foreground process by writing them to a disk file.

Example

```
BACKGRND end_month_roll
```

This command executes the process *end_month_roll* as a background task.

CALC Command

The CALC command calculates the result of an arithmetic expression.

Format

CALC *operand* [#] *operator operand* [#] [*operator operand ...*]
= *result_data_ref* [*;command*]

Parameters

operand

May be one of the following:

- Whole number.
- Numeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- *TOTALS(n): where n = 1 to 16.
- *COUNT(n), or *CROSS(n): where n = 1 to 5.

operand#

Certain operands can be treated as absolute values. That is, their value is treated as a positive number. To specify an operand as absolute, enter the character # after the operand name. You cannot specify the result of the calculation as absolute.

You can specify the following operands as absolute:

- Numeric variables.
- Numeric constants.
- Screen field names.

12-8 Logic Commands

- Scratch-pad field names.
- File record field names.
- Work area field names.

operator

The operation to be performed on the two operands. It may be one of the following arithmetic operators:

+	Add
-	Subtract
X or x	Multiply
/	Divide
M or m	Modulus
e	Raised to the power of

result_data_ref

Where the result of the CALC is to be placed. It may be one of the following:

- Numeric variable.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- *TOTALS(n): where n = 1 to 16.
- *COUNT(n), *CROSS(n): where n = 1 to 5.

command

A command to be executed if an error occurs while the CALC command is being executed. You can use any of the following commands:

- ENTER
- EXIT

- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

HP ALLBASE/4GL executes the command in the following conditions:

- Machine overflow or underflow occurs.
- Division by zero occurs.
- The result is too large to fit in the result field.
- One or more of the operands contains non-numeric data, or an operand contains a minus sign and has an N edit code.

If you don't specify an optional command, and one of these error conditions occurs, HP ALLBASE/4GL displays a warning message and then executes the next command in the logic block.

Description

The CALC command calculates the result of an expression. The operands are processed in the order in which they are encountered. Algebraic rules of precedence do not apply.

The first two operands and their associated operator are mandatory. You can enter successive operator-operand pairs as often as desired.

HP ALLBASE/4GL uses full floating point arithmetic. The result of the CALC command is rounded off to suit the number of decimal places specified for *result_data_ref*. If the *result_data_ref* is a scratch-pad field, the number of decimal places used for the result of the calculation is the number of decimal places specified for the first operand of the CALC command.

Operands for the CALC command, and values computed by the CALC command, can contain a maximum of 14 significant digits. Overflow or

12-10 Logic Commands

truncation errors may occur if you attempt to use values with more than 14 significant digits.

Example

CALC V-rate/100 + 1 e V-years x F-principal.bank = F-interest.bank

This command calculates the compound interest of the value in *principal* in the file *bank* and places the result in *interest* in the same file. This command is equivalent to the formula $I = P(1 + r/100)^n$.

CALL Command

The CALL command provides dynamic loading of routines written in the C programming language.

Formats

$$\text{CALL } [*REFRESH]fn_name \text{ library_name}$$
$$\left[\begin{array}{l} parm_n \left\{ \begin{array}{l} *R \\ \{ datatype \} \\ *V \end{array} \right\} \dots \end{array} \right]$$

Parameters

*REFRESH

If you specify *REFRESH, ALLBASE/4GL saves the current screen settings and turns the screen control over to the C function. While the CALL command executes, messages from the C function are displayed on the screen. After the function completes, ALLBASE/4GL repaints the original screen. If you do not use the *REFRESH argument, C function information may overwrite available fields on the screen, and the original screen is not repainted.

The *REFRESH modifier has two effects. Before the library routine is called, the terminal is set to the terminal setting that exists before ALLBASE/4GL is invoked. On return from the library routine, the terminal is restored to the original terminal setting.

fn_name

The *fn_name* must be one of the following, whose content contains the name of the C function you want to call.

- alphanumeric constant
- screen field name
- scratch-pad field name

12-12 Logic Commands

- file record field reference
- work area field reference
- literal
- variable

library_name

The *library_name* must be one of the following, whose content contains the name of the HP-UX shared library where you want ALLBASE/4GL to search for the called function.

- alphanumeric constant
- screen field name
- scratch-pad field name
- file record field reference
- work area field reference
- variable
- literal

parm_n

These are the optional parameters that can be passed to and from the library function. The type of the variable passed to the C function is determined by the storage used for the ALLBASE/4GL item and must match those required by and used by the function. If you use a parameter, you *must* specify the ALLBASE/4GL storage type for each parameter used.

For each argument used in the CALL command two modifiers are specified. The first modifier must have the value *R or *V, which signifies one of the following:

- *R Argument by reference (any change to the value is reflected in ALLBASE/4GL).
- *V Argument by value (any change to the value is not reflected in ALLBASE/4GL).

The second modifier specifies the argument type, which may be the value *INT, *LONG, *CHAR, *FLOAT, or *DOUBLE. The arguments correspond to the native C language pointer to SHORT, pointer to INT, pointer to CHAR (string), pointer to FLOAT and pointer to DOUBLE. Any argument of type *CHAR (string) is terminated with a NULL character before being passed to the library routine to easily determine string lengths. The NULL character is ignored by ALLBASE/4GL on return.

data type

Data types are:

- *CHAR
- *INT
- *LONG
- *FLOAT
- *DOUBLE

The following table shows the ALLBASE/4GL storage types (as per the storage fields in the data dictionary) and the corresponding C storage types.

4GL Data Type	Storage Type	C Type
CHAR	C	Native C pointer to character data type. Arguments of this type are terminated with a NULL character before being passed to the library routine. The NULL character is ignored by ALLBASE/4GL on return.
INT	I	Address of short integer data type
LONG	L	Address of integer data type
FLOAT	F	Address of single-precision floating-point data type.
DOUBLE	D	Address of double-precision floating-point data type.

12-14 Logic Commands

Note

There is no corresponding C type for packed decimal, so any ALLBASE/4GL item with this storage type *cannot* be passed to a C function. It can, however, be MOVED to another 4GL item with a non-packed decimal storage type and used from there.

The values for these parameters are placed in ALLBASE/4GL storage elements. Any of the following types of ALLBASE/4GL storage elements can be used as parameters:

- Variables
- Screen field references
- Scratch-pad field references
- File record field references
- Work area field references
- File record buffers
- Work area buffers
- Communication area fields

The parameters can be passed by **reference (*R)**, which means that any change made to the parameter in the C function is reflected in ALLBASE/4GL. If you pass the parameter by **value (*V)**, the C function can modify the parameter, but it is not reflected in the parameter when returning from the function.

Parameters are **always** passed to the C function by address. This means that any C functions called by ALLBASE/4GL must declare parameters as pointers. See the type declarations in Example 3 in the next section for clarification.

Description

CALL allows developers to use existing code libraries and develop applications faster. The CALL command is similar to the EXTERNAL command except that it executes routines in the *same* process as ALLBASE/4GL whereas the EXTERNAL command starts up one or more completed programs as separate processes.

The CALL command enables the routine to become an active part of the ALLBASE/4GL process, providing maximum flexibility and high performance. With this enhancement, ALLBASE/4GL becomes truly extensible. Developers can add their own functionality to ALLBASE/4GL, build up libraries of object code, and create a development system geared toward maximum productivity.

The CALL command executes a function stored in a shared library. The function must be written in the C language and the variables must match *exactly* the variables required by the C function. Arguments can be passed to the library routine by value (*V) or by reference (*R).

Warning

It is the developer's responsibility to ensure that the type and number of parameters agree between ALLBASE/4GL and the C function. If they don't agree, this would result in an error that would not be detected by ALLBASE/4GL. It is also the developer's responsibility to ensure that the C function code is not faulty. Since the C function is executing in the same process as ALLBASE/4GL, anything that causes the C function to abort also causes ALLBASE/4GL to abort. For instance, a segmentation violation due to faulty C code causes ALLBASE/4GL to abort.

Example 1

```
CALL V-functionx V-my_lib V-my_variable *V *INT
```

This example passes the address of the ALLBASE/4GL variable *my_variable* to the C function. Any changes to the variable are not reflected in the variable when control returns to ALLBASE/4GL.

This example assumes that no information needs to be written to the screen and therefore, the *REFRESH option is not used.

Example 2

```
CALL "functionx" V-my_lib V-my_variable *R *INT
```

This example also passes the address of the ALLBASE/4GL variable *my_variable* so that the C function can actually change it. However, any changes made to the variable in the C function are also reflected in the variable when control returns to ALLBASE/4GL.

This example assumes that no information needs to be written to the screen and therefore, the *REFRESH option is not used.

Example 3

```
MOVE "2" V-var1
MOVE "This is a test string" V-var2
MOVE "mylib.sl" V-lib_name
MOVE "my_func" V-func_name
CALL *REFRESH V-func_name V-lib_name V-var1 *R *INT V-var2 *V *CHAR
```

This example shows the relationship between the ALLBASE/4GL command and the associated function that it calls. On returning to ALLBASE/4GL the variable *V-var1* is incremented by 1. The standard I/O function `printf` is used in this example.

The function *my_func* could be as follows:

```
#include <stdio.h>

int my_func(number, string)
int *number;
char *string;
{
    /* type L in ALLBASE/4GL */
    /* type C in ALLBASE/4GL */

    *number = *number + 1;
    printf("The string is %s \n Press return to continue.",string);
    getchchar ();
    return 0;
}
```

Note



The function return value is put into the communication area *ERROR. The function return type must be of type `int`.

12-18 Logic Commands

CHECK Command

The CHECK command matches the contents of a field against a validation table or a series of contiguous fields in an indexed (KSAM) file record. The command returns the matched position in the table or file record to the *PASS communication area field.

You can only use the CHECK command with KSAM data files. You cannot use the CHECK command with serial data files, HP TurboIMAGE/iX data sets, HP ALLBASE/SQL tables, or select lists.

Formats

$$\text{CHECK } data_ref \left\{ \begin{array}{l} *TABLE=table \\ file_ref \ [*INDEX=index_name] \\ *KEY=key [entries_field_name] \end{array} \right\}$$

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	CHECK			
Check: Data Name			Against:				
*TABLE=		(Validation Table Name)	Or:				
File[.record]							
*INDEX=							
KEY=							
If Checking Against a File:		Field Name of Entry Count					
Command	Function	Process	Generate	System	Commit	Help	Previous
Help	Header	Detail	Function	Keys	Data		Menu

Parameters

data_ref

The value to be checked against the table or field in the file record. It can be one of the following:

- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.

- Work area field reference.

table

If checking against a table, this must be the name of a validation table.

file_ref

If checking against a file record which contains a series of contiguous fields, this should be the name of the file, optionally followed by a record name.

If you don't specify a record name, HP ALLBASE/4GL uses the default record for the file.

index_name

If checking against a file, this parameter allows you to specify the name of the index to be used to access the file. The index name must be the name of a field specification defined as a key field on the default record layout for the files. You can also specify the index by entering the number of an index for the file in the **INDEX=* field.

If you don't specify an index name, HP ALLBASE/4GL accesses the file according to the index specified by the current value of **INDEXNO*.

key

If checking against a file, this field is mandatory. It can be one of the following:

- Literal.
- Alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Non-numeric communication area field.

entries_field_name

This field is optional and, if specified, should be a file record field reference.

12-20 Logic Commands

Description

The CHECK command matches the contents of *data_ref* against either:

- A validation table, or
- A series of contiguous fields in a specified file record.

If a match occurs, the CHECK command places the matching position number into the *PASS communication area field. Only the first matching position is reported.

If no match occurs, the CHECK command places a zero in the *PASS field.

You can then use the content of *PASS as desired. For example, it could be used to execute a specific step within a SELECT command.

If checking against a table, **TABLE=table* specifies the validation table to be searched. This table must be defined on the dictionary validation table definition screen.

If checking against a series of fields in an indexed (KSAM) file record, **KEY=key* defines the key of the file record to be retrieved before performing the search.

If you don't use the **INDEX = index_name* argument, HP ALLBASE/4GL uses the current value of *INDEXNO to determine the key number for the file search. If the index allows duplicate key values, only the first occurrence is used. If the required record can't be accessed, the CHECK command places a zero in *PASS (no match occurred).

When checking against fields in a file record, HP ALLBASE/4GL expects the record to contain:

- A field containing a number equal to the number of fields to be searched.
 - immediately followed by:
- A series of contiguous fields, all of the same length, containing the data to be searched.

Unless you specify the optional parameter *entries_field_name*, these fields must occur at the start of the record and they must always occur together in the relationship defined above.

The optional parameter *entries_field_name* specifies the name of a field in the file record defined by *KEY. This field must contain a number equal to the number of fields to be searched. The fields to be searched must immediately follow this field, and must be contiguous.

Example 1

```
CHECK * trans_code *KEY=*S01
```

This command accesses the file *trans_code* using the contents of screen field number 1 as the key. When HP ALLBASE/4GL reads the specified record, it assumes that the first field contains a number equal to the number of fields to be searched. HP ALLBASE/4GL then searches those fields for a match against the contents of the current screen field. If a match occurs, the CHECK command places the number of the matching field into *PASS. If no match occurs, the CHECK command places the digit zero into *PASS.

Example 2

```
CHECK V-state *TABLE=valid_states
```

This command checks the contents of variable *state* against the validation table *valid_states*.

CLEAR Command

The CLEAR command clears all or part of a screen or the scratch pad.

Formats

$$\text{CLEAR} \left\{ \begin{array}{l} *S \text{ [from_data_ref [to_data_ref]]} \\ *P \text{ [from_data_ref [to_data_ref]]} \\ *MAP \text{ map_from_data_ref [map_to_data_ref]} \end{array} \right\}$$

Parameters

from_data_ref, to_data_ref

These fields are optional. They define the extent to which the screen or scratch pad is cleared. They can be one of the following:

- Whole number.
- Numeric constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.
- File record field reference.
- Work area field reference.

In the CLEAR command, you can specify a screen field by the field name, and optionally omit the screen name.

map_from_data_ref, map_to_data_ref

These fields specify the actual name(s) of the range of fields to be cleared. They can be one of the following:

- Screen field name.
- Scratch-pad field name.

Description

The CLEAR command clears a single field, a range of fields, or all fields on either the current screen or the scratch pad. When used with a screen, the CLEAR command clears both the screen display and the screen field buffers for the field or fields specified in the command parameters.

Two forms of the command are available. The first form of the command is:

$$\text{CLEAR } \left\{ \begin{array}{l} *S \\ *P \end{array} \right\} [from_data_ref [to_data_ref]]$$

This form of the CLEAR command identifies the extent of the fields to be cleared by specifying either the field numbers or data references which contain the numbers of the fields to be cleared.

*S specifies that the clearing occurs on the current screen.}

*P specifies that the clearing occurs in the scratch pad.}

If you don't specify any data references, HP ALLBASE/4GL clears the whole screen or set of scratch pads.

If you only specify one *data_ref*, HP ALLBASE/4GL only clears one field.

If you specify both a *from_data_ref* and a *to_data_ref*, HP ALLBASE/4GL clears that range of fields. The range of fields cleared includes the *from_data_ref* and *to_data_ref* fields.

The *from* and *to* values must either be numbers, or data references which contain the field numbers to be cleared.

The second form of the CLEAR command is:

$$\text{CLEAR } *MAP \text{ map_from_data_ref [map_to_data_ref]}$$

This form of the CLEAR command identifies the extent of the fields to be cleared by specifying the actual field names. These names are then mapped into their corresponding field positions.

The *map_to_data_ref* or *map_from_data_ref* is a required field. That is, at least one field must be cleared using this form of the command.

If you specify both *map_from_data_ref* and *map_to_data_ref*, HP ALLBASE/4GL clears those fields and all fields between them.

12-24 Logic Commands

If the fields you want to clear are scratch-pad fields, they must have names assigned to them in the dictionary scratch-pad name declaration screen.

Example 1

```
CLEAR *S
```

This command clears all fields on the current screen.

Example 2

```
CLEAR *P 2 V-last
```

This command clears fields in the scratch pad from field number 2 to the field whose number is defined by the current value of the variable *last*.

Example 3

```
CLEAR *MAP P-Account_name
```

This command clears the scratch-pad field *Account_name*.

Using the *MAP option means the actual field *Account_name* is cleared and not the field identified by the contents of *Account_name*.

Example 4

```
CLEAR *MAP S-Acct_name S-Closing_bal
```

This command clears the screen fields *Acct_name* and *Closing_bal*, and all fields between them on the current screen.

DATE Command

The DATE command performs various calculations on dates.

The DATE command exists in four formats as shown below. The first format calculates the difference in days between two given dates. The second and third formats calculate the date obtained by adding or subtracting a number of days to a given date. The fourth format calculates the day of the week corresponding to a given date.

Formats

DATE *date_field_1* *DIFF *date_field_2* = *no_days*

DATE *date_field_1* $\left\{ \begin{array}{l} + \text{ days} \\ - \text{ days} \end{array} \right\}$ = *result_date* [; *command*]

DATE *date_field_1* *DAY = *day_field*

Parameters

date_field

A field containing a valid date. It can be any of the following:

- Date in quotes in the form “YY/MM/DD”, where YY is the year, MM is the month, and DD is the day.
- Variable or calculated item.
- Alphanumeric constant.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- *DATE.

If this field doesn't contain a valid date, HP ALLBASE/4GL displays an error message and processing then continues.

12-26 Logic Commands

no_days

The first format of the DATE command calculates the difference in days between two dates and places the result in the field *no_days*. This field can be any of the following:

- Variable.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Numeric communication area field.

days

The second and third formats for the DATE command calculate a date by adding or subtracting a number of days in the field *days* from the date in the field *date-field*. The field *days* can be one of the following:

- Whole number.
- Numeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Numeric communication area field.

result_date

The field where the result date calculated in the second and third formats of the DATE command is placed. This field can be any of the following:

- Variable with edit code D.
- Screen field reference.
- Scratch-pad field reference.

- File record field reference.
- Work area field reference.

command

An optional command executed if the century of the date represented by the value in *result_date* differs from the century of the date in *date_field*. The command can be one of the following:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

If you don't specify an optional command and a century overflow error occurs, the application aborts.

day_field

The fourth format of the DATE command calculates the day of the week corresponding to the date in *date_field*. This form of the command returns a single digit between 1 and 7, representing the days from Sunday through to Saturday respectively. The *day_field* can be any of the following:

- Variable.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Numeric communication area field.

12-28 Logic Commands

Example 1

```
DATE *DATE *DIFF * = V- days_elapsed}
```

This command compares the current date with the date in the current screen field. The command writes the difference between the two dates into the variable *days_elapsed*.

Example 2

```
DATE V-Due_date *DAY = *P01
```

This command determines the day of the week corresponding to the date in the variable *Due_date*. The command writes a single digit from 1 through 7 to scratch pad field number one. The output number represents the day of the week, with 1 representing Sunday and 7 representing Saturday.

DECISION Command

The DECISION command executes an HP ALLBASE/4GL decision table.

Formats

$$\text{DECISION } \left\{ \begin{array}{l} \textit{decision_table_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

decision_table_name

The name of a decision table.

data_ref

Can be one of the following containing the name of the decision table to be executed:

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

Example

```
DECISION V-dec_table
```

This command executes the decision table whose name is contained in the variable *dec_table*.

DEFINE Command

The DEFINE command defines an abbreviation for an expression (defines a macro) for subsequent referencing in the current logic block.

Format

DEFINE *%definition_name% expression*

Parameters

definition_name

A string, consisting of 1 to 4 characters, which identifies the expression.

expression

May be any series of characters that are to be inserted into a logic command.

Description

The DEFINE command defines a macro. That is, it equates a definition name to an expression so you can use the abbreviated name in place of the expression in subsequent commands in the same logic block.

You must always surround the abbreviation by percent signs (%) whenever you use it in a command. When HP ALLBASE/4GL encounters the construct *%xxx%* (where *xxx* is the abbreviation), it replaces the abbreviation with the defined expression. HP ALLBASE/4GL replaces the abbreviation with the expression before it performs any syntax checking or generation.

Using DEFINE macros can simplify some changes to HP ALLBASE/4GL logic commands. For example, if you equate a file name to a DEFINE macro, any changes to the file name simply require a change to one command, assuming all subsequent commands in the same logic block use the abbreviated form of the file name.

Limitations

DEFINE commands must be at the beginning of the logic block and precede all other commands in the logic block.

The DEFINE command is only valid within one logic block. If you want to use the same abbreviation in several logic blocks, you must define the abbreviation in each logic block.

The DEFINE command cannot be used to define abbreviations for logic command names or command arguments.

Example 1

```
DEFINE %F% TRANS.UNITCOST
DEFINE %kf% V-Current_Key
FILE *READ %F% *KEY= %kf%
MATH F-YEARLY_CST.%F% / 12 = V-EST_MTH_COST
```

These commands equate the file reference *TRANS.UNITCOST* to F, and the key *V-Current_Key* to kf. You can subsequently reference either by entering %F% in place of the file name and %kf% in place of the file's key specification.

DISPLAY Command

The DISPLAY command displays data on the current screen at a nominated line within its defined scroll area.

Formats

$$\text{DISPLAY} \left\{ \begin{array}{l} *RESET = \textit{line_number} \\ *RESET = S \end{array} \right\} [n] \textit{data_ref} [n] [\textit{data_ref}] \dots$$

Parameters

line_number

The absolute screen line number where the display line is to be placed. It may be one of the following:

- Whole number.
- Numeric constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.
- File record field reference.
- Work area field reference.

data_ref

The item or items to be displayed, each of which may be one of the following:

- A number (representing a number of spaces).
- A literal.
- A master title name.
- A numeric or alphanumeric constant.
- A variable or calculated item.
- A screen field reference.

- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- A communication area field.

n

A number representing the number of spaces to be inserted between the items when they are displayed.

Description

The DISPLAY command displays information in the scroll area of the current screen on a specified line.

The data references define the items to be displayed. You must specify at least one item.

HP ALLBASE/4GL displays each item next to the preceding item. To insert blank spaces into the display line between items, enter a digit that corresponds to the number of spaces required between each data reference. Refer to example 1.

You can express the screen line where the display occurs as either the letter S, or *screen_line_number* as defined above. The letter S indicates that display starts on the first line of the scroll area. The first line of the scroll area is either the top line of the scroll area or the bottom line of the scroll area, depending on the scrolling direction defined for the screen.

If you specify a screen line number, it must be within the scroll area defined for the current screen. If the *screen_line_number* is outside the scroll area, the display occurs on the first scroll line automatically.

With the *RESET = S format of the command, HP ALLBASE/4GL clears the entire scroll area and displays the item(s) on the first line of the scroll area.

With the *RESET = *screen_line_number* format of the command, HP ALLBASE/4GL clears the specified line and all lines below this line in the scroll area for a downwards scroll area. For an upwards scroll area, the system clears the specified line and all lines above it in the scroll area. HP

12-34 Logic Commands

ALLBASE/4GL then displays the item(s) on the specified screen line in the scroll area.

If the resulting display line is too long to fit on the screen, HP ALLBASE/4GL truncates the line to fit.

Example 1

```
DISPLAY *RESET=S F-emp_no.emp 2 F-name.emp 2 "Added"
```

This command displays the fields *emp_no* and *name* from file *emp* on the first line of the scroll area. Two blank spaces are left between the fields. The word *added* is also displayed after *name*. HP ALLBASE/4GL clears the scroll area before it displays this information.

Example 2

```
DISPLAY *RESET=V-line *S01 1 *S02 1 *S05 1 *S06[1,3]
```

This command displays the screen field numbers 1, 2, 5 and the first 3 characters of field 6 on the screen line defined by the contents of the variable *line*. One space is left between each field.

Example 3

```
DISPLAY *RESET=S ""
```

This command clears the scroll area on the current screen.

DM Command

DM is used as a prefix to logic commands specific to individual data managers. The following pages describe the data manager commands.

Currently, HP ALLBASE/4GL only offers data manager specific commands for the HP TurboIMAGE/iX data manager. All HP TurboIMAGE/iX specific commands begin with *DM IMAGE*.

DM IMAGE *CLOSE Command

The DM IMAGE *CLOSE command closes one or more HP TurboIMAGE/iX databases.

Format

DM IMAGE *CLOSE *:D-database ...* [*;
command*]

Parameters

database

The name of the HP TurboIMAGE/iX database(s) to be closed. The name must be an HP ALLBASE/4GL name for the database, as defined in the administrator application.

command

A command to be executed if an error occurs while the DM IMAGE *CLOSE command is being executed. You can use any of the following commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES

12-36 Logic Commands

- TOP
- VISIT
- ZIP

Description

Full HP TurboIMAGE/iX database closure occurs implicitly only at the end of the HP ALLBASE/4GL session. To explicitly close a complete database during a session, you must use the DM IMAGE *CLOSE command.

When a database is closed, all database, data set, and record (data entry) level locks for the specified database are released.

The DM IMAGE *CLOSE command maps to the HP TurboIMAGE/iX DBCLOSE library procedure (mode 1).

Example

```
DM IMAGE *CLOSE :D-stock_ctrl
```

This command closes the database *stock_ctrl*, and releases any locks currently held on the database.

DM IMAGE *LOCK Command

The DM IMAGE *LOCK command locks an entire HP TurboIMAGE/iX database, or specified data sets or records within an HP TurboIMAGE/iX database.

Formats

```
DM IMAGE *LOCK :D-database [descriptor] ... [;command]
```

Parameters

database

The name of the HP TurboIMAGE/iX database that is to be fully or partially locked. The name must be an HP ALLBASE/4GL name for the database, as defined in the administrator application.

descriptor

The descriptors identify the parts of the database to be locked. This field is optional. If you do not supply a descriptor, the entire database is locked.

Each descriptor takes the following format:

$$\text{descriptor} = \left\{ \begin{array}{l} :R\text{-fileref} \\ :F\text{-field.fileref} \leq \text{value} \\ :F\text{-field.fileref} \geq \text{value} \\ :F\text{-field.fileref} = \text{value} \end{array} \right\}$$

field

The name of a field specification.

fileref

Specifies the HP ALLBASE/4GL name of the application file on which the operation is to be performed. It can be expressed in either of the following ways:

■ *file_id*

- or

■ *file_id.record_name*

If you don't specify *.record_name* for the file, HP ALLBASE/4GL uses the default record of the file.

value

The value can be one of the following:

- Numeric or alphanumeric literal.
- Numeric or alphanumeric constant.

12-38 Logic Commands

- Master title.
- Scratch-pad field reference.
- Screen field reference.
- File record field reference.
- Variable or calculated item.
- Work area field reference.
- Communication area field.

You must precede a reference to an HP ALLBASE/4GL item with a colon (:). That is, you must express the reference in the format:

:V-due_date

Literal text must not be preceded by a colon.

command

A command to be executed if an error occurs while the DM IMAGE *LOCK command is being executed. You can use any of the following commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

Description

This command allows you to lock all or parts of an HP TurboIMAGE/iX database. The lock is maintained until one of the following occurs:

- A DM IMAGE *UNLOCK command is executed for the locked database.
- A DM IMAGE *CLOSE command is executed for the locked database.
- A FILE *UNLOCK command is executed for any part of the database (if any locked data sets have an access mode of MODE *WRITE or MODE *UNLOCK).
- The current process ends.

Locking Strategy

In general, you should establish locks that cover all records that you intend to modify, or all records which must not change during the transaction. However, you must also consider whether other users would be likely to access the same database or data set at the same time. If you lock a whole database, no other user can alter any part of the database. A data set lock ensures that all records within the data set will not be accessed by another user.

You do not need to lock any parts of a database accessed when the DM IMAGE *MODE is set to *MODEXCL or *READEXCL, because these modes guarantee the user exclusive use of a database.

When adding or deleting entries from a detail data set, record level (data entry level) locking is sufficient. You do not need to have locks covering additions or deletions that occur in any associated automatic masters.

To delete or insert master data set records, use data set level locking. For updating master data set records, record level locks are adequate.

If you wish to establish multiple record level locks for a single data set, each lock descriptor must use the same field (or data item) to identify the record. HP TurboIMAGE/iX does not allow more than one field per data set to be used for locking purposes at the same time. You may also use the DM IMAGE *LOCK command to lock the whole data set instead.

For more guidance, use the locking guidelines provided in the *HP TurboIMAGE/iX Database Management System Reference Manual*.

12-40 Logic Commands

The DM IMAGE *LOCK command maps to the HP TurboIMAGE/iX DBLOCK library procedure. If the *LOCKWAIT communication area field contains the value *-1*, the lock is unconditional, and mode 1, 3 or 5 of the DBLOCK library procedure is used, depending upon the descriptor provided. Otherwise, mode 2, 4 or 6 of the DBLOCK library procedure is used, depending upon the descriptor provided.

***LOCKWAIT Communication Area Field**

If any of the specified records, data sets, or databases are already locked, you may use the *LOCKWAIT communication area field to make the current process or function wait for the lock to be released.

The actions taken by HP ALLBASE/4GL for different values in *LOCKWAIT are summarized below.

Value	Action
+n	HP ALLBASE/4GL attempts to lock the file every second for <i>n</i> seconds. If the file cannot be locked after <i>n</i> seconds, HP ALLBASE/4GL returns an error number 19107.
0	HP ALLBASE/4GL attempts to lock the record every second for 5 seconds. If the record cannot be locked after 5 seconds, HP ALLBASE/4GL returns an error number 19107.
-1	HP ALLBASE/4GL waits indefinitely to access locked records.

Implicit Locking

If you do not use the DM IMAGE *LOCK command, HP ALLBASE/4GL will use implicit locking when data sets are accessed. The type of implicit locking is determined by the MODE command.

For details about implicit locking, refer to the FILE Command (HP TurboIMAGE/iX) and the MODE command.

Example 1

```
DM IMAGE *LOCK :D-stock_ctrl
```

This command locks all records within the *stock_ctrl* database.

Example 2

DM IMAGE *LOCK :D-stock_ctrl :F-supp_no.supplier = V-last_supplier}

This command locks all records in the *supplier* data set (which is part of the *stock_control* database), where the supplier number (*supp_no*) equals the value of the *last_supplier* variable.

DM IMAGE *MODE Command

The DM IMAGE *MODE command specifies the access mode used when opening one or more HP TurboIMAGE/iX databases. The command does *not* actually open any data sets.

Formats

$$\text{DM IMAGE *MODE} \left\{ \begin{array}{l} *MODEXCL \\ *MODLOCK \\ *MODREAD \\ *READEXCL \\ *READLOCK \\ *READMOD \\ *READSHAR \\ *UPDSHAR \end{array} \right\} :D\text{-database} \dots [; \text{command}]$$

Parameters

database

The name of the HP TurboIMAGE/iX database(s) for which an access mode is set. The name must be the HP ALLBASE/4GL name for the database, as defined in the administrator application.

command

A command to be executed if an error occurs while the DM IMAGE *MODE command is being executed. You can use any of the following commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES

- TOP
- VISIT
- ZIP

Description

The DM IMAGE *MODE command specifies the access mode that HP ALLBASE/4GL uses to open the named HP TurboIMAGE/iX database(s) in the current process.

The default database access mode is *MODE *READLOCK. If no database access mode has been set for a database, it is opened in the default mode.

Once the access mode is set for any database, the access will be used whenever that database is opened. You cannot reset the access mode for a database while the database is open.

Each mode corresponds to an HP TurboIMAGE/iX access mode. These access modes are briefly defined in the table below. This table outlines, for each mode, the equivalent HP TurboIMAGE/iX mode, the modes that can be used by concurrent users, and the actions that concurrent users may perform. If a user attempts to access a database in an access mode that cannot be used with a mode already specified for the database, an error occurs.

For more information about HP TurboIMAGE/iX access modes, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

12-44 Logic Commands

Mode	Mapping to HP Turbo-IMAGE/XL Access Modes	Read	Modify	Concurrent User Modes	Concurrent Actions
*MODLOCK	1	Y	Y	* MODLOCK and *READLOCK users	Modify with locks
*UPDSHAR	2	Y	Update only	*READMOD and *UPDSHAR users	Update only
*MODEXCL	3	Y	Y	None	None
*MODREAD	4	Y	Y	*READMOD users	Read
*READLOCK	5	Y	N	*MODLOCK and *READLOCK users	Modify with locks
*READMOD	6	Y	N	*READMOD, *READSHAR, and *UPDSHAR users. One *MODREAD user	Modify
*READEXCL	7	Y	N	None	None
*READSHAR	8	Y	N	*READMOD and *READSHAR users	Read

***MODLOCK.** The user who accesses the database in this mode can perform all database operations, but must lock records before altering them. It is also wise to lock a record before reading it.

***UPDSHAR.** The user is able to read and update existing records in this mode, but is unable to add or delete records. Concurrent users may also update or read the database.

***MODEXCL.** The database can only be opened in this mode if no other users are accessing any part of the database. The user can perform all database operations.

***MODREAD.** This is a semi-exclusive mode. A user may access the database and modify records, and other users may read records (in READMOD mode). File locking before modifying a record is recommended but not necessary.

***READLOCK.** The user who accesses a file in this mode may only read records. However, concurrent users may lock, and then modify, records.

***READMOD.** This is similar to READLOCK mode, but concurrent users need not lock records before modifying them.

***READEXCL.** The database is opened for exclusive read only access. No other users can access any part of the database.

***READSHAR.** The database is opened for users for read access only. The database values can be guaranteed to be unchanging.

Choosing an Access Mode

To select an access mode, use the least capability that will accomplish the task.

If you only need to locate and read or report on information, choose mode *READLOCK, *READMOD, *READEXCL or *READSHAR.

If you need to update data but not add or delete entries, and no other parallel processes need to add or delete at the same time, use mode *UPDSHAR.

To perform all database operations, choose mode *MODLOCK, *MODEXCL, or *MODREAD.

Using the MODE command with DM IMAGE *MODE

The DM IMAGE *MODE command works in conjunction with the MODE command to specify the access restrictions for each HP TurboIMAGE/iX database and data set. The DM IMAGE *MODE command sets the overall access authority for the database. The MODE command specifies the access restrictions for each data set. For example, you can allow write access to the

12-46 Logic Commands

database, using the DM IMAGE *MODE command, but allow read access only to a specific data set within the database, using the MODE command.

For each file operation, the DM IMAGE *MODE command and the MODE command must both be sufficient to do the required tasks. However, the MODE command must not provide more capability than the DM IMAGE *MODE command. For example, DM IMAGE *MODE *MODLOCK and MODE *READ can be combined because reading is a subset of DM IMAGE *MODE *MODLOCK capability, but DM IMAGE *MODE *READLOCK and MODE *WRITE cannot be used together, because the MODE command specifies more capability than the DM IMAGE *MODE command has given to the database. If the MODE command specifies more capability than the DM IMAGE *MODE command, an error will occur at run-time if an action is attempted that requires more capability than the DM IMAGE *MODE command provides.

Example

```
3  DM IMAGE *MODE *UPDSHAR :D-suppdb
4  MODE *WRITE supplier.
.
6  FILE *NEXT supplier
.
```

This command sets the database access mode to *UPDSHAR, sets the data set access mode to *WRITE, and then accesses the *supplier* data set. The database is opened in *UPDSHAR mode, and the user is able to update existing records in the *supplier* data set.

DM IMAGE *UNLOCK Command

The DM IMAGE *UNLOCK command releases all locks held on one or more HP TurboIMAGE/iX databases.

Format

DM IMAGE *UNLOCK *:D-database ...* [*; command*]

Parameters

database

The name of the HP TurboIMAGE/iX database(s) to be unlocked. The name must be an HP ALLBASE/4GL name for the database, as defined in the administrator application.

command

A command to be executed if an error occurs while the DM IMAGE *UNLOCK command is being executed. You can use any of the following commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

Description

The DM IMAGE *UNLOCK command allows you to release all locks held on one or more HP TurboIMAGE/iX databases. You must release all locks held on a database; HP TurboIMAGE/iX does not allow you to release only some of the locks held on a database.

When a DM IMAGE *UNLOCK command is executed, all implicit file locks on the specified databases are also unlocked.

Example

```
DM IMAGE *UNLOCK :D-stock_ctrl
```

This command releases all locks held on the *stock_ctrl* database and any locks held on part of the *stock_ctrl* database.

ENTER Command

The ENTER command specifies the number of the next command step to be executed in the current logic block.

Formats

$$\text{ENTER} \left\{ \begin{array}{l} \textit{step_number} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

step_number

An integer number in the range 1 to 99.

data_ref

May be one of the following, whose content is a number in the range 1 to 99:

- Numeric constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.
- File record field reference.
- Work area field reference.

Description

The ENTER command is used to unconditionally GOTO the specified command step within the current logic block.

If the content of *data_ref* does not refer to a valid step in the current logic block at run time, the logic block exits. If *data_ref* contains a non-integer value, it is truncated to the next lowest whole number to obtain the number of the next step to be executed.

Example 1

12-50 Logic Commands

ENTER 12

This command causes step 12 to be executed next.

Example 2

ENTER V-MONTH

This command causes HP ALLBASE/4GL to execute the step whose number is contained in the variable MONTH.

EXIT Command

The EXIT command indicates the end of processing in the current logic block.

Format

EXIT [*COMMIT]

Description

This command indicates the end of the logic block. The command may appear more than once in a logic block.

When HP ALLBASE/4GL encounters an EXIT command in a function, processing resumes at the command immediately after the VISIT command in the logic block if the function was called from within a logic block. If the function was called from a data screen or menu, control returns to the data screen or menu.

When the system encounters an EXIT command in a process, processing resumes with the latest menu displayed. If no menu has been displayed, the application is terminated. A process initiated via the BACKGRND command is terminated.

In a function, the optional *COMMIT argument commits the current screen in the same way as the I{-}COMMIT internal routine from a function key. If the commit is successful, the function exits normally. You can only use the *COMMIT argument in the following circumstances:

- In after functions for input fields on data screens.
- In prior functions for display-only fields on data screens,
- In a function called from a function key on a data screen.

The EXIT *COMMIT command overrides the effect of the *ENDFIELD switch. If you use the EXIT *COMMIT command, HP ALLBASE/4GL commits the current field even if *ENDFIELD has been set *on*.

12-52 Logic Commands

EXTERNAL Command

The EXTERNAL command transfers program execution to an external program written in a language other than HP ALLBASE/4GL.

Parameters can be passed to and from the external program.

Formats

$$\text{EXTERNAL} \left\{ \begin{array}{l} [*REFRESH] \left\{ \begin{array}{l} \text{program_name} \\ \text{data_ref} \end{array} \right\} [\text{parm}_1 \dots \text{parm}_n] \\ \left\{ \begin{array}{l} \text{program_name} \\ \text{data_ref} \end{array} \right\} *COMMS \end{array} \right\}$$

Parameters

program_name

The name of the program to be executed.

data_ref

Can be one of the following containing the name of the external program.

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

parameter_1, ... parameter_n

The names of values to be passed to the program. They can be any of the following:

- Literal.
- Master title name.
- Numeric or alphanumeric constant.
- Variable or calculated item.

- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area field.
- File record buffer.
- Work area buffer.

Description

The EXTERNAL command executes an external program which is written in a language other than HP ALLBASE/4GL. The language must be one supported by the MPE/iX system.

HP ALLBASE/4GL searches for the external program in the current group, and then in the groups specified by the MPE/iX system variable HPPATH. After the EXTERNAL command calls the external program, HP ALLBASE/4GL waits until the program has finished. HP ALLBASE/4GL then resumes processing at the command following the EXTERNAL command.

The EXTERNAL command with parameters passes the values of the named parameters to the program. Parameters are passed to the external program as an information string. The external program is called as:

```
PROGRAM_NAME; INFO="parm_1 ... parm_n"
```

The program must be written to expect the exact number and type of parameters specified. The total length of the information string cannot exceed 256 bytes.

If the parameters for the external command contain any of the following key words, they are passed directly to the external program with the same meaning as in the MPE/iX RUN command:

- ;PARAM=
- ;STDLIST=
- ;STDIN=

12-54 Logic Commands

These key words are passed to the external program after the parameters that make up the INFO string. A parameter can contain the complete statement for the key word expression in the form “;*key_word*=*value*”, or can contain only the key word itself in the form “;*key_word*=”. If a parameter only contains a key word, the next parameter in the list is taken as the argument. For example, an EXTERNAL command could contain the following parameters:

```
EXTERNAL myprog parm_1 ";PARAM=3" ... parm_n ";STDLIST=" V-list_file
```

If the current value of V-list_file is “logfile”, this command is passed to MPE/iX as:

```
RUN myprog; INFO="parm_1 ... parm_n";PARAM=3;STDLIST=logfile
```

The program can return modified values to the parameters by writing them to a message file *HP4EXTP*. The file is a variable length record binary message file.

The external program must return parameters in the following format.

```
EXTERNAL␣parm_no␣value_length␣value␣  
[parm_no␣value_length␣value] ...
```

The external program must insert a single space character (indicated by the symbol ␣) between each term in this expression. Values can be returned to any of the following HP ALLBASE/4GL items:

- Variables.
- Screen field references.
- Scratch-pad field references.
- File record field references.
- Work area field references.
- File record buffers.
- Work area buffers.
- Communication area fields.

The returned values cannot contain null characters.

You must use parameters if you want the external program to return values.

Values cannot be returned for constants, calculated items, literals or read-only communication area fields.

The optional *REFRESH argument controls the terminal behavior when the external program is called, and on return to HP ALLBASE/4GL. If you specify the *REFRESH argument, HP ALLBASE/4GL returns the terminal configuration to the state that existed when HP ALLBASE/4GL was first started. When control returns to HP ALLBASE/4GL, the system restores the terminal to its HP ALLBASE/4GL configuration, and redraws the screen.

If you specify the *REFRESH argument, the system displays a message when the EXTERNAL command is executed.

If you use the EXTERNAL command without the *REFRESH argument, the terminal remains in HP ALLBASE/4GL mode.

The EXTERNAL command with the *COMMS argument passes the contents of the communication area field *PASS to the external program as parameter number 3 (that is, the fourth parameter in the list). If you use the *COMMS format of the command, HP ALLBASE/4GL does not change the terminal configurations when the external program is called. However, the system does restore the terminal to HP ALLBASE/4GL mode and also redraws the screen on return to HP ALLBASE/4GL. With this format of the EXTERNAL command, the external program cannot return values to HP ALLBASE/4GL.

Example 1

```
EXTERNAL factorial "5" V-result
```

This example shows the use of a Pascal program called by the HP ALLBASE/4GL EXTERNAL command. The EXTERNAL command calls the external program *factorial*. The external program receives a value (in this case 5) from HP ALLBASE/4GL and returns the factorial of the value to the variable *result*. The external program *factorial* could be as follows:

```
program factorial(info, output);

const  WR_ACC = 1;
       CONTROLCODE = 1;

type   name_type = packed array[1..30] of char;
       error_type = packed array[1..10] of char;
       status_type = record
           info : integer;
```

12-56 Logic Commands


```

        subsys : integer;
    end;

var    fname : name_type;
       filenum: integer;
       num, i : integer;
       status : status_type;
       domain : integer;
       access : integer;
       ew : integer;
       t : 1..128;
       info : string[255];
       str : string[32];
       buf1, buf : string[128];

/* These are the intrinsic declarations */

       procedure HPFOPEN; intrinsic;
       procedure FCONTROL; intrinsic;
       procedure FWRITE; intrinsic;
       procedure TERMINATE; intrinsic;

/* This function calculates the factorial of the number passed to it. */

function fact(n: integer): integer;
begin
    if n <= 0 then
        fact := 1
    else
        fact := n*fact(n-1);
end; /* fact */

/* Error routine to exit out of program */

procedure error(err_str : error_type);
begin
    writeln(err_str, ' failed!');
    TERMINATE;
end; /* error */

begin /* main */
    domain := 2;
    access := 1;
    ew := 1;
    num := 0;
    i := 1;
    buf1 := 'EXTERNAL!';
    fname := '&HP4EXTP&';

```

```

    { * Convert the 1st parameter to an integer * }

    while (info[i] <> ' ') do { first space denotes end of field }

    begin
        num := num*10 + ord(info[i]) - ord('0');
        i := i + 1
    end;

    { * Convert the factorial to a string * }

    strwrite(str,1,t,fact(num):1);

    { * Send parameters back to HP ALLBASE/4GL by writing them on
      the message file 'HP4EXTP'. * }

    HPFOPEN(filename, status, 2, fname, 3, domain, 11, access);
    if status.info <> 0 then
        error('HPFOPEN');

    FCONTROL(filename, 45, ew); { ** extended wait ** }
    if ccode <> 2 then
        error('FCONTROL');

    FWRITE(filename, buf1, -9, CONTROLCODE);
    if ccode <> 2 then
        error('FWRITE');

    strwrite(buf,1,t,'2 ',strlen(str):1,' ',str,chr(13));
    FWRITE(filename, buf, -(strlen(buf)), CONTROLCODE);
    if ccode <> 2 then
        error('FWRITE');
end. { * main * }

```

12-58 Logic Commands

Example 2

EXTERNAL factorial "5" V-result

This example shows a similar program written in COBOL. The external command calls the external program *factorial*. The external program receives a value (in this case 5) from HP ALLBASE/4GL and returns the factorial of the value to the variable *result*. The external program *factorial* could be as follows:

```
Identification Division.
Program-id. Factorial.
Author. HP ASO
Date-written. 08/09/88.

Environment Division.
Configuration Section.
Source-computer. HP3000 series 930.
Object-computer. HP3000 series 930.
Special-names.
    condition-code is ccode.

Data Division.
Working-storage section.
77 str                pic x(32) value spaces.
77 cntstr             pic x(9) value spaces.
77 tmpstr            pic x(32).
77 num               pic 9(9).
77 factorial         pic 9(9) value 1.
77 numz              pic z(8)9.
77 cr                pic x value %15.
77 counter           pic s9(9) comp value 0.
01 intrinsic-parms.
    05 infostring    pic x(128) usage display.
    05 infolength    pic s9(4) comp value 128.
    05 filenum       pic s9(9) comp.
    05 status-val    pic s9(9) comp.
    05 status-valr   redefines status-val.
        08 sw1       pic s9(4) comp.
        08 sw2       pic s9(4) comp.
    05 it2           pic s9(9) comp value 2.
    05 pname         pic x(9) value "&HP4EXTP&".
    05 it3           pic s9(9) comp value 3.
    05 domain        pic s9(9) comp value 2.
    05 it11          pic s9(9) comp value 11.
    05 accesstype    pic s9(9) comp value 1.
    05 fconcode      pic s9(4) comp value 45.
    05 ew            pic 9(4) value 1.
    05 ename         pic x(9) value "EXTERNAL!".
    05 len           pic s9(4) comp value 9.
```

```

05 controlcode      pic s9(4) comp value 0.
05 buf              pic x(128) value spaces.
05 strlen           pic s9(9) comp.

```

Procedure Division.

S0100-main section.

```

perform P0110-init.
perform P0120-calculate-factorial.
perform P0130-pack-string.
perform P0150-open-pipe.
perform P0170-write-parms.
perform P0999-end.

```

P0110-init.

```

*
*   Get parameters from HP ALLBASE/4GL
*
call intrinsic "getinfo" using infostring, infolength.
*
*   Split up information recieved
*
unstring infostring delimited by all spaces into num,
      tmpstr.

```

P0120-calculate-factorial.

```

*
*   Calculate the factorial
*
if num <= 1 then
    move 1 to factorial
else
    perform P0125-factorial until num = 1.

```

P0125-factorial.

```

multiply num by factorial giving factorial.
subtract 1 from num.

```

P0130-pack-string.

```

*
*   Pack factorial value into string to be passed back to
*   ALLBASE/4GL.
*
move factorial to numz.
unstring numz delimited by all spaces into tmpstr str.
inspect str tallying counter for characters before initial
      space.
move counter to numz.
unstring numz delimited by all spaces into tmpstr cntstr.
string "2", ";", cntstr, ";", str, cr delimited by
      spaces into buf.

```

12-60 Logic Commands

```
inspect buf tallying strlen for characters before initial
space.
inspect buf replacing all ";" by space.
```

```
P0150-open-pipe.
```

```
*
* Send parameters back to HP ALLBASE/4GL by writing them on
* message file
*
call intrinsic "hpfopen" using filenum, status-val,
it2, pname, it3, domain,
it11, accesstype.
if status-val is not = zero then
display "HPFOPEN failed! (" sw1 "," sw2 ")".
perform P0999-end.
call intrinsic "fcontrol" using filenum, fconcode, ew.
if ccode not = 0 then
display "FCONTROL failed!".
perform P0999-end.
```

```
P0170-write-parms.
```

```
multiply len by -1 giving len.
call intrinsic "fwrite" using filenum, ename, len,
controlcode.
if ccode not = 0 then
display "FWRITE failed!".
perform P0999-end.
multiply strlen by -1 giving strlen.
call intrinsic "fwrite" using filenum, buf, strlen,
controlcode.
if ccode not = 0 then
display "FWRITE failed!".
perform P0999-end.
```

```
P0999-end.
```

```
stop run.
```

Example 3

EXTERNAL factorial "5" V-result

This example shows a similar program written in C. The external command calls the external program *factorial*. The external program receives a value (in this case 5) from HP ALLBASE/4GL and returns the factorial of the value to the variable *result*. The external program *factorial* could be as follows:

```
#include <stdio.h>
#include "mpe.h"

extern void  HPFOPEM();
extern void  FCONTROL();
extern void  FWRITE();
extern int   ccode();

#define CONTROLCODE    0
#define fpr(x) (fprintf(stderr, "%s failed!\n", x))

main (argc, argv)
int   argc;
char  *argv[];
{
    int    num,
           factorial,
           filenum,
           domain = 2,
           access = 1,
           status;
    char  str[20],
           buf[128],
           fname[128];
    short ew = 1;

    /*
     * Convert the 1st parameter to an integer
     */
    num = atoi(argv[1]);

    /*
     * Calculate the factorial
     */
    for (factorial = 1; num > 1; num--)
        factorial *= num;

    /*
     * Convert the factorial to a string
     */
}
```

12-62 Logic Commands

```

sprintf(str, "%d", factorial);

/*
 * Pack string to be passed back to ALLBASE/4GL
 */
sprintf(buf, "2 %d %s\n", strlen(str), str);

/*
 * Send parameters back to ALLBASE/4GL by writing them on
 * the message file
 */
sprintf(fname, "%HP4EXTP&");
HPFOPEN(8, &filenum, &status, 2, fname, 3, &domain, 11, &access);
if (status != 0) {
    fpr("HPFOPEN");
    exit(1);
}

FCONTROL(filenum, 45, (unsigned short ~) &ew);
err_check("FCONTROL");

sprintf(fname, "EXTERNAL!");
FWRITE((short)filenum, (char ~)fname, (short)-9, CONTROLCODE);
err_check("FWRITE");

FWRITE((short)filenum, (char ~)buf, (short)-strlen(buf), CONTROLCODE);
err_check("FWRITE");
}

err_check(name)
char *name;
{
    if (ccode() != CCE) {
        fpr(name);
        exit(1);
    }
}
}

```

FIELD Command

The FIELD command changes the attributes or behavior of one or more fields on the current screen.

Formats

$$\text{FIELD} \left\{ \begin{array}{l} [*DEFAULT] \left[\begin{array}{l} *INPUT \\ *DISPLAY \end{array} \right] \left[\begin{array}{l} *REQUIRED \\ *OPTIONAL \end{array} \right] \left[\begin{array}{l} *ECHO \\ *NOECHO \end{array} \right] \left[[attribute] \dots \right] \\ [from_data_ref [to_data_ref] \end{array} \right\}$$

Parameters

from_data_ref, to_data_ref

The starting and ending field numbers to be changed. Each can be one of the following:

- Whole number.
- Screen field name.
- * (representing the current field+).
- Relative screen field reference such as *-9 to *+9.

In the FIELD command you can specify a screen field by the field name only. You can optionally omit the screen name. If you specify only one data reference, HP ALLBASE/4GL changes the attributes or behavior of that field only.

If you don't specify any data references, the FIELD command applies to all fields on the screen.

attribute

The attributes that you want to apply to the named screen fields. *Attribute* can be any of the following:

12-64 Logic Commands

Color Attributes

*BLACK
*BLUE
*CYAN
*GREEN
*MAGENTA
*RED
*WHITE
*YELLOW

Highlight attributes

*NOINV
*INV
*FBRIGHT
*HBRIGHT
*BLINK
*NOBLINK
*ULINE
*NOULINE

Description

The FIELD command allows you to change the attributes or behavior of one field or a range of fields on the current screen. Attribute keywords can be listed in any order after the FIELD command.

All attributes changed by this command remain in effect until they are changed by another FIELD command or a different screen is displayed.

If you specify both a *from_data_ref* and a *to_data_ref* with the FIELD command, HP ALLBASE/4GL changes the attributes of all fields in the range on the current screen. If the *from_data_ref* is greater than the *to_data_ref*, the FIELD command doesn't change the attributes of any field.

If you use the FIELD command to change an attribute of a field and the field already has that attribute, no change is noticeable to the user.

The *INPUT and *DISPLAY keywords change the specified field or fields to be either input or display fields. The field highlighting changes to reflect the system-wide specifications set by the system administrator on the terminal display control screen.

When a field attribute is explicitly set by a FIELD command, that attribute is not altered by a subsequent FIELD *INPUT or FIELD *DISPLAY command. For example, assume that a field attribute for input fields is the color yellow and a field attribute for display fields is the color cyan. If a FIELD *RED command is executed on an input field, the input field changes to the color red and remains in inverse video. If a FIELD *DISPLAY command is executed on

the field sometime later, the field becomes a display field and exhibits all the attributes of a display field except that it remains red in color because of the previous FIELD command.

The *REQUIRED and *OPTIONAL keywords change the specified field or fields to be either required or optional.

Note

Changing this field attribute can affect the flow of the screen logic. If the field is changed from being optional to being required, it can never be left blank. Therefore, the contents of the *After Function if Blank* field on the screen field details screen has no effect. However, if the field is changed from being required to being optional, the contents of the *After Function if Blank* field determines whether any after function is executed.

The *ECHO and *NOECHO keywords control whether user input is displayed (echoed) on the screen or not. *NOECHO is particularly useful for password fields as it allows the user to enter a password without displaying the password on the screen.

The color keywords change the specified fields to the specified color.

The *INV and *NOINV keywords change the field attributes to and from inverse video.

The *FBRIGHT and *HBRIGHT keywords change the field attributes to and from full brightness and half brightness.

The *BLINK and *NOBLINK keywords change the field attributes to and from blinking.

The *ULINE and *NOULINE keywords change the field attributes to and from being underlined.

The *DEFAULT keyword resets the specified screen fields back to their original behavior and highlighting attributes.

If HP ALLBASE/4GL detects conflicting keywords in a FIELD command when it generates a logic block, it displays an error message.

12-66 Logic Commands

Note

Some terminals cannot support all the above attribute changes. In that case, the terminal's default setting is displayed.

Example 1

```
FIELD *DISPLAY 1 10
```

This command changes fields 1 to 10 to display fields. Each of these fields appears to the user to be display only.

Example 2

```
FIELD *OPTIONAL S-customer S-zip_code
```

This command changes the *customer* and *zip_code* fields and all intervening fields to optional fields. If any of these fields were already optional, their behavior is unchanged. However, if any field was required, the behavior changes as follows. If the *After Function if Blank* field on the screen field details screen is marked as Y, the after function is always executed. If the *After Function if Blank* field is marked as N (the default value), the after function is not executed if the screen field is left empty.

Example 3

```
FIELD *FBRIGHT *INV
```

This command changes every field on the current screen to full-bright and to inverse video.

Example 4

```
FIELD *DEFAULT *RED 2 4
```

This command resets the attributes of fields 2 through 4 back to their defaults and then changes the fields to the color red.

FILE Command (KSAM)

The file command allows you to perform various operations on application data files. You can use the file command to perform operations on HP ALLBASE/SQL tables and select lists, HP TurboIMAGE/iX data sets, KSAM data files, and serial data files.

This description covers the use of the FILE command with KSAM data files.

Formats

$$\text{FILE} \left\{ \begin{array}{l} *BUFFER \\ *CLOSE \\ *MODIFY \\ *UNLOCK \end{array} \right\} \text{file_ref} [; \text{command}]$$
$$\text{FILE} \left\{ \begin{array}{l} *FIRST \\ *LAST \\ *NEXT \\ *PREVIOUS \end{array} \right\} [*NOLOCK] \text{file_ref} [*INDEX = \text{index_name}] [; \text{command}]$$
$$\text{FILE} \left\{ \begin{array}{l} *DELETE \\ *FIND \\ *INSERT \\ *READ \\ *WRITE \end{array} \right\} [*NOLOCK] \text{file_ref} [*INDEX = \text{index_name}] [*KEY = \text{key}] [; \text{command}]$$

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	FILE				
File Operation		(*BUFFER/*CLOSE/*DELETE/*FIND/*FIRST/*INSERT/*LAST/*NOLOCK	(Y/N)	(*MODIFY/*NEXT/*PREVIOUS/*READ/*REMOVE/*UNLOCK/*WRITE)				
File ID[.Record]								
On Error								
Command	Function	Process	Generate	17* 18	System	Commit	Help	Previous
Help	Header	Detail	Function		Keys	Data		Menu

Parameters

file_ref

Specifies the HP ALLBASE/4GL name of the application file on which the operation is to be performed. It can be expressed in either of the following ways:

- *file_id*

- or

- *file_id.record_name*

If you don't specify *.record_name* for the file, HP ALLBASE/4GL uses the default record for the file.

*NOLOCK

The *NOLOCK term is a command argument to control file locking for the file access. If the *NOLOCK argument is present, no locking is performed by the file access provided the file is not already locked. If the file has already been locked by an earlier access, the *NOLOCK argument has no effect. Refer to KSAM File Locking for further information.

index_name

This parameter allows you to specify the name of the index or the number of the index to be used to access the file. The index name must be the name of a field specification that you have defined as a key field on the default record layout for the file. You can also specify the index by entering the number of an index for the file. The index you specify here is used for one file access only.

Specifying an index name or number here does not change the current value in the communication area field *INDEXNO.

If you don't specify an index name or number, HP ALLBASE/4GL accesses the file according to the index specified by the current value of *INDEXNO.

key

This parameter defines the key data to be used to access the file. It can be one of the following:

- Literal.
- Alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Non-numeric communication area field.

This parameter is optional. If you don't specify a key, HP ALLBASE/4GL accesses the file according to the existing value in the record buffer key field identified by the current value of *INDEXNO or the *INDEX= argument. If you do specify a key value, HP ALLBASE/4GL moves this value to the key field (as identified by *INDEXNO or the *INDEX= argument) in the file record buffer.

command

A command to be executed if the requested file operation fails. You can use any of the following HP ALLBASE/4GL logic commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED

12-70 Logic Commands

- SERIES
- TOP
- VISIT
- ZIP

Description

The FILE command performs various operations on KSAM data files.

Opening and Closing Files

HP ALLBASE/4GL automatically opens each KSAM file when it is first accessed in a process. There is no need to explicitly open KSAM data files. If a file is specifically closed (refer to *CLOSE below), HP ALLBASE/4GL automatically opens the file when it is next accessed.

HP ALLBASE/4GL automatically closes all data files at the start of a process.

File Indexes

The communication area field *INDEXNO specifies the number of the index used to access the file. HP ALLBASE/4GL initializes the value in this field to 1 when the application starts, assuming that all file access is via the primary file indexes. HP ALLBASE/4GL does not reset the value of *INDEXNO if you change it at any time. Therefore, you must ensure that *INDEXNO is set to the correct value before each FILE command is executed.

If you want to access records on a file with more than one index, you must set *INDEXNO to the number of the required file index before the FILE command is executed.

You can specify a different index for a particular file access by specifying the name of a field defined as a key field on the default record layout for the file in *INDEX= argument on the FILE command. You can also specify the index for a file access by specifying the number of an index in *INDEX= clause.

File Access Modes

You must specify any files that are to be updated in the current process by using the MODE command with the *WRITE, *LOCK, or *UNLOCK

argument. Any files not specified are available for reference purposes only.
(Refer to the MODE command.)

File Record Buffers

HP ALLBASE/4GL maintains a separate buffer for each record layout defined for each file.

HP ALLBASE/4GL clears the file record buffer or buffers automatically when a file is first accessed, or at the start of any process.

Where multiple records have been defined for an indexed (KSAM) file, data in one record buffer is not automatically available to another record buffer for the same file. To make the data available, you can read the record again, or move the data between buffers using either the MOVE or MOVELOOP commands.

Communication Area Fields

HP ALLBASE/4GL places the external name of the file currently being accessed in the communication area field *FILENAME.

File Operations

Operations you can perform on KSAM files are:

***BUFFER.** Clears the specified buffer of all data. The buffer cleared is defined by *file_ref*.

***CLOSE.** Closes the nominated file. You can use this operation to close any file that is not required. HP ALLBASE/4GL automatically reopens the file if it is accessed after being closed. Closing a file releases all locks belonging to the current process, regardless of the original file access mode.

***DELETE.** Deletes a record from the file. If the file has unique primary keys, the record deleted is determined by the primary key value.

If the file allows duplicate primary key values, you must read a record before it can be deleted. If you haven't read a record, HP ALLBASE/4GL returns an error number 19112.

After a delete operation, the file pointer position may be undefined. You may need to use a FILE *FIND, or a FILE *READ command to reposition the file pointer before performing further operations.

12-72 Logic Commands

***FIND.** Finds the first record in the file whose key is greater than or equal to the value of *key*. The file key is tested using the index number specified in *INDEXNO or by *INDEX=.

HP ALLBASE/4GL does not place data into the record buffer as a result of a *FIND. To retrieve the record found, use the operation *NEXT.

HP ALLBASE/4GL only maintains one file position pointer for the *FIND operation. To retrieve a record located by the *FIND operation, you must use the *NEXT operation to read the record before you perform any other operation on the same file.

You can use *FIND to locate a record in the file when the complete value of the key is not specified. For example, If *key* contains SMIT, *FIND will find SMITH.

The *FIND operation returns error status 19111 (record not found) if there is no record with a key value equal to or greater than the value of *key*.

***FIRST.** Reads the first record in the file for the index specified by the current value in *INDEXNO or by *INDEX=.

***INSERT.** Adds a new record to the file.

This operation forces new records onto the file if duplicate primary key values have been specified for the file.

If the file does not allow duplicate primary key values, using *INSERT will cause an error if you attempt to add a record to the file with a primary key value the same as one that already exists.

***LAST.** Reads the last record in the file for the index specified by the current value in *INDEXNO or by *INDEX=.

***NEXT.** Reads the next record in the file for the index specified by the current value in *INDEXNO or by *INDEX=.

***MODIFY.** Changes an existing record in the file.

You must read a record before you can use the *MODIFY command to change its contents.

***PREVIOUS.** Reads the previous record in the file for the index specified by the current value in *INDEXNO or by *INDEX=.

***READ.** Reads the record in the file whose key value, for the index specified by the current value in *INDEXNO or by *INDEX=, exactly matches the value specified by *KEY=*key*. If you don't specify the *KEY=*key* argument on the FILE command, HP ALLBASE/4GL uses the current value in the record buffer field specified by *INDEXNO or the *INDEX= argument.

***UNLOCK.** The *UNLOCK argument releases the locks on a file that has been accessed under MODE *WRITE or MODE * UNLOCK. HP ALLBASE/4GL ignores this argument if the file is being accessed under MODE *READ or MODE *LOCK.

***WRITE.** Writes a new record or modifies an existing record on the file.

Don't use this operation if the file allows duplicate primary key values. Use the *INSERT operation instead.

If the file does not allow duplicate primary key values, then one of the following actions occurs:

- If a record exists on the file whose primary key matches the primary key value in the record buffer, the file record is modified. The modify is performed in the same manner as the *MODIFY operation.
- If a record does not exist, a new record is added to the file. The add is performed in the same manner as the *INSERT operation.

*WRITE performs the same operation as the UPDATE command except that it operates on one file only.

Error Handling

If the FILE command contains an optional logic command, HP ALLBASE/4GL executes the command if an error occurs as a result of the file access. For example, when performing a *NEXT operation, the optional command is executed if the end of the file is encountered.

When an error occurs, the HP ALLBASE/4GL data manager writes an error code into the communication area field *IOSTATUS. Appendix C lists the data manager error codes. The value returned to *IOSTATUS is zero if the file command is successful.

If you include the optional error command, the data manager errors for the following conditions are not displayed on the terminal screen:

12-74 Logic Commands

- Duplicate key errors (error 19100).
- Record locked (error 19107).
- End of file (error 19110).
- Record not found (error 19111).
- Record not read (error 19112).
- File exclusively locked (error 19113).
- File already locked (error 19130).

For other error conditions, even if you have specified an error command, the error message is displayed on the screen.

File Errors

The following table summarizes the most common file error values returned to the communication area field *IOSTATUS. Refer to appendix C for a list of MPE/iX KSAM file error returns.

*IOSTATUS

Value	Condition
19000	MPE/iX file access error detected by the KSAM file manager
19013	MPE/iX error. Access permission denied to file.
19100	Duplicate primary key error. The file only allows unique primary key values, and a record with the same primary key value already exists.
19107	Record locked error. Another user or process has locked the record you are attempting to access.
19110	Beginning or end of file reached during a *NEXT or *PREVIOUS file operation.
19111	Record not found. No record has a key value matching the value specified for *KEY= <i>key</i> .
19112	Record not read. No record has been read for the file.
19113	Exclusive locking error. Another user or process has locked the file.
19130	File locking error. The current process cannot lock the file because the file is already locked by another process.

KSAM File Locking

HP ALLBASE/4GL provides file locking for entire files only. The KSAM data manager does not provide record level locking. The file locking system prevents two users modifying the same file in such a way that records or the file structure may be corrupted.

If a file is locked, another user or process cannot lock the same file while the lock is current.

Data file locking in HP ALLBASE/4GL depends on both the file operation used with the FILE command, and the file access mode specified in the MODE command. The following table summarizes the file access and locking that can occur if two users (or a user and a background process) attempt to access the same data file.

	Existing Access Mode			
	*LOCK	*WRITE	*READ	*UNLOCK
Attempted access mode	File locked on first access	File locked on each access		File locked on each access
*LOCK	Not permitted	Not permitted	Permitted	Not permitted
*WRITE Read or write operation	Not permitted	Not permitted	Permitted	Not permitted
*READ	Permitted	Permitted	Permitted	Permitted
*UNLOCK	Not permitted	Not permitted	Permitted	Not permitted

Under the MODE *LOCK, MODE *WRITE, or MODE *UNLOCK commands, the file is normally locked when it is first accessed. However, the file is not locked when you first access a file if you access the file using a command that contains the *NOLOCK argument.

When a file is accessed under MODE *WRITE or MODE *UNLOCK, the file remains locked until the end of the process, the file is closed with the FILE *CLOSE command, or the file is unlocked with a FILE *UNLOCK command.

12-76 Logic Commands

When a file is accessed under MODE *LOCK, the file remains locked until the end of the process, or until the file is closed with the FILE *CLOSE command.

All file locks are also released by a TRANSACT *END command or a TRANSACT *UNDO command.

When a file is accessed under MODE *READ, no file locking is performed.

At any time, you can explicitly specify that a particular file access does not lock a file by specifying the *NOLOCK argument on the FILE command.

Note that any time you read a file without locking the file, HP ALLBASE/4GL clears the MPE/iX buffer using the FCONTROL intrinsic with code 7 to ensure that the record is read from the disk. This is necessary to ensure that the latest possible information is obtained by the read. As a result, performing a large number of reads on a KSAM file without locking the file can create a substantial disk I/O overhead.

***LOCKWAIT Communication Area Field**

HP ALLBASE/4GL uses the value in the communication area field *LOCKWAIT to determine its behavior when it attempts to access a file that is locked by another process. The *LOCKWAIT field can contain the values -1, 0, or a positive integer number. The actions taken by HP ALLBASE/4GL for different values in *LOCKWAIT are summarized below.

Value	Action
+n	HP ALLBASE/4GL waits for <i>n</i> seconds to access the file. If the file is not accessible after <i>n</i> seconds, HP ALLBASE/4GL returns an error number 19107.
0	HP ALLBASE/4GL attempts to access the file every second for 5 seconds. If the file is not accessible after 5 seconds, HP ALLBASE/4GL returns an error number 19107.
-1	HP ALLBASE/4GL waits indefinitely to access locked files.

When HP ALLBASE/4GL starts an application, the value in *LOCKWAIT is initialized to zero. You can use the MOVE command to move a different value into *LOCKWAIT. HP ALLBASE/4GL does not reset the value in *LOCKWAIT at any time except when the application starts.

Example 1

```
5 FILE *READ stock *KEY=S-stock_no.stock_add; ENTER 12
.
.
12 MESSAGE no_stock
```

This command reads the file *stock*, and searches for a record with its key value equal to the value of the field *stock_no* on the screen *stock_add*. If the record cannot be located, the system executes step 12, which displays the message *no_stock*.

Example 2

```
FILE *BUFFER trans.cust
```

This command clears the buffer for the record *cust* in the file *trans*. **Example 3**

```
4 FILE *NEXT trans; ENTER 11
5 SHOW *REFRESH
6 TOP
.
.
11 MESSAGE no_transactions
```

These commands read and display records from the file *trans*. When the system encounters the end of the file, it executes step 11 which displays the message *no_transaction*.

Example 4

```
FILE *WRITE stock *KEY=*S01
```

This command checks if a record with a key equal to the contents of screen field number 1 exists on the file *stock*. If such a record exists, the system updates it. If the record doesn't exist, the system creates a new record.

Example 5

```
1 MODE *LOCK filename
2 FILE *READ filename *KEY=*; ENTER 8
.
.
```

12-78 Logic Commands

```
5 FILE *WRITE filename *KEY=*; ENTER 8
6 FILE *CLOSE filename
7 EXIT
8 FILE *CLOSE filename
9 MESSAGE fileerrmsg
10 EXIT
```

In this logic block, the file *filename* is locked to the logic block. The file is locked at step 2 when the first file access occurs. When the logic block is completed or an error occurs, the file is closed, releasing the file locks.

FILE Command (Serial)

The file command allows you to perform various operations on application data files. You can use the file command to perform operations on HP ALLBASE/SQL tables and select lists, HP TurboIMAGE/iX data sets, KSAM data files, and serial data files.

This description covers the use of the FILE command to perform operations on serial files.

Formats

```
FILE { *BUFFER
      *CLOSE
      *FIRST
      *LAST
      *MODIFY
      *NEXT
      *PREVIOUS
      *REMOVE } file_ref [ ; command ]
```

```
FILE { *INSERT
      *READ
      *WRITE } file_ref [ *REC=record_number ] [ ; command ]
```

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	FILE
File Operation	*NOLOCK	(Y/N)	(*BUFFER/*CLOSE/*DELETE/*FIND/*FIRST/*INSERT/*LAST/*MODIFY/*NEXT/*PREVIOUS/*READ/*REMOVE/*UNLOCK/*WRITE)	
File ID[.Record]				
On Error				
Command Help	Function Header	Process Detail	Generate Function	17* 18 System Keys
				Commit Data
				Help
				Previous Menu

12-80 Logic Commands

Parameters

file_ref

Specifies the HP ALLBASE/4GL name of the serial file on which the operation is to be performed. You can express this as:

- *file_id*

- or

- *file_id.record_name*

If you don't specify *record_name* for the file, HP ALLBASE/4GL uses the default record for the file.

record_number

HP ALLBASE/4GL allows you to access a fixed length record serial file record by specifying the number of the record. Record number 1 is the first physical record in the file. You can also use a record number with the FILE *INSERT operation for a fixed length record serial file. In this case, the record number specifies the number of the record that is overwritten by the current contents of the file record buffer.

You can specify *record_number* with any of the following items containing a non-negative integer number.

- A whole number.
- A numeric constant.
- A variable or calculated item.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- A communication area field.

command

A command to be executed if the requested file operation fails. You can use any of the following HP ALLBASE/4GL logic commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

Description

The FILE command allows you to perform various operations on serial files.

HP ALLBASE/4GL supports serial files with fixed length records, or variable length records.

Fixed length record serial files have records that contain a fixed number of characters as defined by an HP ALLBASE/4GL record layout.

When HP ALLBASE/4GL reads a record from a variable length serial file, it reads characters until it encounters the end of the record. Characters read from the file are read into the file buffer as defined by the record layout specified on the FILE command.

If HP ALLBASE/4GL reads fewer characters than the length of the file buffer, the unused characters in the buffer are filled with spaces.

If you attempt to read a record longer than the record buffer, HP ALLBASE/4GL reads the number of characters that the buffer can accommodate, and sets the switch *MOREREC *on*. The next FILE *READ command or FILE *NEXT command either reads the remaining characters in the record, or reads the number of characters that the buffer can accommodate and then sets *MOREREC *on* again if the record still contains unread characters.

HP ALLBASE/4GL substitutes space characters for all non-printable characters and control characters read from serial files.

12-82 Logic Commands

When HP ALLBASE/4GL writes to a fixed length serial file, it writes the data contained in the current file record buffer. The number of characters written to the file is determined by the length of the default record for the file. If the current record layout is longer than the default record layout for the file, the additional characters are truncated.

When HP ALLBASE/4GL writes to a variable length record serial file, the length of the record written to the file is determined by the length of current record layout. HP ALLBASE/4GL discards any trailing blanks.

You must specify any serial files that are to be updated by the FILE command in the current process by using a MODE command with the *WRITE argument. Any files not specified in the MODE command cannot be updated or modified from HP ALLBASE/4GL logic and are available for reference purposes only. (Refer to the MODE command.)

When accessing a serial file, HP ALLBASE/4GL places the external name of the file in the communication area field *FILENAME.

If the file has fixed length records, HP ALLBASE/4GL places the number of the record read or written in the communication area field *RECNO.

Serial File Creation

If a serial file does not already exist, HP ALLBASE/4GL automatically creates it when the application first accesses the file. The process containing the FILE command that creates the file must contain a MODE *WRITE *file_name* command to give the process write access permission to the file.

The dictionary data file creation screen does not allow you to create serial files.

Temporary Serial Files

HP ALLBASE/4GL allows you to specify the external name of a serial file as *TEMP on the dictionary file definition screen.

HP ALLBASE/4GL creates a temporary file when the application first accesses the the file. The file is deleted when you close it, or at the end of the process. The process must contain a MODE *WRITE *file_name* command.

HP ALLBASE/4GL creates temporary serial files in the temporary domain of the user's current group and account.

Dynamic File Naming

HP ALLBASE/4GL allows you to define a serial file with the external name specified as *FILENAME on the dictionary file definition screen.

By moving the external file name into the communication area field *FILENAME before HP ALLBASE/4GL executes the FILE command that opens the file, you can specify which file the command operates on.

Normally, HP ALLBASE/4GL appends the current value of the MPE/iX variable HP4DATAPATH to the value in *FILENAME to determine the external name of the file. However, if *FILENAME contains a fully qualified file name, (that is, a file name in the form FILE.GROUP.ACCOUNT), HP ALLBASE/4GL performs the operation on the file specified by this name.

Error Handling

If the FILE command contains an optional error command, HP ALLBASE/4GL executes the command if an error occurs as a result of the file access. For example, when performing a *NEXT operation HP ALLBASE/4GL executes the optional command if the end of the file is encountered.

When an error occurs, the data manager writes an error code into the communication area field *IOSTATUS. Appendix C lists the data manager error codes. The value in *IOSTATUS is zero if the file command is successful.

If you include the optional error command, the data manager error messages will not be displayed on the terminal screen.

File Errors

The following table summarizes the most common file error values returned to the communication area field *IOSTATUS. Refer to appendix C for a complete list of error returns.

*IOSTATUS

Value	Condition
19002	File does not exist
19013	Access permission denied
19110	Beginning or end of file encountered by a FILE *NEXT or FILE *PREVIOUS command.

File Record Buffers

HP ALLBASE/4GL maintains a separate buffer for each record layout for a serial file.

HP ALLBASE/4GL clears the file record buffers automatically when a file is first accessed, or at the start of any process.

Serial File Locking

HP ALLBASE/4GL does not provide any locking mechanism for serial files.

Transaction Logging

HP ALLBASE/4GL does not provide any transaction logging or transaction roll-back mechanism for serial files.

File Operations

Operations that may be performed on serial files are:

***BUFFER.** Sets all fields in the specified file record buffer to spaces. The buffer cleared is defined by *file_ref*.

***CLOSE.** Closes the nominated file. If the file is a temporary file, this operation also deletes the file.

***FIRST.** Reads the first record in the file.

***INSERT.** For both fixed length record files and variable length record files, this operation adds a new record to the end of the file. For fixed length record files only, this operation overwrites the record at the specified position if you

use the **REC=record_number* argument. Record number 1 is the first physical record in the file.

***LAST.** Fixed length record serial files only. This argument reads the last record in the file.

***MODIFY.** Fixed length record serial files only. This form of the command changes the contents of the current record. You must read a record before you can use the **MODIFY* argument to change it.

***NEXT.** Reads the next record in the file. If a FILE **NEXT* command encounters the end of file, the value returned to **IOSTATUS* is 19110.

For variable length record files, HP ALLBASE/4GL sets the switch **MOREREC on* if the current record contains more characters than the buffer can hold. The next FILE **NEXT* or FILE **READ* command retrieves the remaining characters in the record. If the current record is the same length as the file buffer or shorter than the buffer, the switch **MOREREC* remains *off*.

***PREVIOUS.** Fixed length record serial files only. This command reads the previous record in the file.

***READ.** Reads the next record in the file. For fixed length record serial files only, you can use the **REC=record_number* argument to read a specific record. Without the **REC=* argument, this operation is identical to the **NEXT* operation. Record number 1 is the first physical record in the file.

For variable length record files, HP ALLBASE/4GL sets the switch **MOREREC on* if the current record contains more characters than the buffer can hold. The next FILE **NEXT* or FILE **READ* command retrieves the remaining characters in the record. If the current record is the same length as the file buffer or shorter than the buffer, the switch **MOREREC* remains *off*.

***REMOVE.** Physically deletes the file. The file is deleted regardless of the mode specified for the file in the current process.

The user must have the correct MPE/iX authorities to be able to delete the file.

***WRITE.** Adds a record to the end of the file. For fixed length record serial files you can use the **REC=* argument to specify the number of the record to be written. This form of the command overwrites the record at the position identified by **REC*.

12-86 Logic Commands

Example 1

```
1  MODE *LOCK outfile
2  FILE *FIRST infile; ENTER 7
3  MOVE R-infile R-outfile
4  FILE *INSERT outfile; ENTER 8
5  FILE *NEXT infile; ENTER 7
6  ENTER 3
7  IF *IOSTATUS = N-end_of_file
   THEN MESSAGE read_complete; EXIT
   ELSE MESSAGE infile_error; EXIT
8  MESSAGE outfile_err
9  EXIT
```

This logic block shows a technique for importing data from a serial file *infile* and writing it out to an indexed file or an HP ALLBASE/SQL table *outfile* for use in an HP ALLBASE/4GL application.

Both files use the same record layout format. Records read from the input serial file are moved from the record buffer for the input file to the buffer for the output file and written to the output file. The process continues until an error condition occurs by reaching the end of the input file.

FILE Command (HP ALLBASE/SQL)

The file command allows you to perform various operations on application data files. You can use the file command to perform operations on HP ALLBASE/SQL tables and select lists, HP TurboIMAGE/iX data sets, KSAM data files, and serial data files.

This description covers the use of the FILE command to perform operations on SQL tables and select lists.

Also refer to chapter 9 for more information about working with HP ALLBASE/SQL databases from within HP ALLBASE/4GL.

Formats

FILE { *BUFFER
*CLOSE
*DELETE } *file_ref* [; *command*]
*INSERT
*NEXT

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	FILE
File Operation		(*BUFFER/*CLOSE/*DELETE/*FIND/*FIRST/*INSERT/*LAST/*NLOCK	(Y/N)	(*MODIFY/*NEXT/*PREVIOUS/*READ/*REMOVE/*UNLOCK/*WRITE)
File ID[.Record]				
On Error				
Command Help	Function Header	Process Detail	Generate Function	17* 18
			System Keys	Commit Data
			Help	Previous Menu

Parameters

file_ref

Specifies the HP ALLBASE/4GL name of the HP ALLBASE/SQL table or select list on which the operation is to be performed. You must use the HP ALLBASE/4GL dictionary names for HP ALLBASE/SQL tables with this command.

12-88 Logic Commands

command

A command to be executed if the requested file operation fails. You can use any of the following HP ALLBASE/4GL logic commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

Description

The FILE command allows you to perform various operations on HP ALLBASE/SQL tables and select lists.

Before you can use the FILE *NEXT, FILE *DELETE, and FILE *CLOSE forms of the FILE command to manipulate HP ALLBASE/SQL data, you must execute a SELECT command in an SQL logic block. This command implicitly declares and opens an HP ALLBASE/SQL cursor.

You must specify any base tables that are to be updated by the FILE command in the current HP ALLBASE/4GL process by using a MODE command with the *WRITE argument. Any files not specified in the MODE command cannot be updated or modified from HP ALLBASE/4GL logic and are available for reference purposes only. (Refer to the MODE command.)

Error Handling

If the FILE command contains an optional error command, HP ALLBASE/4GL executes the command if an error occurs as a result of the file access. For example, when performing a *NEXT operation HP ALLBASE/4GL executes the optional command if the end of the file is encountered.

When an error occurs, the HP ALLBASE/4GL data manager writes an error code into the communication area field *IOSTATUS. Appendix C lists the data manager error codes. The value in *IOSTATUS is zero if the file command is successful.

If you include the optional error command, the data manager error messages will not be displayed on the terminal screen.

File Errors

The following table summarizes the most common file error values returned to the communication area field *IOSTATUS. Refer to appendix C for a complete list of the HP ALLBASE/SQL file error return values.

***IOSTATUS**

Value	Condition
60000	No SQL data base has been defined for the application.
60010	HP ALLBASE/4GL is unable to open the HP ALLBASE/SQL message catalog.
60110	Beginning or end of file encountered by a FILE *NEXT command.
60112	A record has not been read for the file. You cannot update or delete a record with the HP ALLBASE/4GL FILE command, or use the WHERE CURRENT OF clause in an SQL logic block unless you read a record to position the cursor.
60150	The requested file operation cannot be performed on an SQL select list. You cannot insert a new record into a select list.
60151	The requested FILE *DELETE operation is out of sequence. You must use a SELECT command in an SQL logic block to open a cursor, and a FILE *NEXT command to position the cursor before you can delete a record.
60153	The requested FILE *NEXT operation is out of sequence. You must use a SELECT command in an SQL logic block to open a cursor before you can use the FILE *NEXT command to read a record.
102509	\Index uniqueness violation. You are attempting to insert a new record that causes duplicate values in a column defined as a unique index.

File Record Buffers

HP ALLBASE/4GL maintains a separate file record buffer for each HP ALLBASE/SQL table and each select list. Note that you cannot use multiple record layouts with HP ALLBASE/SQL tables.

HP ALLBASE/4GL clears the file record buffer automatically when a file is first accessed, or at the start of any process.

HP ALLBASE/SQL Table Locking

All HP ALLBASE/SQL transactions initiated from HP ALLBASE/4GL are subject to the HP ALLBASE/SQL locking mechanisms. Refer to the *HP ALLBASE/SQL Reference Manual* for more information.

File Operations

Operations that may be performed on base tables or select lists are:

***BUFFER.** Sets all fields in the specified file record buffer to nulls. The buffer cleared is defined by *file_ref*. The *BUFFER operation on an SQL table does not release any locks.

***CLOSE.** Closes the nominated file. This command performs the same function as the HP ALLBASE/SQL CLOSE CURSOR command, closing the cursor for a base table or a select list. The COMMIT WORK command in an SQL logic block also closes all open HP ALLBASE/SQL cursors.

You may wish to use the *CLOSE operation to release the read locks held on a table before you complete a transaction. To do this you must start the transaction with a BEGIN WORK CS (cursor stability) command.

In cursor stability mode, page level read locks are released as the cursor moves off the page. By using this approach, you can use tables for reading without locking other users out of the same tables while you keep your transaction active.

Since the locks are released when the cursor is closed, do not use the data read from that cursor as a basis for a later update. This can introduce inconsistencies in the database. Refer to the *HP ALLBASE/SQL Reference Manual* for more information about cursor stability and its implications.

***DELETE.** Deletes a record from a base table. You can only use the FILE *DELETE command on base table cursors, or on a select list that only accesses a single table, and does not use a GROUP BY clause. To use this command, you must declare and open a cursor with a SELECT command in an SQL logic block, and then position the cursor with a FILE *NEXT command in an HP ALLBASE/4GL logic block.

***INSERT.** Adds the current file buffer contents to a base table as a new record. This operation will cause an error if you attempt to insert a new record which

12-92 Logic Commands

would result in duplicate values in a column defined as a unique index. If an index uniqueness error occurs, the value returned to *IOSTATUS is 102509.

***NEXT.** Reads the next record in the active set for the base table cursor or select list cursor. This command performs the same function as the HP ALLBASE/SQL FETCH command. If a FILE *NEXT command encounters the end of file, the value returned to *IOSTATUS is 60110. To use this command, you must first declare and open a cursor with a SELECT command in an SQL logic block.

Using HP ALLBASE/SQL Data

In combination, the FILE command and HP ALLBASE/SQL commands in SQL logic blocks allow you to retrieve records from HP ALLBASE/SQL tables and views, insert new records in base tables, modify existing base table records, and delete base table records.

Refer to chapter 9 for a description and examples of the use of the FILE command to manipulate SQL data.

FILE Command (HP TurboIMAGE/iX)

The FILE command allows you to perform various operations on application data files. You can use the file command to perform operations on HP ALLBASE/SQL tables and select lists, HP TurboIMAGE/iX data sets, KSAM data files, and serial data files.

This description covers the use of the FILE command with HP TurboIMAGE/iX data sets.

Formats

$$\text{FILE} \left\{ \begin{array}{l} *BUFFER \\ *CLOSE \\ *MODIFY \\ *UNLOCK \end{array} \right\} \text{file_ref} [; \text{command}]$$
$$\text{FILE} \left\{ \begin{array}{l} *NEXT \\ *PREVIOUS \end{array} \right\} [*NOLOCK] \text{file_ref} [*INDEX = \text{index_name}] \\ [; \text{command}]$$
$$\text{FILE} \left\{ \begin{array}{l} *DELETE \\ *FIND \\ *FIRST \\ *INSERT \\ *LAST \\ *WRITE \end{array} \right\} [*NOLOCK] \text{file_ref} [*INDEX = \text{index_name}] [*KEY = \text{key}] \\ [; \text{command}]$$
$$\text{FILE} *READ \left\{ \begin{array}{l} [*REC = \text{recno}] \\ [*KEY = \text{key}] [*INDEX = \text{index_name}] \end{array} \right\} [*NOLOCK] \text{file_ref}$$

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	FILE				
File Operation		(*BUFFER/*CLOSE/*DELETE/*FIND/*FIRST/*INSERT/*LAST/*NOLOCK	(Y/N)	(*MODIFY/*NEXT/*PREVIOUS/*READ/*REMOVE/*UNLOCK/*WRITE)				
File ID[.Record]								
On Error								
Command	Function	Process	Generate	17* 18	System	Commit	Help	Previous
Help	Header	Detail	Function		Keys	Data		Menu

Parameters

file_ref

Specifies the HP ALLBASE/4GL name of the application data set on which the operation is to be performed. It can be expressed in either of the following ways:

- *file_id*

- or

- *file_id.record_name*

If you don't specify *.record_name* for the data set, HP ALLBASE/4GL uses the default record for the data set.

*NOLOCK

The *NOLOCK term is a command argument to control file locking for the data set access. If the *NOLOCK argument is present, no implicit locking is performed when the data set is accessed, provided the data set is not already locked. If the data set has already been locked by an earlier access, the *NOLOCK argument has no effect. The *NOLOCK also has no effect on locks set by the DM IMAGE *LOCK command. Refer to Data Set Locking for further information.

index_name

This parameter allows you to specify the name of the index or the number of the index to be used to access the data set. The index name must be the name of a field specification that you have defined as a key field on the default record

layout for the data set. You can also specify the index by entering the number of an index for the data set.

The index you specify here is used for one data set access only. Specifying an index name or number here does not change the current value in the communication area field *INDEXNO.

If you don't specify an index name or number, HP ALLBASE/4GL accesses the data set according to the index specified by the current value of *INDEXNO.

key

This parameter defines the key data to be used to access the data set. It can be one of the following:

- A literal.
- An alphanumeric constant.
- A variable or calculated item.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- A non-numeric communication area field.

This parameter is optional. If you don't specify a key, HP ALLBASE/4GL accesses the data set according to the existing value in the record buffer key field identified by the current value of *INDEXNO or the *INDEX=argument. If you do specify a key value, HP ALLBASE/4GL moves this value to the key field (as identified by *INDEXNO or the *INDEX=argument) in the file record buffer.

command

A command to be executed if the requested data set operation fails. You can use any of the following HP ALLBASE/4GL logic commands:

- ENTER
- EXIT

12-96 Logic Commands

- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

recno

HP ALLBASE/4GL allows you to access an HP TurboIMAGE/iX data set record by specifying the number of the record. Record number 1 is the first physical record in the data set.

You can specify *recno* with any of the following items containing a non-negative integer number.

- A whole number.
- A numeric constant.
- A variable or calculated item.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- A communication area field.

Using FILE *READ with the *REC = *recno* option performs the same task as the HP TurboIMAGE/iX library procedure DBGET in mode 4 (directed read).

Description

The FILE command performs various operations on HP TurboIMAGE/iX data sets.

Opening and Closing Data Sets

HP ALLBASE/4GL automatically opens an HP TurboIMAGE/iX data set when it is first read, written to, or locked, in an HP ALLBASE/4GL process. There is no need to explicitly open an HP TurboIMAGE/iX data set. If a data set is specifically closed (refer to *CLOSE below), HP ALLBASE/4GL automatically reopens the data set when it is next accessed.

HP ALLBASE/4GL automatically closes all HP TurboIMAGE/iX data sets at the start of a process. A data set can also be specifically closed using the FILE *CLOSE command, which is described later. All data sets in a database, that is, the whole database, can be closed at once using the DM IMAGE *CLOSE command.

File Indexes

The communication area field *INDEXNO specifies the number of the index used to access the data set. HP ALLBASE/4GL initializes the value in this field to 1 when the application starts, assuming that all data set access is via the data set primary index. HP ALLBASE/4GL does not reset the value of *INDEXNO if you change it at any time. Therefore, you must ensure that *INDEXNO is set to the correct value before each FILE command is executed.

If you want to access records in a data set with more than one index, you must set *INDEXNO to the number of the required data set index before the FILE command is executed.

You can specify a different index for a particular data set access. To do so, specify the name of a field defined as a key field on the default record layout for the data set in the *INDEX= argument on the FILE command. You can also specify the index for a data set access by specifying the number of an index in the *INDEX=clause.

File Access Modes

The DM IMAGE *MODE command and the MODE command work in conjunction to set the access mode for the database and data sets.

12-98 Logic Commands

You must ensure that the access mode for the database is set appropriately with the DM IMAGE *MODE command. If no DM IMAGE *MODE is specified, the default mode allows shared read access (with locks) to all data sets within a database.

You must also specify the data set access modes for any data sets that are to be accessed in the current HP ALLBASE/4GL process by using the MODE command with the *READ, *WRITE, *LOCK, or *UNLOCK argument. (Refer to the MODE command.) If you do not specify a data set access mode, the default mode *READ is assumed.

The capabilities specified in the MODE command must be a subset of the capabilities specified in the DM IMAGE *MODE command. (Refer to the DM IMAGE *MODE command for more information.)

File Record Buffers and Position Pointers

HP ALLBASE/4GL maintains a separate buffer for each record layout defined for each data set. HP ALLBASE/4GL clears the data set record buffer or buffers automatically when a data set is first accessed, or at the start of any process.

Where multiple record layouts have been defined for an HP TurboIMAGE/iX data set, data in one record buffer is not automatically available to another record buffer for the same data set. To make the data available, you can read the record again, or move the data between buffers using either the MOVE or MOVELOOP commands. This will only work if both record layouts have the same structure.

HP ALLBASE/4GL also maintains a data set position pointer which contains the address of the current record.

Communication Area Fields

HP ALLBASE/4GL places the external name of the data set currently being accessed in the communication area field *FILENAME}.

File Operations

Operations you can perform on HP TurboIMAGE/iX data sets are:

***BUFFER.** Clears the specified buffer of all data. The buffer cleared is defined by *file_ref*. Unless the data set has been opened in mode *UNLOCK, clearing the data set record buffer unlocks the current data set record.

***CLOSE.** Closes the nominated data set. You can use this operation to close any data set that is not required. HP ALLBASE/4GL automatically reopens the data set if it is accessed after being closed. Data sets are automatically closed when the process exits. They are also closed when the entire database is closed with the DM IMAGE *CLOSE command.

***DELETE.** Deletes a record from the data set.

Detail data set records must be read before they can be deleted. If you haven't read a record, HP ALLBASE/4GL returns an error number.

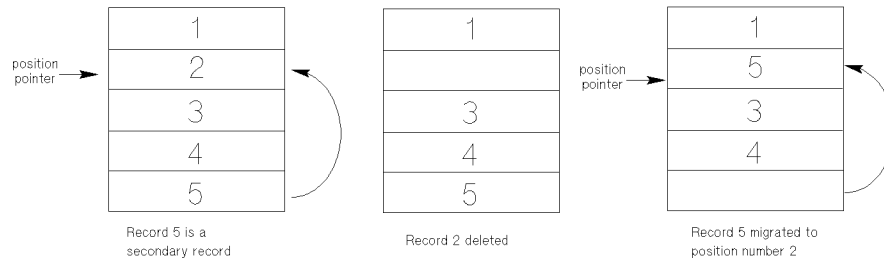
Master data sets act as indexes to detail data sets, so you cannot delete a master data set record if any detail data set records contain the key value of the master data set record.

When a manual master data set record is deleted, it may cause a secondary record in the data set to migrate to the position of the deleted record. If the next serial operation were to move the data set position pointer to the next or previous record, the record migrating to the position of the deleted record would not be read. To avoid this, HP ALLBASE/4GL sets an indicator so that the next serial operation re-establishes the data set position pointer on the same record. For more information about migrating secondary records, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

The following diagram illustrates backward migration. Forward migration is similar.

12-100 Logic Commands

Backward Migration



Automatic master data set records are deleted automatically if all the detail data set records linked to it are deleted.

***FIND**

For detail data sets, *FIND finds the first record whose key is equal to the value of *key*. This operation is used to place the data set position pointer at the beginning of a chain of detail data set records with the same key value. After the *FIND operation, *NEXT and *PREVIOUS operations will move the pointer along the chain rather than serially through the records.

Master data sets must contain unique keys, so chains of records with identical key values do not exist. To find the record in a master data set with a certain key value, *FIND mimics the operation of *READ *KEY = *key*.

*FIND only places the data set position pointer; to move the data into the record buffer, use the FILE *NEXT operation. Because HP ALLBASE/4GL maintains only one position pointer for each data set, you must use the *NEXT operation before you perform any other operation on the same data set.

The *FIND operation returns error status 19111 (record not found) if there is no record with a key equal to the value of *key*.

***FIRST.** Reads the first record for the data set.

If *KEY is not specified, the first record in serial order is read. If *KEY is specified, the first record with that key value is read.

***INSERT.** Adds a new record to the data set.

This operation adds new records to detail data sets. HP TurboIMAGE/iX also adds a record to any automatic master data sets linked to the detail data set.

This operation causes an error if a manual master data set linked to the detail data set does not contain a record with the key value of the detail data set being added.

Using *INSERT with master data sets causes an error if you attempt to add a record to the data set with a primary key value the same as one that already exists.

***LAST.** Reads the last record for the data set.

If *KEY is not specified, the last record in serial order is read. If *KEY is specified, the last record with that key value is read.

***MODIFY.** Changes an existing record in the data set. Automatic master data sets linked to a detail data set that is modified are automatically updated.

You must read a record before you can use the *MODIFY command to change its contents.

You can only alter a key value for a master data set if no detail data sets linked to the master data set have records containing the key value.

When a master data set record is modified and a key value is changed, the record may be moved to a new address and cause a secondary record to migrate to the position of the modified record. Because of this, do not use the FILE *NEXT or FILE *PREVIOUS command after modifying the key value of a master data set record. The record migrating to the old position of the modified record may not be read.

***NEXT.** Serially reads the next record in the data set.

For detail data sets, it is possible to read along a chain of records with the same key value. This will occur if the *NEXT operation is preceded by a *FIND operation, or a *READ operation with a key provided. If the *NEXT operation is executed when the data set position pointer is at the end of a chain, HP ALLBASE/4GL returns the error status 19115.

In some cases, for master data sets, *NEXT re-reads the current record. This occurs when a secondary master data set record has migrated backward to the position of a newly deleted record. If *NEXT was to move the data set position pointer to the next record before the read, the record migrating

12-102 Logic Commands

backward to the position of the deleted record would not be read. To avoid this, HP ALLBASE/4GL sets an indicator so that *NEXT re-establishes the data set position pointer at the same position before the read. For further information about migrating secondary records, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

With master data sets, do not use a FILE *NEXT command after a FILE *MODIFY command that modifies a key value. The record migrating to the old position of the modified record may not be read.

***PREVIOUS.** Serially reads the previous record in the data set.

For detail data sets, it is possible to read along a chain of records with the same key value. This will occur if the *PREVIOUS operation is preceded by a *FIND operation, or a *READ operation with a key provided. *PREVIOUS reads the previous record in the chain. If the *PREVIOUS operation is executed when the data set position pointer is at the start of a chain, HP ALLBASE/4GL returns the error status 19115.

In some cases, for master data sets, *PREVIOUS re-reads the current record. This occurs when a secondary record has migrated forward to the position of a newly deleted record. If *PREVIOUS was to move the data set position pointer to the next record before the read, the record migrating forward to the position of the deleted record would not be read. To avoid this, HP ALLBASE/4GL sets an indicator so that *PREVIOUS re-establishes the data set position pointer at the same position before the read. For further information about migrating secondary records, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

With master data sets, do not use a FILE *PREVIOUS command after a FILE *MODIFY command that modifies a key value. The record migrating to the old position of the modified record may not be read.

***READ.** If a key value is specified, *READ reads the first record in the data set whose key value exactly matches the value specified by *KEY = *key*. If you don't specify the *KEY = *key* argument in the FILE command, HP ALLBASE/4GL uses the current value in the record buffer field specified by *INDEXNO or the *INDEX = argument.

If you include the *REC = *recno* argument, *READ reads the specified record number. If no record exists at that address, HP ALLBASE/4GL returns an error status 19111.

In some cases, for master data sets, *READ re-reads the current record. This occurs when a secondary record has migrated backward to the position of a newly deleted record. If *READ was to move the data set position pointer to the next record before the read, the record migrating backward to the position of the deleted record would not be read. To avoid this, HP ALLBASE/4GL sets an indicator so that *READ re-establishes the data set position pointer at the same position before the read. For further information about migrating secondary records, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

***UNLOCK.** The *UNLOCK argument releases the locks on a data set that has been accessed under MODE *WRITE or MODE *UNLOCK. HP ALLBASE/4GL ignores this argument if the data set is being accessed under MODE *READ or MODE *LOCK.

Note

If you unlock a data set, you release ALL locks held on the database that contains the data set specified in the *UNLOCK operation.

***WRITE.** Writes a new record or modifies an existing record within a manual master data set.

You cannot use this operation for automatic master data sets because these are automatically maintained by HP TurboIMAGE/iX.

Don't use this operation for detail data sets, which allow duplicate primary keys. Use the *INSERT operation instead.

When you use this operation for manual master data sets, one of the following actions occurs:

- If a record exists on the data set whose primary key matches the primary key value in the record buffer, the data set record is modified. The modify is performed in the same manner as the *MODIFY operation.
- If a record does not exist, a new record is added to the data set. The addition is performed in the same manner as the *INSERT operation.

*WRITE performs the same operation as the UPDATE command except that it operates on one data set only.

12-104 Logic Commands

Error Handling

HP ALLBASE/4GL executes an optional logic command specified in the FILE command if an error occurs as a result of the data set access. For example, when performing a *NEXT operation, the optional command is executed if the end of the data set is encountered.

When an error occurs, the HP ALLBASE/4GL data manager writes an error code into the communication area field *IOSTATUS. Appendix C lists the data manager error codes. The value returned to *IOSTATUS is zero if the data set command is successful.

If you include the optional error command, no data manager errors are displayed on the terminal screen.

The following table summarizes the most common file error values returned to the communication area field *IOSTATUS. Refer to appendix C for a list of file error returns.

*IOSTATUS

Value	Condition
19000	MPE/iX file access error detected by HP TurboIMAGE/iX.
19013	MPE/iX error. Access permission denied to file.
19100	Duplicate primary key error. The file only allows unique primary key values, and a record with the same primary key value already exists.
19107	Record locked error. Another user or process has locked the record you are attempting to access.
19110	Beginning or end of file reached during a *NEXT or *PREVIOUS file operation.
19111	Record not found. No record has a key value matching the value specified for *KEY= <i>key</i> .
19112	Record not read. No record has been read for the file.
19113	Exclusive locking error. Another user or process has locked the file.

- 19115 Beginning or end of detail data set chain reached during a *NEXT or *PREVIOUS file operation.
- 19130 File locking error. The current process cannot lock the file because the file is already locked by another process.

In addition to HP ALLBASE/4GL error numbers, each HP TurboIMAGE/iX library procedure returns a status array. This is recorded in the *IMSTAT communication area field. The first halfword (16 bits) of the status array contains the HP TurboIMAGE/iX error number, and this error number is also returned to the *IOSTATDM communication area field.

The meaning of status arrays and error numbers returned by HP TurboIMAGE/iX library procedures can be found in the *HP TurboIMAGE/iX Database Management System Reference Manual*. Refer to chapter 10 of this manual for a list of the common library procedures initiated by HP ALLBASE/4GL.

The communication area field *ERRORDM contains the text of any error messages returned from HP TurboIMAGE/iX.

Data Set Locking

You may control data set locking with the DM IMAGE *LOCK and *UNLOCK commands. This allows you to make the best use of HP TurboIMAGE/iX locking capabilities, and is recommended for applications with multiple users. If you do not use these commands, HP ALLBASE/4GL uses implicit locking. The type of implicit locking used is determined by the MODE command.

HP TurboIMAGE/iX provides three levels of explicit locking; database locking, data set locking, and record locking. All three levels of locking are provided by the DM IMAGE *LOCK command. To delete or insert detail data set records, only record level locking is required. To delete or insert master data set records, use data set level locking. For updating master data set records, record level locks are adequate.

12-106 Logic Commands

Implicit File Locking

Implicit data set locking in HP ALLBASE/4GL depends on both the data set operation used with the FILE command and the data set access mode specified by the MODE command.

MODE *WRITE should be used with care. It does not guarantee any transaction integrity, but can be used by developers without significant knowledge of HP TurboIMAGE/iX locking. When using MODE *WRITE, a data set level lock is placed.

The record remains locked to the current process until another record is accessed through the same file buffer, the buffer is cleared, the file is closed, or a FILE *UNLOCK command is executed.

Note

When a lock is released, all locks held on the database are released.

MODE *UNLOCK is similar to MODE *WRITE, except that a lock is not released until a specific FILE *UNLOCK command is executed. If a data set is locked, another user or process cannot lock the same data set while the lock is current.

The command MODE *LOCK specifies that the process has write access to the file. Each file is locked when it is first accessed. The file or files remain locked until the end of the process, or the files are closed.

While a file is locked to a process that uses the *LOCK form of the MODE command, another process or user can read the file provided that the file is accessed under MODE *READ. However, while the MODE *LOCK command is in effect, other processes or users cannot write to the file or lock the file.

The following table summarizes the data set access and locking that can occur if two users (or a user and a background process) attempt to access the same data set.

	Existing Access Mode			
	*LOCK	*WRITE	*READ	*UNLOCK
Attempted access mode	Implemented on first access	File locked on read or write		File locked on read or write
*LOCK	Not permitted	Not Permitted	Permitted	Not permitted
*WRITE Read or write operation	Not permitted	Permitted	Permitted	Permitted
*READ	Permitted	Permitted	Permitted	Permitted
*UNLOCK Read or write operation	Not permitted	Permitted	Permitted	Permitted

Under the MODE ***LOCK**, MODE ***WRITE**, or MODE ***UNLOCK** commands, the data set is normally locked when it is first accessed. However, the data set is not locked if you access the data set using a command that contains the ***NOLOCK** argument. Any read or write access that does not use the ***NOLOCK** argument will lock the data set. It remains locked until one of the following occurs:

- The process ends.
- The data set is closed with the FILE ***CLOSE** command.
- Any data set from the same database is unlocked with the FILE ***UNLOCK** command within the current process.

Data set locks are also released by a DM IMAGE ***CLOSE** command or a TRANSACT ***END** command.

When a data set is accessed under MODE ***READ**, no data set locking is performed.

At any time, you can explicitly specify that a particular data set access does not lock a data set by specifying the ***NOLOCK** argument on the FILE command.

12-108 Logic Commands

***LOCKWAIT Communication Area Field**

HP ALLBASE/4GL uses the value in the communication area field *LOCKWAIT to determine its behavior when it attempts to access a data set that is locked by another process. The *LOCKWAIT field can contain the values -1, 0, or a positive integer number. The actions taken by HP ALLBASE/4GL for different values in *LOCKWAIT are summarized below.

Value	Action
+n	HP ALLBASE/4GL attempts to access the data set every second for <i>n</i> seconds. If the data set is not accessible after <i>n</i> seconds, HP ALLBASE/4GL returns an error number 19107.
0	HP ALLBASE/4GL attempts to access the data set every second for 5 seconds. If the data set is not accessible after 5 seconds, HP ALLBASE/4GL returns an error number 19107.
-1	HP ALLBASE/4GL waits indefinitely to access locked data sets.

When HP ALLBASE/4GL starts an application, the value in *LOCKWAIT is initialized to zero. You can use the MOVE command to move a different value into *LOCKWAIT. HP ALLBASE/4GL does not reset the value in *LOCKWAIT at any time except when the application starts.

Example 1

```
5 FILE *READ stock *KEY=S-stock_no.stock_add; ENTER 12
.
.
12 MESSAGE no_stock
```

This command reads the data set *stock*, and searches for a record with its key value equal to the value of the screen field *stock_no* on the screen *stock_add*. If the record cannot be located, the system executes step 12, which displays the message *no_stock*.

Example 2

```
FILE *BUFFER trans.cust
```

This command clears the buffer for the record *cust* in the data set *trans*.

Example 3

```
4 FILE *NEXT trans; ENTER 11
5 SHOW *REFRESH
6 TOP
.
.
11 MESSAGE no_transactions
```

These commands read and display records from the data set *trans*. When the system encounters the end of the data set, it executes step 11 which displays the message *no_transaction*.

Example 4

```
FILE *WRITE stock *KEY=*S01
```

This command checks if a record with a key equal to the contents of screen field number 1 exists on the data set *stock*. If such a record exists, the system updates it. If the record doesn't exist, the system creates a new record.

Example 5

```
1 DM IMAGE *MODE *MODLOCK :D-customer_db
2 MODE *WRITE customer
3 DM IMAGE *LOCK :D-customer_db :R-customer
.
5 FILE *WRITE customer *KEY=*; ENTER 8
6 FILE *CLOSE customer
7 EXIT
8 MESSAGE customer_errmsg
9 EXIT
```

In this logic block, the database *customer_db* is opened in a mode that allows shared modify access to the database if data sets are locked. The data set access mode is set to **WRITE* in step 2, and step 3 locks the *customer* data set. If step 5 executes correctly, the data set closes and all locks on the database are released. If an error occurs, the data set remains open until the end of the current process.

The database is not closed until the end of the HP ALLBASE/4GL session, or until the *DM IMAGE *CLOSE* logic command is issued.

12-110 Logic Commands

IF Command

The IF command conditionally executes some logic commands as a result of testing one condition or a combination of two conditions.

Formats

$$\text{IF } condition_1 \left[\left\{ \begin{array}{l} \& \\ | \end{array} \right\} condition_2 \right] \text{ THEN command(s) ELSE command(s)}$$

Where & - AND, and | - OR

and *condition_1* and *condition_2* can be:

operand relational_operator *operand*

operand *BLANK

operand *ALPHA

operand *NULL

operand *NUMERIC

switch_name *ON

switch_name *OFF

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command IF				
Condition 1	Data Name 1	IF Test	Data Name 2				
AND/OR	(&,)						
Condition 2							

Command Help	Function Header	Process Detail	Generate Function	System Keys	Commit Data	Help	Previous Menu

Parameters

condition_1, condition_2

The condition being tested. The condition may be testing the relationship between two operands, the content of an operand, or the status of a switch.

operand

The item being compared. This item may be compared with another item, or its content may be tested for a certain status. The operand may be one of the following:

- Whole number.
- Literal.
- Numeric or alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area field.

Operand Content Test.

Tests the contents of one operand. The test can be one of the following:

Test Name	Meaning
*BLANK	Is operand all spaces?
*ALPHA	Is operand all letters? (Leading and trailing spaces are ignored)
*NULL	Does operand contain a null value? This test is only valid if the operand is a field on a record layout for an HP ALLBASE/SQL table or a select list.
*NUMERIC	Does operand meet N or S edit code requirements?

You cannot use these tests if the operand is a literal.

12-112 Logic Commands

relational_operator

Tests the relationship between two operands. The relational operator may be one of the following:

Relational Operator	Meaning
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

When you use a relational operator to compare two operands, the edit code of the operands determines the nature of the relational comparison. If you compare two numeric fields, the comparison is made on strict numeric order. If you compare two alphanumeric fields, or compare an alphanumeric field against a literal, the comparison is made on a character-by-character basis using the system collating sequence (refer to appendix A).

Note



If you use a relational operator to compare two fields, make sure that both fields have the same edit code, or contain similar types of data.

switch_name

The name of the switch to be tested. The switch name can be one of the following:

- A number in the range 1 to 8, representing one of the eight user switches.
- *SHOWING.
- *ENTERED.
- *BYPASS.
- *ENDLINE.

- *MOREREC.

The status of the switch may be *ON or *OFF.

command

May be any logic block commands except:

- IF.
- IFLOOP.
- SELECT.

If you use more than one command, separate the commands with semicolons.

Some logic commands allow you to specify an optional command to be executed if an error condition occurs. These are the CALC, DATE, FILE, MATH, MATHLOOP and VALIDATE commands. If you use one of these commands in the *THEN* command list or the *ELSE* command list of an IF command, you must include the error condition command. If you don't, HP ALLBASE/4GL interprets the next command in the command list as the error command, and will only execute it if an error occurs in the previous command.

Description

The IF command tests one or two conditions and then performs one or more commands, depending upon the result of the conditional test. For repeatedly performing a conditional test, use the IFLOOP command.

The IF command can test the following conditions:

- The status of a switch.
- The contents of an operand.
- The relationship between two operands according to a relational operator.

You can test more complex combinations of conditions with a decision table. Refer to chapter 11 for more information about decision tables.

When HP ALLBASE/4GL is comparing two operands of unequal length according to a relational operator, it extends the shorter of the two operands with spaces until it is the same length as the other operand.

The IF command tests if the condition is true or false:

12-114 Logic Commands

- If the condition is true, the *THEN* command(s) are executed.
- If the condition is false, the *ELSE* command(s) are executed if you have specified any. The argument *ELSE* and its associated commands are optional. If you don't specify any *ELSE* arguments, no commands are executed if the condition is false.

The IF command can test a compound condition consisting of two different conditions. In this case, a Boolean or logical AND or logical OR operator links the two conditions. The system tests the compound condition as one condition.

The following statements illustrate the use of the AND/OR logic:

condition_1 **AND** condition_2

In this situation **both** *condition_1* **and** *condition_2* must be true before the compound condition is true. If either condition is false, then the compound condition is false.

condition_1 **OR** condition_2

In this situation at least one of *condition_1* or *condition_2* must be true before the compound condition is true. If both conditions are false, then the compound condition is false.

HP ALLBASE/4GL uses the following symbols to specify the AND or OR logical connectives:

Symbol	Meaning
&	AND
	OR

Example 1

```
IF *ENTERED *OFF THEN CLEAR *S; EXIT
```

This command tests the status of the *ENTERED switch. If *ENTERED is *off* then HP ALLBASE/4GL clears all screen fields and then exits the logic block. If *ENTERED is *on*, HP ALLBASE/4GL executes the next logic command.

Example 2

```
IF * = "A" | * = "C" THEN ZIP ELSE VISIT delete_rec
```

This command checks the contents of the current screen field. If the field contains either the letter A or C, no action is taken. If the field contains neither of these letters, HP ALLBASE/4GL executes the function *delete_rec*. In either case, HP ALLBASE/4GL then executes the next command after the IF command.

Example 3

IF V-code &<> "NEW" THEN **SERIES** 10 20; **ENTER** 1

This command tests the content of the variable *code*. If the variable doesn't contain the string *NEW*, HP ALLBASE/4GL executes command step numbers 10 through 20. Processing then returns to step 1.

IFLOOP Command

The IFLOOP command performs a conditional IF test repeatedly.

Format

IFLOOP *loop_count* *condition* THEN *command(s)* [ELSE *command(s)*]

where *condition* may be:

- *operand step_factor* relational_operator *operand step_factor*
- *operand step_factor* *BLANK
- *operand step_factor* *ALPHA
- *operand step_factor* *NULL
- *operand step_factor* *NUMERIC
- *switch_name* 0 *ON
- *switch_name* 0 *OFF

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	IFLOOP
Execute Loop	Times			
Data Name 1			Step Factor	
Comparison Test				
Data Name 2			Step Factor	

Command	Function	Process	Generate	System	Commit	Help	Previous
Help	Header	Detail	Function	Keys	Data		Menu

Parameters

loop_count

A number specifying the number of times the test is performed.

condition

The condition being tested.

operand

The item being compared or tested. This item may be compared with another item, or its content may be tested for a certain status. The operand may be one of the following:

- Whole number.
- Literal.
- Numeric or alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area field.

Operand Content Test.

Tests the contents of one operand. The test can be one of the following:

Test Name	Meaning
*BLANK	Is operand all spaces?
*ALPHA	Is operand all letters? (Leading and trailing spaces are ignored.)
*NULL	Does operand contain a null value? This test is only valid if the operand is a field on a record layout for an HP ALLBASE/SQL table or a select list.
*NUMERIC	Does operand meet N or S edit code requirements?

You cannot use these tests if the operand is a literal.

step_factor

A number specifying the increment to be applied to the operand name to obtain the operand to be tested on the next loop of the command.

The step factor may be a positive or negative number, or zero.

12-118 Logic Commands

The increment is applied to the operand name, and not to its value. This enables the IFLOOP command to test successive fields on a screen or file record.

The step factor may be zero. A zero step factor causes the IFLOOP command to test the same operand each time. You must use a zero step factor for the following:

- Numeric or alphanumeric constant.
- Variable or calculated item.
- Literal or number.
- Switch.

relational_operator

Tests the relationship between two operands. The relational operator may be one of the following:

Relational Operator	Meaning
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

When you use a relational operator to compare two operands, the edit code of the operands determines the nature of the relational comparison. If you compare two numeric fields, the comparison is made on strict numeric order. If you compare two alphanumeric fields, or compare an alphanumeric field against a literal, the comparison is made on a character-by-character basis using the system collating sequence (refer to appendix A).

Note

If you use a relational operator to compare two fields, make sure that both fields have the same edit code, or contain similar types of data.

switch_name

A reference to a system or user switch. This may be one of the following:

- A number between 1 and 8 representing one of the user switches.
- *BYPASS
- *ENDLINE
- *ENTERED
- *SHOWING
- *MOREREC

You must use a step factor of 0 with switches.

The status of the switch may be *ON or *OFF.

commands

May be any logic commands except:

- IF
- IFLOOP
- SELECT

If you use more than one command, separate the commands with semicolons.

Some logic commands allow you to specify an optional command to be executed if an error condition occurs. These are the *CALC*, *DATE*, *FILE*, *MATH*, *MATHLOOP* and *VALIDATE* commands. If you use one of these commands in the *THEN* command list or the *ELSE* command list of an *IFLOOP* command, you must include the error condition command. If you don't, HP ALLBASE/4GL interprets the next command in the command list as the error command, and will only execute it if an error occurs in the previous command.

12-120 Logic Commands

Description

The IFLOOP command repeatedly performs a conditional test. The command operates in a similar manner to the IF command except that the system repeats the conditional test for each occurrence of the *loop_count*.

In the IFLOOP command, each operand within a condition incorporates a step factor. The step factor determines, on each successive iteration of the loop, the next operand to be tested. If an operand is a file record field reference or a work area field reference, HP ALLBASE/4GL steps through the fields on the record(s) according to the step factor. HP ALLBASE/4GL will step through multiple occurrences of fields defined as having more than one occurrence in the dictionary.

The IFLOOP command can test the following conditions:

- Status of a switch.
- Contents of an operand.
- Relationship between two operands according to a relational operator.

You can test combinations of conditions with a decision table. Refer to the DECISION command for further details.

When the system is comparing two operands of unequal length according to a relational operator, it extends the shorter of the two operands with the spaces until it is the same length as the other operand.

The IFLOOP command is typically used to:

- Test a number of consecutive fields.
- Repeatedly test a switch.

If one of the commands after the THEN or ELSE is PROCEED, TOP, ENTER, or EXIT, the IFLOOP command terminates when the test reaches the specified condition. That is, the IFLOOP command may not be executed for the full *loop_count* number of times.

Example 1

```
IFLOOP 10 1 0 *OFF THEN VISIT update_data
ELSE VISIT delete_data
```

This command tests the status of user switch number 1 ten times (with a step factor of 0). If the switch is *off*, HP ALLBASE/4GL executes the function *update_data*. Otherwise, HP ALLBASE/4GL executes the function *delete_data*. The functions can set or reset switch number 1 as part of their logic.

Example 2

```
IFLOOP 7 S-acc_no.updt_acc 1 &<> F-account.act_mast 1
THEN MOVE "CHANGED" V-status; ENTER 12
```

This command checks (at most) the seven fields on the screen *updt_acc*, starting with *acc_no*, against the seven fields on the record *act_mast*, starting with *account*. If there is any difference, the system sets the value of the variable *status* to CHANGED and then executes step 12.

Example 3

```
IFLOOP 10 *S01 1 *NUMERIC THEN VISIT numeric_field
ELSE VISIT alpha_field
```

This command checks the first 10 fields of the current screen and for each field, executes the function *numeric_field* if the field is numeric. If the field is not numeric, the function *alpha_field* is executed.

KEYS Command

The KEYS command displays a set of function keys on the current screen.

Formats

$$\text{KEYS } \left\{ \begin{array}{l} \textit{keys_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

keys_name

The name of the set of function keys that you want to display on the screen.

data_ref

One of the following, containing the name of the set of function keys that you want to display on the screen:

- A master title name.
- An alphanumeric constant.
- A variable or calculated item.
- A screen field reference.
- A scratch-pad field name.
- A file record field reference.
- A work area field reference.

Description

The KEYS command allows you to display a set of function keys on the current screen. You may use the command to display a different set of function keys from those originally defined for the screen. If HP ALLBASE/4GL cannot locate a set of function keys with the specified name, a message is displayed and the existing function keys remain unchanged.

Example

KEYS C-update_keys

This command displays the set of function keys whose name is contained in the alphanumeric constant *update_keys*.

LENGTH Command

The LENGTH command calculates the number of characters in a field.

Format

LENGTH *data_ref result_data_ref*

Parameters

data_ref

The field whose length is to be calculated. It can be one of the following:

- A master title name.
- A variable or calculated item.
- A screen field reference.
- A named scratch-pad field reference.
- A file record field reference.
- A work area field reference.

result_data_ref

The field into which the length of *data_ref* is placed. It can be one of the following:

- A variable.
- A screen field name.
- A named scratch-pad field name.
- A file record field reference.
- A work area field reference.

Description

The LENGTH command calculates the length in characters of the contents of *data_ref*. This length is placed in *result_data_ref*.

The LENGTH command ignores leading zeros on numeric fields and trailing spaces on non-numeric fields.

Example

LENGTH * V-length

This command determines the number of characters in the current screen field and places the result into the variable *length*.

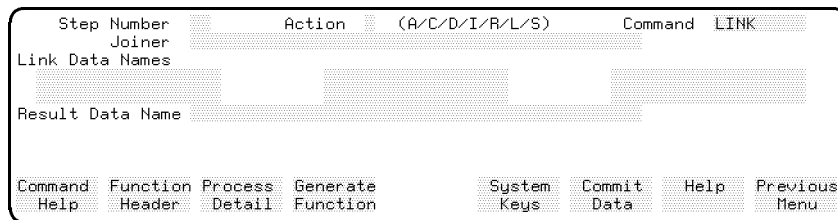
LINK Command

The LINK command concatenates a number of fields, and optionally separates them with a common literal.

Formats

LINK [*JOINER=*joiner*] *link_count link_data_ref ... result_data_ref*

Window



Parameters

link_count

A number greater than or equal to 2 that specifies the number of fields to be linked together.

link_data_ref

The fields to be concatenated, each of which can be one of the following:

- A literal.
- A numeric or alphanumeric constant.
- A variable or calculated item.
- A master title name.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.

- A communication area field.

result_data_ref

The field where the result of the link is placed. It can be one of the following:

- A variable.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- The communication area field *PASS.

joiner

The data used to separate each of the *link* fields. The joiner is optional, but if you specify it, you must precede it with the argument *JOINER=.

Enter the joiner into the *Joiner* field on the window.

The joiner can be one of the following:

- A literal.
- A numeric or alphanumeric constant.
- A variable or calculated item.
- A master title name.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- A communication area field.

12-128 Logic Commands

Description

The LINK command concatenates (links or joins together) specified fields. Use the LINKLOOP command to concatenate a series of fields.

You can specify an optional separator or joiner which is placed between each of the concatenated fields.

You must specify the count of the number of fields being concatenated. If you're using the window, HP ALLBASE/4GL supplies this count automatically and places it into the appropriate position in the command.

The LINK command links each *link_data_ref* without any modification. That is, blanks (if any) are not removed during the concatenation process into the *result_data_ref*.

The optional joiner, if specified, separates each *link_data_ref* in the *result_data_ref* field. If you specify a joiner, it must be immediately preceded by the command argument *JOINER=.

Example 1

```
LINK 2 V-P_Number S-Opt_Number.Opt_Lst *PASS
FILE *READ option *KEY=*PASS
```

This command links the contents of the variable *P_Number* with the contents of the field *Opt_Number* on the screen *Opt_Lst* and places the result in the communications area field **PASS*. The FILE command then uses this result as a key to access the file *option*.

Example 2

```
LINK *JOINER = "/" 3 * *+1 *+2 F-date.stock
```

This command concatenates the current and the next two screen fields. The command separates each field by a slash and places the result into the field *date* on the file *stock*. This operation could be picking up three fields on the screen, representing the month, day, and year, formatting them as a date and then saving the date in the file buffer.

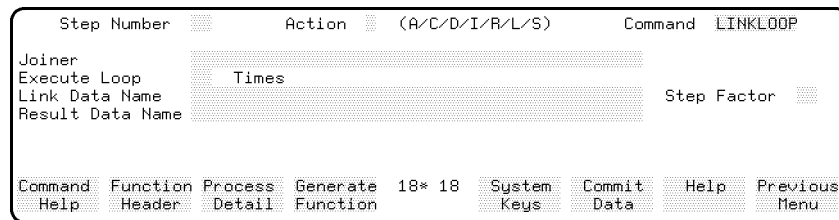
LINKLOOP Command

The LINKLOOP command concatenates a series of fields.

Formats

LINKLOOP [*JOINER=*joiner*] *loop_count link_data_ref step_factor result_data_ref*

Window



Parameters

loop_count

A number specifying the number of fields to be concatenated. The loop count must be a value greater than or equal to 2.

link_data_ref

The name of the first field to be linked in the loop. Enter one of the following:

- A literal.
- A numeric or alphanumeric constant.
- A variable or calculated item.
- A master title name.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.

12-130 Logic Commands

- A communication area field.

step_factor

A number specifying the increment applied to the *link_data_ref* name to obtain the next data reference which is linked on each successive loop of the command.

The step factor may be a positive or negative number.

The increment is applied to the name and not to its value. This enables successive, but not necessarily consecutive, fields to be linked.

You must use a zero step factor if the *link_data_ref* is a constant, a literal, a master title, a variable, a calculated item, or a communication area field other than *COUNT(n), *CROSS(n), or *TOTALS(n).

result_data_ref

The field where the result of the link is placed. Enter one of the following:

- A variable.
- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- The communication area field *PASS.

joiner

The data used to separate each of the *link_count* fields. The joiner is optional but, if specified, must be preceded by the argument *JOINER=.

Enter the joiner into the *Joiner* field on the window.

The joiner may be one of the following:

- A literal.
- A numeric or alphanumeric constant.
- A variable or calculated item.
- A master title name.

- A screen field reference.
- A scratch-pad field reference.
- A file record field reference.
- A work area field reference.
- A communication area field.

Description

The LINKLOOP command concatenates (links or joins together) successive fields.

The LINKLOOP command operates in a similar manner to the LINK command in the way it concatenates fields. Unlike the LINK command however, you don't need to specify each field to be concatenated explicitly. You only need to specify the first field.

The *loop_count* specifies the number of fields starting from and including the *link_data_ref* to be concatenated in the *result_data_ref*. The loop-count must be greater than or equal to 2. That is, at least two fields must be linked together.

The *step_factor* determines, on each successive iteration of the loop, the next field to be linked. If the *link_data_ref* is a file record field reference or a work area field reference, HP ALLBASE/4GL links successive fields on the record depending on the step factor. HP ALLBASE/4GL also steps through the multiple occurrences of fields defined as having multiple occurrences in the dictionary.

Example 1

```
LINKLOOP *JOINER="/" 3 * 1 F-date.stock
```

This command concatenates the current screen field and the next two screen fields. The command separates each field with the character /, and places the result into the field *date* on the file *stock*. This operation could be picking up three fields on the screen, representing the month, day, and year, formatting them as a date, and then saving the date in the file buffer.

12-132 Logic Commands

Example 2**LINKLOOP 4 *S03 2 *P01**

This command links four fields, starting with screen field number 3 on the current screen. The command links every second field (that is, fields *S03, *S05, *S07, and *S09) and places the result into scratch-pad field *P01.

MATH Command

The MATH command performs a simple arithmetic operation.

Format

MATH *operand_1* **operator** *operand_2* = *result_data_ref* [*;command*]

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	MATH
Data Name 1				
Operator	(+,-,x,/,M,e)			
Data Name 2				
Result Data Name				
On Error				
Command Help	Function Header	Process Detail	Generate Function	System Keys
				Commit Data
				Help
				Previous Menu

Parameters

operand_1, operand_2

The items on which the arithmetic operation is performed. The operands can be any of the following:

- Whole number.
- Numeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- *TOTALS(n) where n = 1 to 16.
- *COUNT(n), *CROSS(n): where n = 1 to 5.

12-134 Logic Commands

Certain operands can be treated as absolute values. That is, their value is treated as a fixed positive number. To specify an operand as absolute, enter the character # after the operand name. You cannot specify the result of the calculation as absolute.

You can specify the following operands as absolute:

- Numeric variables.
- Numeric constants.
- Screen field names.
- Scratch-pad field names.
- File record field references.
- Work area field references.

operator

The operation to be performed on the two operands. An operator may be one of the following arithmetic symbols:

+	Add
-	Subtract
X or x	Multiply
/	Divide
M or m	Modulus
e	Raised to the power of

result_data_ref

This specifies where the result of the arithmetic operation is placed. It can be one of the following:

- Numeric variable.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.

- Work area field reference.
- *TOTALS(n) where n = 1 to 16.
- *COUNT(n), *CROSS(n): where n = 1 to 5.

command

A command to be executed if an error occurs while the MATH command is being executed. You can use any of the following commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

HP ALLBASE/4GL executes the command in the following conditions:

- Machine overflow or underflow occurs.
- Division by zero occurs.
- The result is too large to fit in the result field.
- One or more of the operands contains non-numeric data, or one of the operands contains a minus sign and has an N edit code.

If you don't specify an optional command, and one of these error conditions occurs, HP ALLBASE/4GL displays a warning message and then executes the next command in the logic block.

12-136 Logic Commands

Description

The MATH command performs a simple arithmetic operation.

Use the MATHLOOP command to perform a series of simple arithmetic operations.

HP ALLBASE/4GL uses full floating point arithmetic to perform the MATH command. The result is rounded off to suit the number of decimal places defined for the result data field. If the *result_data_ref* is a scratch-pad field, the number of decimal places used for the result of the calculation is the number of decimal places specified for the first operand of the MATH command.

Operands for the MATH command, and values computed by the MATH command can contain a maximum of 14 significant digits. Overflow or truncation errors may occur if you use values with more than 14 significant digits.

Example 1

```
MATH *COUNT(2) + 1 = *COUNT(2)}
```

This command increments the value in the communication area field *COUNT(2) by 1 and stores the result back in *COUNT(2).

Example 2

```
MATH V-principal# x P-rate = F-interest.bank
```

This command calculates simple interest. It uses the absolute value of the value in the variable *principal* and multiplies it by the contents of the scratch-pad field *rate*. The result is stored in the field *interest* in the file *bank*.

Example 3

```
MATH *S01 m 2 = V-remainder
```

This command checks if the value in screen field number 1 is odd or even. By specifying a modulus of 2, the number is divided by 2 and the remainder placed in the variable *remainder*. If the result of the modulus operation is a zero, the value in screen field number 1 is even; otherwise it is odd.

MATHLOOP Command

The MATHLOOP command repeatedly performs a simple arithmetic operation.

Formats

MATHLOOP *loop_count operand_1 step_factor_1 operator*
operand_2 step_factor_2 = result_data_ref step_factor_3 [:command]

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	MATHLOOP
Execute Loop	Times			
Data Name 1			Step Factor	
Operator	(+,-,x,/,M,e)			
Data Name 2			Step Factor	
Data Name 3			Step Factor	
On Error				
Command	Function	Process	Generate	System
Help	Header	Detail	Function	Keys
				Commit
				Help
				Previous
				Menu

Parameters

loop_count

A number specifying the number of times the arithmetic operation is performed.

operands

The items on which the arithmetic operation is performed. The operands can be any of the following:

- Whole number.
- Numeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.

12-138 Logic Commands

- Work area field reference.
- *TOTALS(n) where n = 1 to 16.
- *COUNT(n), CROSS(n): where n = 1 to 5.

Certain operands can be treated as absolute values. That is, their value is treated as a fixed positive number. To specify an operand as absolute, enter the character # after the operand name. You cannot specify the result of the calculation as absolute.

You can specify the following operands as absolute:

- Numeric variables.
- Numeric constants.
- Screen field names.
- Scratch-pad field names.
- File record field references.
- Work area field references.

operator

The operation to be performed on the two operands. An operator may be one of the following arithmetic symbols:

+	Add
-	Subtract
X or x	Multiply
/	Divide
M or m	Modulus
e	Raised to the power of

result_data_ref

This specifies where the result of the arithmetic operation is placed. It can be one of the following:

- Numeric variable.

- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- *TOTALS(n) where n = 1 to 16.
- *COUNT(n), *CROSS(n): where n = 1 to 5.

step_factor_1, step_factor_2, step_factor_3

The step factors are numbers specifying the increments to be applied to the operand names or the result data field to obtain the operands and result field for the next loop of the command.

The increments are applied to the operand and and result data field names, not to their values. This enables arithmetic operations to be performed on a series of operands, and the results of the calculations to be stored in a series of result data fields.

You can use different values for all three step factors in the MATHLOOP command if required.

The step factor may be a positive or negative number, or zero.

A zero step factor causes the MATHLOOP command to use the same operand or result data field each time in the arithmetic operation. You must use a zero step factor for the following operands:

- Number.
- Numeric constant.
- Numeric variable or calculated item.

command

A command to be executed if an error occurs while the MATHLOOP command is being executed. You can use any of the following commands:

- ENTER
- EXIT
- EXTERNAL

12-140 Logic Commands

- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

HP ALLBASE/4GL executes the command in the following conditions:

- Machine overflow or underflow occurs.
- Division by zero occurs.
- The result is too large to fit in the result field.
- One or more of the operands contains non-numeric data, or one of the operands contains a minus sign and has an N edit code.

If you don't specify an optional command and one of these error conditions occurs, HP ALLBASE/4GL displays a warning message and then executes the next command in the logic block.

Description

The MATHLOOP command repeatedly performs the same type of arithmetic operation on a series of operands. To perform a single arithmetic operation, use the MATH command.

The MATHLOOP command operates in a similar manner to the MATH command except that the operation is repeated for each occurrence of the loop-count. In the MATHLOOP command, each operand also incorporates a step factor.

The step factors determine the operands for each successive iteration of the loop. If one of the data references for the MATHLOOP command is a file record field reference or a work area field reference, HP ALLBASE/4GL steps through fields in the buffer according to the defined step factor. HP ALLBASE/4GL will step through successive occurrences of fields defined in the dictionary as having more than one occurrence.

HP ALLBASE/4GL uses full floating point arithmetic to perform the MATHLOOP command. The result is rounded off to suit the number of decimal places defined for the result data field. If the *result_data_ref* is a scratch-pad field, the number of decimal places used for the result of the calculation is the number of decimal places specified for the first operand of each iteration of the MATHLOOP command.

Operands for the MATHLOOP command, and values computed by the MATHLOOP command, can contain a maximum of 14 significant digits. Overflow or truncation errors may occur if you attempt to use values with more than 14 significant digits.

Example 1

```
MATHLOOP 16 *TOTALS(1) 1/100 0 = *TOTALS(1) 1
```

This command is performed 16 times. It causes each of the 16 communication fields *TOTALS(n) to be divided by 100. As 100 is a number, it must have a zero step factor.

Example 2

```
MATHLOOP 5 *S01 2 x P-scale 1= F-amt.rate 1
```

This command takes the five screen fields *S01, *S03, *S05, *S07 and *S09 on the current screen, and multiplies them by the five scratch-pad fields, starting with the field named *scale*. The command places the result of each multiplication into the file record buffer *rate*, starting with the field *amt*.

Example 3

```
MATHLOOP 7 *S03 2 x V-rate 0 = *S04 2
```

This command calculates the interest amounts for the principals given in the seven odd numbered screen fields starting with field number 3. The command places the result in the seven even numbered screen fields starting with field number 4. You can then redisplay these fields using the SHOW command.

12-142 Logic Commands

MESSAGE Command

The MESSAGE command displays a message on the screen.

Formats

MESSAGE *message_name*

Parameters

message_name

The name of a message defined previously using the dictionary message definition screen.

Description

The MESSAGE command displays a message on the screen. You can use this command to request user input, to query input, or to display information and warnings.

You must define the message using the dictionary message definition screen (refer to chapter 5).

A code associated with each message determines how HP ALLBASE/4GL displays the message. The codes are:

- **Mess.** A note for user information only. No user response is required.
- **Query.** A query or prompt that requires a response from the user.
- **Warn.** A warning requiring no response from the user.
- **Error.** An error that requires corrective action by the user.
- **Abort.** A system error that terminates the current HP ALLBASE/4GL application.

HP ALLBASE/4GL displays all messages on a two-line window at screen lines 23 and 24.

Query Messages

When you use a MESSAGE command to request and acquire user input, the system positions the cursor at the start of an input field in the message window to receive the user's response.

Pressing **(Return)** terminates the user's entry. A carriage return character by itself is acceptable as data. HP ALLBASE/4GL checks the user's response to ensure that it complies with the the edit code defined for the message response item. HP ALLBASE/4GL stores data entered by the user in the response item specified in the message definition screen.

Example

```
MESSAGE no_stock
```

This command displays the message *no_stock* within the message window.

MODE Command

The MODE command specifies the way files are used in the current process.

Formats

$$\text{MODE} \left\{ \begin{array}{l} *READ \\ *WRITE \\ *LOCK \\ *UNLOCK \end{array} \right\} \text{file_ref} \dots$$

Parameters

file_ref

The name of the application data files or HP TurboIMAGE/iX data sets that you want to specify access conditions for. Use the HP ALLBASE/4GL file name, not the external file name.

Description

The MODE command specifies the way the application's data files are accessed in the current process.

Note



You can only use the MODE command in a process. It cannot be used in a function.

Each MODE command remains in effect until another process is initiated. You can include more than one MODE command in a process, but subsequent MODE commands in a process will not affect the usage mode of any files that are open at the time.

You can specify as many file references as you wish.

The command argument *READ specifies that the file or files are available for reference purposes only. The files cannot be updated in any manner. The *READ argument is the default condition and applies to all files not explicitly specified in a MODE command.

The MODE command does not have any effect on HP ALLBASE/SQL table locking.

KSAM Files

With KSAM files, the MODE *LOCK, MODE *WRITE and MODE *UNLOCK forms of the command all perform the same function. They each specify that the file or files are available for reading and writing. Under any of these modes, HP ALLBASE/4GL locks the file at the first access that reads or writes a file record. (Locking does not occur if the file access command uses the *NOLOCK argument, and the file has not been locked by an earlier access.)

These different forms of the MODE commands are provided to allow application portability to and from other operating systems that allow record based locking of indexed files.

The MODE command operates in conjunction with the FILE command to determine the file locking procedure used by HP ALLBASE/4GL. Refer to the FILE command (\SAM) for a description of the HP ALLBASE/4GL file locking system.

HP TurboIMAGE/iX Data Sets

With HP TurboIMAGE/iX files, the MODE *LOCK, MODE *WRITE, and MODE *UNLOCK forms of the command operate also specify that the file or files are available for reading and writing.

MODE *WRITE should be used with care. It does not guarantee any transaction integrity, but can be used by developers without significant knowledge of HP TurboIMAGE/iX locking. When using MODE *WRITE, a data set level lock is placed.

The record remains locked to the current process until another record is accessed through the same file buffer, the buffer is cleared, the file is closed, or a FILE *UNLOCK command is executed.

Note

When a lock is released, all locks held on the database are released.



MODE *UNLOCK is similar to MODE *WRITE, except that a lock is not released until a specific FILE *UNLOCK command is executed. If a data set is

12-146 Logic Commands

locked, another user or process cannot lock the same data set while the lock is current.

The command argument `*LOCK` specifies that the process has write access to the file. Each file is locked when it is first accessed. The file or files remain locked until the end of the process, or the files are closed.

While a file is locked to a process that uses the `*LOCK` form of the `MODE` command, another process or user can read the file provided that the file is accessed under `MODE *READ`. However, while the `MODE *LOCK` command is in effect, other processes or users cannot write to the file or lock the file.

The `MODE` command and the `DM IMAGE *MODE` command.

The `*MODE` command may be used in conjunction with the `DM IMAGE *MODE` command. The two commands perform similar but discrete functions. The `DM IMAGE *MODE` command sets the access mode that HP ALLBASE/4GL uses to access an HP TurboIMAGE/iX database. The access mode specifies read and write access restrictions as well as enforcing locking requirements. The `MODE` command specifies, to HP ALLBASE/4GL, the read and write permissions for a single data set, and also specifies how HP ALLBASE/4GL should perform implicit locking of files if no `DM IMAGE *LOCK` commands are used to lock a file.

The capability provided by the `MODE` command must be a subset of a `DM IMAGE *MODE` command.

For example, `DM IMAGE *MODE *MODLOCK` and `MODE *READ` can be used together because reading is a subset of `DM IMAGE *MODE`

`*MODLOCK` capability, but `DM IMAGE *MODE *READLCK` and `MODE *WRITE` cannot be used together, because the `MODE` command specifies more capability than the `DM IMAGE *MODE` command.

If the `MODE` command used in conjunction with a `DM IMAGE *MODE` command is not a subset of the `DM IMAGE *MODE` command, an error will occur at run-time if an action is attempted that requires more capability than the `DM IMAGE *MODE` command provides.

Example

```
1 MODE *WRITE stock stck_trn
```

This command specifies that the two files *stock* and *stck_trn* may be updated until another process is initiated. All other files may be accessed for reference purposes only.

MOVE Command

The MOVE command copies the contents of a field to another field.

Format

$$\text{MOVE } \left\{ \begin{array}{l} \textit{from_data_ref} \\ *NULL \end{array} \right\} \textit{to_data_ref}$$

Parameters

from_data_ref

The field containing the data to be copied. It can be any of the following:

- Literal.
- Master title name.
- Numeric or alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area field.
- File record buffer.
- Work area buffer.

In the MOVE command, you can specify a screen field by the field name and optionally omit the screen name.

to_data_ref

The field to which the data is to be copied. It can be any of the following:

- Variable.
- Screen field reference.

- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Alterable communication area field.
- File record buffer.
- Work area buffer.

Description

The MOVE command copies the contents of one field to another field. The contents of the originating field remain unchanged. The previous contents of the receiving field are overwritten.

If you use the MOVE command to copy the contents of one field to another, and the *to_data_ref* field is longer than the *from_data_ref* field, the extra space in the *to_data_ref* field is filled with the pad character.

If you use the MOVE command to copy the contents of one buffer to another buffer, and the *to_data_ref* is longer than the *from_data_ref*, HP ALLBASE/4GL fills the remaining fields in the destination buffer with spaces.

If the *to_data_ref* buffer or field is shorter than the *from_data_ref* buffer or field, HP ALLBASE/4GL will truncate the data.

The MOVE *NULL form of the command sets the indicator variable associated with a select list field or an SQL table field to the null value setting, and fills the field with pad characters. You can only use this form of the MOVE command if the *to_data_ref* is a select list record field or an SQL table record field. Null value status is not preserved if you use the MOVE command to copy the contents of one buffer to another.

If you specify a file record buffer or a work area buffer as either the *from_data_ref* or the *to_data_ref* you cannot use substring references.

Example 1

```
MOVE *S01 F-item_no.stock
```

This command moves the contents of screen field number 1 to the field *item_no* on the file *stock*.

12-150 Logic Commands

Example 2

MOVE *DATE[4,2] V-month

This command moves the current month from the communication area field *DATE to the variable *month*.

Example 3

MOVE B-build_rec R-outfile

This command moves the entire contents of the work area buffer *build_rec* to the default record buffer for the file *outfile*.

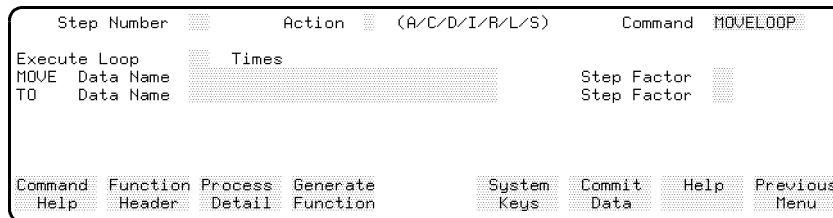
MOVELOOP Command

The MOVELOOP command repeatedly performs a MOVE operation.

Format

$$\text{MOVELOOP } loop_count \left\{ \begin{array}{l} from_data_ref \ step_factor_1 \\ *NULL \ 0 \end{array} \right\} \quad to_data_ref \\ step_factor_2$$

Window



Parameters

loop_count

A number specifying the number of times that the move is performed.

from_data_ref

The first field containing the data to be copied. It can be any of the following:

- Literal.
- Master title name.
- Numeric or alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.

12-152 Logic Commands

- Work area field reference.
- Communication area field.

to_data_ref

The first field to which the data is to be copied. It can be any of the following:

- Variable.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area fields *COUNT(n), *TOTALS(n), or *CROSS(n).

step_factors

The step factors are numbers specifying the increments to be applied to the *from* and *to* data reference names to obtain the data reference names for the next loop of the command. You can use different step factors for the *from* data reference and the *to* data reference.

The step factors may be a positive or negative number, or zero.

The increments are applied to the data reference names and not to their values. This enables successive fields on a screen or file record to be moved.

A zero step factor causes the same data reference to be used in each successive loop of the command. You must use a zero step factor with the following:

- Numeric or alphanumeric constant.
- Literal.
- Variable or calculated item.
- Master title name.
- Any communication area field other than *TOTALS, *COUNT, or *CROSS.
- The command argument *NULL.

Description

The MOVELOOP command repeatedly copies the contents of fields between a number of consecutive fields. The fields do not need to be adjacent.

The MOVELOOP command operates in a similar manner to the MOVE command except that the field move is repeated for each occurrence of the *loop_count*. In the MOVELOOP command, *from_data_ref* and *to_data_ref* incorporate a step factor.

The step factor determines the operand to be moved on each successive iteration of the loop.

If one of the data references for the MOVELOOP command is a file record field reference, or a work area field reference, HP ALLBASE/4GL steps through fields in the file record buffer or work area buffer according to the defined step factor. HP ALLBASE/4GL will also step through successive occurrences of fields defined in the dictionary as having more than one occurrence.

The MOVELOOP *loop_count* *NULL *to_data_ref* form of the command sets the indicator variable associated with *to_data_ref* to the null status and fills the field with the pad character. You can only use this form of the move command if the *to_data_ref* is a select list field or an HP ALLBASE/SQL table field.

Example 1

```
MOVELOOP 16 "0" 0 *TOTALS(1) 1
```

This command is performed 16 times. Each time, it moves the value zero into the next *TOTALS communication area field starting at *TOTALS(1). This command initializes the 16 *TOTALS fields.

Example 2

```
MOVELOOP 23 F-key.stock.buf1 1 F-key.stock.buf2 1
```

This command moves the 23 fields from the record buffer *buf1* of the KSAM file *stock* into the record buffer *buf2* of the same file. The move starts with the field *key* which occurs in each record.

12-154 Logic Commands

Example 3

```
MOVELOOP 5 * 2 *COUNT(1) 1
```

This command moves five fields from the current screen, starting with the current screen field, into the five *COUNT communication area fields. The command only moves every second screen field.

NOTE Command

The NOTE command inserts a comment into a process or function logic block.

Format

NOTE *comment*

Parameters

comment

Any series of characters except semicolons (;).

Description

The NOTE command inserts comments into a function or process logic block or annotates nominated command steps. HP ALLBASE/4GL ignores all characters in the note.

The NOTE command may precede or follow most other logic commands on the same step. When you use this facility, you must separate the NOTE command from the other commands with a semicolon.

Some logic commands allow you to specify an optional command that is executed if an error occurs while the command is being performed. These commands are CALC, DATE, FILE, MATH, MATHLOOP, and VALIDATE. You cannot use a NOTE command after any of these commands.

Example

VISIT V-fn_name; **NOTE** variable fn_name holds function name

OFF Command

The OFF command sets the status of a switch to *off*.

Formats

$$\text{OFF} \left\{ \begin{array}{l} \textit{switch_number} \\ *ALL \\ *BYPASS \\ *ENDLINE \end{array} \right\}$$

Parameters

switch_number

A number from 1 to 8, representing a user switch.

Description

The OFF command sets a switch to the *off* condition.

You can turn all user switches (number 1 to number 8) off using the argument *ALL. You can also use the OFF command to set the *BYPASS and *ENDLINE switches *off*.

The report generator uses the switch *BYPASS to determine if a record is selected for reporting. The *off* condition shows that the current record is selected for reporting.

The report generator uses the switch *ENDLINE to determine if processing of the current line group should continue. The *off* condition indicates that processing of the current line group continues.

Example

OFF 3

This command sets the status of user switch number 3 to *off*.

ON Command

The ON command sets a switch to *on*.

Formats

$$\text{ON } \left\{ \begin{array}{l} \textit{switch_number} \\ *ALL \\ *BYPASS \\ *ENDFIELD \\ *ENDLINE \end{array} \right\}$$

Parameters

switch_number

A number in the range from 1 to 8, representing a user switch.

Description

The ON command sets a switch to the *on* condition.

You can turn all user switches (number 1 to number 8) on using the argument **ALL*. You can also use the ON command to set the **BYPASS* and **ENDLINE* switches *on*.

The report generator uses the switch **BYPASS* to determine if a record is selected for reporting. The *on* condition indicates that the record is to be bypassed.

The report generator uses the switch **ENDLINE* to determine if processing of the current line group should continue. The *on* condition indicates that processing of the line group is terminated.

You can also use the ON command to set the switch **ENDFIELD* *on* in a function called from a function key or a prior function on an input field on a data screen. Setting **ENDFIELD* *on* terminates input processing for the current screen field.

12-158 Logic Commands

Example

ON *BYPASS

During record selection for reporting, this command bypasses the current file record.

PRINT Command

The PRINT command prints a line group on a report.

Format

PRINT *line_group*

Parameters

line_group

The line group to be printed. *Line_group* must be a defined line group for the current report.

Description

The PRINT command prints all the lines of the specified line group during report printing. The PRINT command is ignored if HP ALLBASE/4GL is not printing a report.

You cannot select lines within a line group being printed with a PRINT command. If you do need to bypass a line within a line group, use the line's before print function to set *BYPASS *on* to prevent the line being printed.

When you print a line group with a PRINT command, HP ALLBASE/4GL treats the line group as a detail line group. The only spacing performed by the system is the before print and after print spacing specified for the line. For example, the command PRINT B1 does not necessarily print the line at the bottom of the page.

Example

PRINT E3

This command prints the line(s) defined for the group E3.

PROCEED Command

The PROCEED command executes a process logic block.

Formats

$$\text{PROCEED } \left\{ \begin{array}{l} \textit{process_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

process_name

The name of the process to be executed.

data_ref

May be one of the following which contains the name of the process to be executed:

- A constant.
- A variable or calculated item.
- A screen field name.
- A scratch-pad field name.

Description

The PROCEED command executes a process logic block.

You must specify a valid *process_name* or a *data_ref* whose content is a valid process name. If the contents of *data_ref* cannot be resolved to a valid process name at run-time, the application aborts.

Whenever a new process starts, HP ALLBASE/4GL:

- Closes all open HP ALLBASE/SQL cursors, and rolls back any incomplete HP ALLBASE/SQL transactions.
- Closes all HP TurboIMAGE/iX data sets and releases all locks held on HP TurboIMAGE/iX data sets, records (data entries), or databases.

- Closes all application data files.
- Clears the data from all file buffers, work area buffers, and screen field buffers.

HP ALLBASE/4GL does not change the contents of the following whenever a new process starts:

- Variables.
- Scratch-pad fields.
- User switches and indicators.
- Communication area fields other than *PROCESS.

At the end of a process, control returns to the last menu displayed before the process commenced. If no menu has been executed, the application terminates at the end of the process.

Example

PROCEED main_update

This command executes the process *main_update*.

REPORT Command

The REPORT command prints a report.

Formats

$$\text{REPORT } \left\{ \begin{array}{l} \textit{report_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

report_name

The name of the report to be produced.

data_ref

Can be any of the following, containing the name of the report to be produced:

- A constant.
- A variable or calculated item.
- A screen field name.
- A scratch-pad field name.

Description

The REPORT command produces the named report. Once initiated, the report runs to completion. With the exception of the start-of-report function, HP ALLBASE/4GL does not allow any user interaction while the report is active. The start-of-report function can display a screen to accept data from the user.

Chapter 7 of this manual describes the HP ALLBASE/4GL report generator.

Any HP TurboIMAGE/iX database opened during the execution of a report is opened in the database access mode set for the database by the DM IMAGE *MODE command. If no mode has been specified for the database, it is opened in the default *READSHAR mode.

If a mode other than *READSHAR is required, you must specify the mode in a logic block. The DM IMAGE *MODE command that specifies the access mode must precede the REPORT command. Refer to the DM IMAGE *MODE command for more information about setting an access mode.

Example 1

```
REPORT stock_list
```

This command prints the report *stock_list*.

Example 2

```
1  DM IMAGE MODE *READEXCL :D-product
   .
   .
   .
5  REPORT products
   .
```

This command sets the HP TurboIMAGE/iX database access mode of the *product* database to *READEXCL . When the *products* report is run, the *product* database is opened in *READEXCL mode.

SCREEN Command

The SCREEN command displays a screen.

Formats

$$\text{SCREEN } \left\{ \begin{array}{l} \textit{screen_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

screen_name

The name of the screen to be displayed.

data_ref

May be one of the following containing the name of the screen to be displayed:

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

Description

The SCREEN command displays a screen.

If the screen is a menu, any current process, function, screen, decision table, or report is terminated immediately.

When HP ALLBASE/4GL starts processing a data screen, the flow of control in the application is determined by the screen and the logic associated with the screen. Displaying a screen terminates the current screen (if one is active) and clears the screen field buffers. The data screen remains active until the user or the logic associated with the screen initiates a screen commit action.

When a screen is committed, HP ALLBASE/4GL resumes running the application in one of three possible ways, depending how the screen was invoked.

- If the screen was displayed by a `SCREEN` command in a logic block, control returns to the command immediately following the `SCREEN` command.
- If the screen was displayed directly from a menu, control returns to the menu.
- If the screen was displayed from a `SCREEN` command in a function on a field on another screen, or from a function called from a function key on another screen, HP ALLBASE/4GL first returns to the function and executes the remaining commands in the function logic block. Since the original screen is terminated when the second screen is displayed, control then returns to the logic block or menu that displayed the original screen.

Note

In general, you should avoid calling further data screens while a data screen is active.

If you need to modify the current screen, use a window in preference to calling a further screen.

Chapter 6 describes the processing that is performed when HP ALLBASE/4GL displays a data screen.

Example

```
SCREEN V-updt_enq
```

This command displays the screen whose name is contained in the variable *updt_enq*.

SCROLL Command

The SCROLL command displays data in the scroll area of the current screen.

Format

SCROLL [*data_ref*] ...

Parameters

data_ref

The item or items to be displayed, each of which may be one of the following:

- Number (representing a number of spaces).
- Literal.
- Master title name.
- Numeric or alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Communication area field.

Description

The SCROLL command displays data on the appropriate scroll line of the current screen. HP ALLBASE/4GL displays data on the first line of the scroll area if the scrolling direction is down, and on the last line of the scroll area if the scrolling direction is up.

HP ALLBASE/4GL displays the contents of each *data_ref* in the order that you enter the arguments. HP ALLBASE/4GL does not add any intervening spaces. If you want to add intervening spaces to the scroll line, you can specify

a number as a data reference. HP ALLBASE/4GL interprets the number as the same number of spaces.

The action of the SCROLL command is similar to the DISPLAY command except that HP ALLBASE/4GL does not clear the scroll area, and you cannot specify the line where the display occurs.

Example

```
SCROLL *S01 3 *S02 "/" *S03 "/" *S04 5 "Added"
```

This command displays the first four screen fields. Field 1 is displayed followed by three spaces. Then the next three fields are displayed with embedded slashes, and finally five spaces are left followed by the word *Added*.

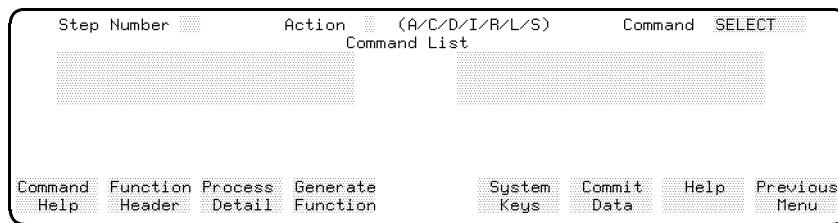
SELECT Command

The SELECT command executes one command in a list of commands according to the value in the communication area field *PASS.

Format

SELECT command; command; ...

Window



Parameters

command

The command to be executed. The command can be any of the following:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP
- VISIT
- ZIP

You must separate the commands in the command list with semicolons.

Description

The SELECT command executes a specified command from a list of logic commands.

The value in the communication area field *PASS determines the number of the command that the SELECT command executes. If the number in *PASS does not correspond to a command in the command list, HP ALLBASE/4GL executes the command in the logic block following the SELECT command.

You must terminate all commands other than the last command in the list with a semicolon. If you enter the commands via the SELECT window, the semicolon terminator is optional. In this case, HP ALLBASE/4GL inserts the semicolon automatically if you don't specify it during entry.

You can use the SELECT command in conjunction with a CHECK command. The CHECK command sets the content of *PASS, and the SELECT command chooses the relevant command to execute from the command list.

Note



The SELECT command only executes one command. If you enter more than one command in the same field via the window without the intervening semicolon, HP ALLBASE/4GL will interpret the commands as two separate commands in the selection list.

Example 1

```
5 MOVE * *PASS
6 SELECT ENTER 5; ENTER 12; ENTER 15; PROCEED main_menu
```

These commands take the number from the current screen field and execute the command in the command list that corresponds to that number. For example, if the current screen field contains the number 2, HP ALLBASE/4GL executes the command *ENTER 12* since this is the second command in the command list.

Example 2

```
3 SCREEN get_command
4 CHECK *S01 *TABLE=commands
5 SELECT PROCEED command_1; PROCEED command_2; EXIT
```

12-170 Logic Commands

6 MESSAGE reenter

7 ENTER 3

These commands accept input from the screen *get_command* and validate the data against the table *commands*. The CHECK command places its result into *PASS. The SELECT command then uses this value to execute the appropriate HP ALLBASE/4GL command from the command list. If the value in *PASS is not between 1 and 3, HP ALLBASE/4GL displays the message *reenter*.

SERIES Command

The SERIES command executes a range of steps in the current logic block.

Format

$$\text{SERIES } \left\{ \begin{array}{l} \textit{begin_step_number} \\ \textit{begin_data_ref} \end{array} \right\} \left\{ \begin{array}{l} \textit{end_step_number} \\ \textit{end_data_ref} \end{array} \right\}$$

Parameters

begin_step_number

The number of the first step to be executed.

end_step_number

The number of the last step to be executed.

begin_data_ref

One of the following, containing the first step number to be executed.

- Numeric constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.
- File record field reference.
- Work area field reference.

end_data_ref

One of the following, containing the number of the last step to be executed.

- Numeric constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

12-172 Logic Commands

- File record field reference.
- Work area field reference.

Description

The SERIES command executes a range of command steps within the current logic block.

The start and end step numbers can have the same value. If this is the case, HP ALLBASE/4GL only executes one step.

HP ALLBASE/4GL executes the command steps nominated in the SERIES command. HP ALLBASE/4GL then resumes executing the logic block with the command following the SERIES command if the SERIES command does not initiate a PROCEED command,

HP ALLBASE/4GL executes the commands specified by the SERIES command as if they were contained in a separate logic block called by the VISIT command. If the commands nominated by the SERIES command include an EXIT command, execution of the commands specified by the SERIES command ceases and control returns to the command immediately following the SERIES command.

The steps nominated by the SERIES command cannot include an ENTER command that refers to a step outside the range of the series command.

If the commands nominated by the SERIES command contain a further SERIES command, the steps nominated by the second SERIES command must be wholly contained within the steps for the first SERIES command.

The step numbers you nominate in a SERIES command must be valid step numbers for the current logic block.

Example 1

```
SERIES 12 15
```

This command executes steps 12 to 15 inclusive. Processing then resumes at step number 4.

Example 2

```
SERIES 8 V-last
```

This command executes steps starting at step number 8 and ending at the step number defined by the variable *last*.

If the value in *last* is greater than the number of steps in the logic block, all steps from step 8 onwards will be executed.

If the value in *last* is less than 8, the SERIES command is ignored.

SHOW Command

The SHOW command displays the contents of fields on the current screen.

Formats

SHOW [*REFRESH] [*from_data_ref* [*to_data_ref*]]

Parameters

from_data_ref, *to_data_ref*

The starting and ending field numbers to be displayed. Each can be one of the following:

- A whole number.
- A screen field name.
- A * (representing the current field).
- A relative screen field reference such as *-9 to *+9.

In the SHOW command you can specify a screen field by the field name only, and you can optionally omit the screen name.

If you only specify one data reference, HP ALLBASE/4GL only displays one field.

Description

The SHOW command displays the contents of a single screen field, a range of screen fields, or all fields on the current screen.

The SHOW command with no arguments displays the contents of all fields on the current screen.

If you only specify one data reference with the SHOW command, HP ALLBASE/4GL only displays the contents of that one field.

If you specify both a *from* data reference and a *to* data reference with the SHOW command, HP ALLBASE/4GL displays the contents of all fields within that range on the current screen.

The SHOW command does not change the movement of the cursor on the screen.

***REFRESH Option**

The normal operation of the SHOW displays the contents of the internal screen buffer only.

If you use the *REFRESH command argument, HP ALLBASE/4GL refreshes the internal screen buffer from the data movement fields specified on the screen field detail screen (refer to chapter 6).

HP ALLBASE/4GL refreshes the internal screen buffer from:

- The primary data movement field, or
- The default data movement field.

HP ALLBASE/4GL only uses the contents of the default data movement field if the screen field buffer is blank. (A screen field cannot have a default data movement field if it has a primary data movement field.)

Typically, you would use the *REFRESH argument to display the contents of a file record after it has been read. By specifying the file record fields as primary data movement fields for the relevant screen fields, the SHOW *REFRESH command transfers the contents of the file buffer to the screen buffer and displays the data on the screen.

Example 1

```
SHOW 2 8
```

This command displays the contents of fields number 2 to number 8 inclusive from the screen buffer.

Example 2

```
1 FILE *NEXT stock
2 SHOW *REFRESH
3 MESSAGE Stock_record
4 TIE 4
```

These commands:

- Read the next record from the file *stock*.

12-176 Logic Commands

- Display the contents of the record on the screen. (Assuming that you have defined the file record fields as the primary data movement fields for the screen fields.)
- Display the message *Stock_record*.
- Position the cursor at field number 4 ready to accept data.

Example 3

SHOW *REFRESH *

This command moves the contents of the primary data movement field (or the default data movement field) into the screen buffer and displays the contents on the screen at the current field.

SQL Command

The SQL command executes an SQL logic block.

Formats

$$\text{SQL } \left\{ \begin{array}{l} \text{block_name} \\ \text{data_ref} \end{array} \right\} [; \text{command}]$$

Parameters

block_name

The name of the SQL logic block to be executed.

data_ref

May be one of the following containing the name of the SQL block to be executed.

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

command

An optional command to be executed if a recoverable error occurs during the execution of the SQL block. If you don't specify an error command, and an error does occur, HP ALLBASE/4GL displays the text of the error message on the screen and control passes to the next command in the logic block.

You can use the following commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED

12-178 Logic Commands

- SERIES
- TOP
- VISIT
- ZIP

Description

The SQL command executes an SQL logic block.

SQL logic blocks are used to pass SQL commands from HP ALLBASE/4GL to HP ALLBASE/SQL. At run-time, the SQL command in an HP ALLBASE/4GL logic block passes the HP ALLBASE/SQL commands in the SQL logic block to the HP ALLBASE/4GL/HP ALLBASE/SQL interface. After the SQL block has been executed, control passes to the next command in the logic block.

If an SQL logic block contains a SELECT command, this must be the only command in the SQL logic block. Otherwise, the SQL block can contain up to eight SQL commands that generate into stored database sections. You must define the SQL logic block using the appropriate header and details screens in the logic menu.

Refer to chapter 9 for further details of the SQL commands that are permitted in an SQL logic block, and the usage of these commands.

Error Handling

If an HP ALLBASE/SQL error occurs during execution of an SQL logic block, execution of the SQL logic block terminates. If you have specified the optional error command on the SQL command, control passes to the error command when an error is detected by HP ALLBASE/SQL. If you haven't specified an error command, control passes to the next command in the HP ALLBASE/4GL logic block when an HP ALLBASE/SQL error occurs.

When HP ALLBASE/SQL detects an error at run-time, HP ALLBASE/4GL places the text of the error message in the communication area field *ERROR, and places the error number, plus 100 000 in the communication area field *IOSTATUS.

If the SQL command that caused the error does not contain an optional error command, the text of the error message is displayed on the screen.

In some cases, a single SQL statement may cause more than one error message. Under these conditions, the first error message and error number are loaded into *ERROR and *IOSTATUS respectively at the time of error detection. If the SQL command contains an optional error command, the error handling logic can invoke the SQLEXPLAIN command in a further SQL logic block to retrieve the text and number of subsequent errors. The text of the message is returned to the *ERROR communication area field, and the error number is returned to *IOSTATUS. (Note that you cannot specify a host variable on the SQLEXPLAIN command in an SQL logic block.) The SQLEXPLAIN command returns a null string after the last error has been returned.

To determine the HP ALLBASE/SQL error number, subtract 100 000 from the HP ALLBASE/4GL error message number. Refer to the *HP ALLBASE/SQL Messages Manual* for the causes and suggested remedial actions for HP ALLBASE/SQL error messages.

Example

```
1 SQL customer
2 FILE *NEXT namelist
3 SHOW *REFRESH
.
.
.
7 SQL customer_updt
```

In this logic block, the first SQL command executes the SQL logic block *customer*. This SQL logic block typically would contain an HP ALLBASE/SQL SELECT command to declare and open a cursor defined by the select list *namelist*. The following FILE *NEXT command retrieves the first record from the active set defined by the SELECT command.

The remaining commands in the logic block then use the record retrieved by the FILE *NEXT command. The final SQL command executes the SQL logic block *customer_updt*. This SQL logic block could contain an SQL UPDATE command and a COMMIT WORK command.

12-180 Logic Commands

TIE Command

The TIE command specifies the number of the next field to be processed on the current screen.

Format

TIE *data_ref*

Parameters

data_ref

One of the following, containing the number of the screen field:

- A whole number.
- A screen field name.
- A * (representing the current field).
- A relative screen field reference such as *-9 to *+9.

In the TIE command you can specify a screen field by the field name only. You can optionally omit the screen name.

Description

The TIE command specifies the number of the field to be processed on the current screen. This command takes effect after HP ALLBASE/4GL has finished processing the current screen field. You only need to use the TIE command if the next required field is not the next numbered field on the screen.

The TIE command is only valid in:

- A prior function called from an input or output field.
- An after function called from an input field.
- A function called from a function key on a data screen.

If you use the TIE command in a function called from a function key or a prior function on a screen field, you can set the switch *ENDFIELD *on* to terminate processing of the current screen field.

If you use the TIE command from within a window, the field you tie to can be a field on the window or on the main screen. For example, the command TIE 1 from a window causes a tie to the first field on the screen, not the first field on the window.

Example

TIE 5

This command specifies that the field number 5 is the next field to be processed.

TOP Command

The TOP command passes control to the first step of current logic block.

Format

TOP

Description

The TOP command returns to and executes the first step in the current logic block.

The TOP command is an alternative for ENTER 1.

TRANSACT Command

The TRANSACT command enables you to define logically connected groups of KSAM file, HP ALLBASE/SQL table, and HP TurboIMAGE/iX data set modification commands that constitute one transaction.

Formats

$$\text{TRANSACT} \left\{ \begin{array}{l} *BEGIN \\ *END \\ *MEMO \\ *UNDO \end{array} \right\} [\textit{message}]$$

Parameters

message

A comment, made up of any characters except semicolons.

Description

The TRANSACT command allows you to define logical groups containing file modification commands that are interdependent. That is, groups of commands where if one file modification command in the group is executed, then all file modification commands in the group must be executed to maintain logical consistency. The *BEGIN and *END arguments define the beginning and end of the transaction command block. If the application has connected to an HP ALLBASE/SQL database, HP ALLBASE/4GL issues implicit HP ALLBASE/SQL BEGIN WORK and COMMIT WORK commands at these points.

The TRANSACT *UNDO command has no effect on HP TurboIMAGE/iX file transactions, but *is* effective for KSAM file transactions and HP ALLBASE/SQL file transactions.

If HP ALLBASE/4GL executes a TRANSACT *UNDO command after a TRANSACT *BEGIN command, but before a TRANSACT *END command is encountered, all file transactions performed since the TRANSACT *BEGIN command are reversed. The TRANSACT *UNDO command:

12-184 Logic Commands

- Issues an SQL ROLLBACK WORK command to undo any uncommitted SQL transactions,
 - and then
- Reverses all KSAM data file transactions that have been performed since the TRANSACT *BEGIN command.
 - but
- Has no effect on HP TurboIMAGE/iX file transactions. All recovery of HP TurboIMAGE/iX file transactions must be set up manually by the HP TurboIMAGE/iX database administrator.

After the TRANSACT *UNDO command, HP ALLBASE/4GL executes the next command in the logic block containing the TRANSACT *UNDO command.

The TRANSACT *BEGIN, TRANSACT *UNDO, and TRANSACT *END commands do not need to be in the same logic block.

If HP ALLBASE/4GL encounters two TRANSACT *BEGIN commands without an intervening TRANSACT *UNDO command or a TRANSACT *END command, all uncommitted SQL transactions and all KSAM file transactions performed since the first TRANSACT *BEGIN command are reversed, and a new transaction block is started.

A user may voluntarily cancel a partially complete transaction by terminating a data screen before the screen has been committed. If the user's action terminates the process containing the TRANSACT *BEGIN command, the system reverses all file transactions performed since the last TRANSACT *BEGIN command.

HP ALLBASE/4GL displays messages to inform the user that KSAM file transactions and/or currently open HP ALLBASE/SQL transactions are being reversed whenever it executes a TRANSACT *UNDO command.

The TRANSACT *MEMO command is only relevant to HP TurboIMAGE/iX file transactions, and is used to include comments in an HP TurboIMAGE/iX log record. This command has no effect on KSAM or HP ALLBASE/SQL transactions. For more information about HP TurboIMAGE/iX log records, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

File Locking

The TRANSACT *END and TRANSACT *UNDO commands issue implicit FILE *UNLOCK commands for KSAM files opened in MODE *UNLOCK. The FILE *UNLOCK command releases all KSAM file locks belonging to the current process.

Limitations

The HP ALLBASE/4GL transaction logging system does not record transactions involving serial data files. This means that the TRANSACT *UNDO command does not reverse any serial file transactions that have occurred since the previous TRANSACT *BEGIN command. The TRANSACT *UNDO also does not reverse any HP TurboIMAGE/iX file transactions.

The TRANSACT command only supports single level transactions, and does not provide savepoints. For transactions that only use HP ALLBASE/SQL, you can use the BEGIN WORK, SAVEPOINT, COMMIT WORK and ROLLBACK WORK commands in SQL logic blocks. If you do use these commands in SQL logic blocks, don't use TRANSACT commands in the HP ALLBASE/4GL logic blocks that call the SQL block.

For transactions that involve more than one of KSAM data files, HP ALLBASE/SQL tables, and HP TurboIMAGE/iX data sets, the begin and end records are logged for each data manager used. If the system fails, or the user cancels a transaction in the time between the creation of two transaction end records, inconsistency of the databases may result. When the user next signs on to HP ALLBASE/4GL and connects to the data base environment, any incomplete HP ALLBASE/SQL transaction is rolled back to ensure logical consistency of the data base. The KSAM data manager does not provide this facility, so the KSAM part of a mixed transaction may be inconsistent with the HP ALLBASE/SQL part of the transaction. The HP TurboIMAGE/iX data manager provides a rollback facility, but unless the HP TurboIMAGE/iX administrator has rolled back the transaction, the HP TurboIMAGE/iX part of a mixed transaction may also be inconsistent with the HP TurboIMAGE/iX part of the transaction.

12-186 Logic Commands

Example 1

```
.  
.
3 TRANSACT *BEGIN
.
5 IF ... THEN ENTER 10 (error condition)
.
.
8 TRANSACT *END
9 EXIT
10 TRANSACT *UNDO
11 MESSAGE no_transaction
12 EXIT
```

In this logic block, steps 3 and 8 mark the beginning and end of the block of interdependent commands. If an error occurs (step 5), HP ALLBASE/4GL restores all partially completed transactions (except any that use HP TurboIMAGE/iX data sets) to their previous state at step 10. Step 11 displays a developer defined message to the user.

Example 2

```
.  
.
3 TRANSACT *BEGIN
.
5 IF ... THEN ENTER 10 (error condition)
6 TRANSACT *MEMO This task has been done successfully.
.
.
8 TRANSACT *END
9 EXIT
10 MESSAGE no_transaction
12 EXIT
```

In this logic block, steps 3 and 8 mark the beginning and end of the block of interdependent commands. If no error occurs, the message “*This task has been done successfully.*” is written to the HP TurboIMAGE/iX database log record.

UPDATE Command

The UPDATE command writes changed data from all KSAM file buffers, serial file buffers, and HP TurboIMAGE/iX file buffers that have been modified.

Format

UPDATE

Description

The UPDATE command updates each KSAM file, HP TurboIMAGE/iX data set, and serial data file whose file buffers have been changed. HP ALLBASE/4GL writes the records as new records or modified records as appropriate.

You cannot update HP ALLBASE/SQL tables using the HP ALLBASE/4GL UPDATE command. Use the FILE *INSERT command or the SQL UPDATE command in an SQL logic block to update HP ALLBASE/SQL tables.

Note



You cannot use the UPDATE command with HP TurboIMAGE/iX data sets or KSAM files that allow duplicate primary keys.

The UPDATE command performs the same action as issuing a FILE command with a *WRITE argument for each file.

For writing to a specific file buffer, refer to the FILE command.

VALIDATE Command

The VALIDATE command checks for the presence or absence of a field on an indexed (KSAM) file and optionally displays a message. The VALIDATE command cannot be used with HP ALLBASE/SQL tables, select lists, HP TurboIMAGE/iX data sets, or serial files.

Formats

VALIDATE *file_field_ref* [*INDEX=*index_name*]

*KEY=*key* { *PRESENT } [*message_ref*] [; **command**]
 *ABSENT }

Window

Step Number	Action	(A/C/D/I/R/L/S)	Command	VALIDATE			
Field Name				F-Data Name.File[.Record]			
*INDEX=							
*KEY=							
Validate as	(<*ABSENT,*PRESENT)						
On Error							
Command	Function	Process	Generate	System	Commit	Help	Previous
Help	Header	Detail	Function	Keys	Data		Menu

Parameters

file_field_ref

The field that you want to check as present or absent within the file.

You must express the field in the format:

F-field_name.file_name[.record_name]

That is, you must use a valid file record field reference.

index_name

This parameter allows you to specify the name or number of the index to be used to access the file. If you do specify an index name or number, you must precede it with *INDEX=. The index name must be the name of a field

specification defined as a key field on the default record layout for the file. The index number must be a valid index for the file.

If you don't specify an index name, HP ALLBASE/4GL accesses the file according to the index specified by the current value of *INDEXNO.

key

The key data used to access the file *file_name[.record_name]* to check for the presence or absence of the field. It can be one of the following:

- Literal.
- Alphanumeric constant.
- Variable or calculated item.
- Screen field reference.
- Scratch-pad field reference.
- File record field reference.
- Work area field reference.
- Non-numeric communication area field.

message_ref

A message to be displayed if the validation fails.

command

A command to be executed if the validation fails. You must precede the command name with a semicolon. The command can be any one of the following logic commands:

- ENTER
- EXIT
- EXTERNAL
- MESSAGE
- PROCEED
- SERIES
- TOP

12-190 Logic Commands

- VISIT
- ZIP

Description

The VALIDATE command checks for the absence or presence of a specified field on a file record.

The absence or presence of a field depends on its type as follows:

Field type	Absent if	Present if
Numeric	Zero	Not zero
Not numeric	Blank	Not blank

A check for absence is also satisfied if the file record specified by *key* cannot be located. In this situation, HP ALLBASE/4GL clears the file record buffer.

HP ALLBASE/4GL performs the file read using the key field specified by the current value of *INDEXNO or the *INDEX= argument. If the index allows duplicate key values, only the first occurrence is used.

If the VALIDATE command does not fail, (that is, the field is found to be absent or present as required) HP ALLBASE/4GL executes the next command in the logic block.

If the validation check (present or absent) fails, and the file record is located, the file record buffer contains the record read from the file.

When the validation check fails, HP ALLBASE/4GL displays any message and executes the optional command.

Example

VALIDATE F-deleted.stock *KEY=* *ABSENT item_deleted; **TOP**

This command reads the file *stock* using the contents of the current screen field as the key for the read. If HP ALLBASE/4GL locates the record, it checks the contents of the field *deleted*. If the field is blank, HP ALLBASE/4GL executes the next command step. If the field is not blank, HP ALLBASE/4GL displays the message *item_deleted* and resumes processing from the first step in the logic block.

VISIT Command

The VISIT command executes a function logic block.

Formats

$$\text{VISIT } \left\{ \begin{array}{l} \textit{function_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

function_name

The name of the function to be executed.

data_ref

One of the following, containing the name of the function to be executed:

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

Description

The VISIT command executes a function logic block. Unless the function contains a PROCEED command, HP ALLBASE/4GL executes the step following the VISIT command after the function has been executed.

Functions can be nested, and recursion is permitted. The maximum depth of nesting of functions is 99.

Example

```
VISIT delete_rec
```

This command executes the function *delete_rec*.

12-192 Logic Commands

WINDOW Command

The WINDOW command displays a window on the current screen.

Formats

$$\text{WINDOW} \left\{ \begin{array}{l} \textit{window_name} \\ \textit{data_ref} \end{array} \right\}$$

Parameters

window_name

The name of a screen that has been defined as a window. (Type W on the screen header screen.)

data_ref

One of the following, containing the name of a window:

- Constant.
- Variable or calculated item.
- Screen field name.
- Scratch-pad field name.

Description

The WINDOW command displays the specified window, starting at the window starting line specified for the underlying data screen. HP ALLBASE/4GL displays the window from the nominated line through to the last line of the screen.

HP ALLBASE/4GL renumbers the fields defined for the window, starting with the highest field number on the main data screen plus one. For example, if the main data screen's highest field number is 23, the first field on the window will be number 24.

When HP ALLBASE/4GL displays a window, the value in *SCREEN becomes the name of the window.

If you define a set of function keys for the window, these function keys replace the function keys for the main screen. If you don't define a set of function keys for the window, the function keys for the main screen remain unchanged. Similarly, if you define a screen help name for a window, it is used in preference to the screen help defined for the main screen.

The effect of the WINDOW command depends on when it is invoked, as follows:

- If there is no current screen: HP ALLBASE/4GL ignores the WINDOW command.
- From a function in a screen, or a function called from a function key: HP ALLBASE/4GL displays the window, or clears the window of data if one is already displayed. HP ALLBASE/4GL then executes the remaining commands in the logic block, and then positions the cursor at the first input field on the window.
- After the SCREEN command has been completed: HP ALLBASE/4GL displays the window and executes the next command in the current logic block.

A WINDOW command does not automatically display the window. In contrast, the SCREEN command displays the named screen immediately.

Where one or more WINDOW commands immediately follow a WINDOW command, HP ALLBASE/4GL displays each window in turn. Only the last WINDOW command takes effect. That is, only the last window remains displayed. For example:

```
WINDOW add_item_1
WINDOW add_item_2
```

If these commands are in a function called from a screen field, HP ALLBASE/4GL displays the window *add_item_1* first, and then immediately displays the window *add_item_2*. The window *add_item_2* remains on the screen and input commences from the first input field defined for that window.

Example 1

```
IF *- "A" THEN WINDOW add_data
ELSE WINDOW enq_data
```

12-194 Logic Commands

This command is called from within a screen and initiates one of two windows depending on the option the user enters.

Example 2

```
4 MOVE "S" V-scr_status
5 SCREEN new_stock
6 IF *S24 &<> "A" THEN ENTER 5
7 WINDOW add_activity
8 MOVE "25" V-field_no
9 MOVE "W" V-scr_status
10 ENTER 5
```

These commands show the use of a window outside the main processing sequence of a screen.

The screen *new_stock* has a prior-entry function on the first field that checks the content of the variable *scr_status*. If it is “W” then a screen window has been displayed. In this case, the function moves the contents of variable *field_no* to *FIELDNO.

Step 8 moves the number of the first field on the window into the variable *field_no*. The last field number on the screen is 24.

On entry to the screen, HP ALLBASE/4GL processes field number 25, (that is, the first field for the window), rather than field number 1. This redirection occurs because the prior function has set the value of *FIELDNO.

In this situation it is necessary to go back into the main data screen as this is the only way that data can be accepted from the window.

ZIP Command

The ZIP command performs no operation.

Format

ZIP

Description

The ZIP command is used when it is required to perform no operation and execute the next command in the current logic block.

You can use this command for the THEN or ELSE portions of an IF or IFLOOP command when no action is required.

Utilities

This chapter describes the screens in the HP HP ALLBASE/4GL utilities menu. The screens are described under the following subheadings:

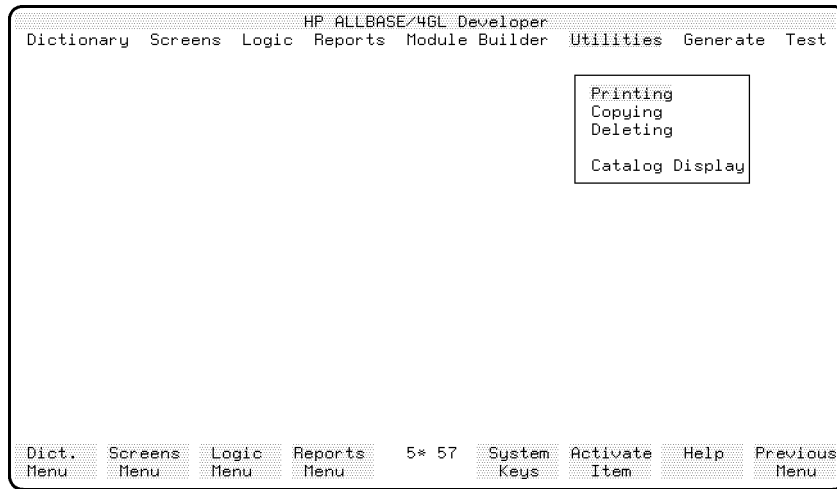
- Screen Image.
- Field Descriptions.
- Function Keys.
- Additional Information.

This chapter also provides an outline description of a number of additional utility tools that may be useful during application development.

Utilities Menu

The utilities menu allows you to access a number of HP HP ALLBASE/4GL utilities to print, copy, and delete components of an application, or display the names of application components on the screen.

Screen Image



Menu Selections

Printing. This selection displays the catalog printing screen, allowing you to print some or all of the developer documentation for an application.

Copying. This selection displays the catalog copying screen, allowing you to copy components within an application or version, or copy components from another application or version.

Deleting. This selection displays the catalog deletions screen, allowing you to delete components within an application.

Catalog Display. This selection displays the catalog display screen. This screen allows you to display lists of application components on the terminal screen.

13-2 Utilities

Function Keys

- f1** Dict. Menu Cancels the current screen and displays the dictionary menu.
- f2** Screens Menu Cancels the current screen and displays the screens menu.
- f3** Logic Menu Cancels the current screen and displays the logic menu.
- f4** Reports Menu Cancels the current screen and displays the reports menu.

Catalog Printing Screen

This screen allows you to print documentation for some or all components of an application. You can print the documentation with or without the full details of each component, and with or without the description text for each component.

Screen Image

The screenshot shows a terminal window titled "Developer Catalog Printing printing". It contains a list of 22 application components, each with a number and a name. Below the list are three control options: "Catalog Item Type Number" (with a cursor), "Short Index List or Full Index List", and "Print Catalog Items Details All Entries or Selected Names". A "Selected Names" field is also present. At the bottom, there are several menu options: "Copying", "Deleting Catalog Display", "13* 39", "System Keys", "Commit Data", "Help", and "Previous Menu".

Number	Component Name
1	Field Specs.
2	Ranges
3	Tables
4	Variables
5	Calculated Items
6	Numeric Constants
7	Alpha-Numeric Constants
8	Scratch-Pad Fields
9	Application Titles
10	Record Layouts
11	Files
12	Messages
13	Help Screens
14	Screens
15	Function Keys
16	Processes
17	Functions
18	Decision Tables
19	Reports
20	SQL Select Lists
21	SQL Logic Blocks
22	Work Areas

Control options:
Catalog Item Type Number: (1 - 22 or A for All Item Types)
Short Index List or Full Index List: (S/F/None)
Print Catalog Items Details All Entries or Selected Names: (Y/N)
All Entries or Selected Names: (A/S)

Selected Names:

Menu options: Copying, Deleting Catalog Display, 13* 39, System Keys, Commit Data, Help, Previous Menu

Field Descriptions

Catalog Item Type Number. Enter the number of the type of application components you want to print, or enter A to print the documentation for all the component types in the catalog.

Short Index List or Full Index List. The entry in this field specifies the type of index list printed with the report. If you enter S, the index list contains the application component names, and the short descriptions from the documentation fields on the relevant item header or definition screen. If you enter F, the index list contains the complete documentation fields and the short description fields for the selected items. If you enter N, the system does not print an index list for the report.

13-4 Utilities

Print Catalog Item Details. If you enter **Y** in this field, the system prints the details for each component. Enter **N** if you want to suppress printing of the component details.

All Entries or Selected Names. If you want to print a report for all items of the selected type, enter **A** in this field.

If you only want to print a report for selected items, enter **S** in this field and enter the names of the items you want reported in the *Selected Names* field.

Selected Names. If you entered **S** in the field above, enter the names of the items you want to include in the report in these fields.

Function Keys

- f2** **Copying** Cancels the current screen and displays the catalog copying screen.
- f3** **Deleting** Cancels the current screen and displays the catalog deletions screen.
- f4** **Catalog Display** Cancels the current screen and displays the catalog display screen.

Additional Information

HP HP ALLBASE/4GL developer documentation reports consist of two components.

The first component is the index list for the report, which is a list of the items in the report. The index can either be a full index or a short index. The full index contains the names of the items and the text from both the short description fields and the full description fields from the relevant item header screens. The short index only contains the item names and the text of the short description fields. In all cases, HP HP ALLBASE/4GL sorts the item names according to the system collating sequence (refer to appendix A). You can suppress the index list by entering **N** in the *Short Index List or Full Index List* field.

The second component in the report contains the details of the items selected for reporting. If you enter **S** in the *ALL Entries or Selected Names* field and leave the *Selected Names* fields blank, HP HP ALLBASE/4GL does not print

the second component of the report. In this way you can obtain an index list of some or all the items defined in an application.

If you enter **A** in the *Catalog Item Type Number* field, HP HP ALLBASE/4GL reports on all items defined for the application.

When you press the **Commit Data** function key, HP HP ALLBASE/4GL calls an external program that formats the documentation and writes the report to DEV=LP. The formal file designator for the report is HP4REPT.

The sequence of printing the documentation is as shown on the top part of the screen. HP HP ALLBASE/4GL displays progress messages indicating the documentation currently being produced.

13-6 Utilities

Catalog Copying Screen

This screen allows you to copy components within an application or version, and to copy components from other applications or versions.

Screen Image

The screenshot shows a terminal window titled "Developer: Catalog Copying copying". It contains a list of 22 items to be copied, organized in three columns. Below the list are input fields for "Application to copy FROM" (with the value "example"), "Item Type Number", "Item Name to Copy FROM", and "Item Name to Copy TO". A "Description" field is also present. At the bottom, there is a "Last Modification:" section with "Date" and "Time" fields, and a row of buttons: "Printing", "Deleting", "Catalog Display", "13* 48", "System Keys", "Commit Data", "Help", and "Previous Menu".

Item Number	Item Name	Item Number	Item Name	Item Number	Item Name
1	Field Specs.	10	Record Layouts	15	Processes
2	Ranges	11	Files	17	Functions
3	Tables			18	Decision Tables
4	Variables	12	Messages	19	Reports
5	Calculated Items	13	Help Screens		
6	Numeric Constants			20	SQL Select Lists
7	Alpha-Numeric Constants	14	Screens	21	SQL Logic Blocks
(n/a	Scratch-Pad Fields)	15	Function Keys		
9	Application Titles			22	Work Areas

Application to copy FROM: example
Item Type Number:
Item Name to Copy FROM:
Item Name to Copy TO:
Description:
Last Modification: Date Time

Buttons: Printing, Deleting, Catalog Display, 13* 48, System Keys, Commit Data, Help, Previous Menu

Field Descriptions

Application to copy FROM. Enter the name of the application or version that you want to copy the item from.

Item Type Number. Enter the number of the type of item you want to copy.

Item Name to Copy FROM. Enter the name of the item you want to copy.

Item Name to Copy TO. Enter the name of the item to be created. This name must not already exist for the type of item being copied. In particular, if you copy a screen, the new name must **not** be defined on the screen header screen. The copy facility copies the screen header as well as the screen image.

Description. Display-only field showing the details from the description fields of the item to be copied.

Function Keys

- f1** **Printing** Cancels the current screen, and displays the catalog printing screen.
- f3** **Deleting** Cancels the current screen, and displays the catalog deletions screen.
- f4** **Catalog Display** Cancels the current screen and displays the catalog display screen.

Additional Information

You cannot copy a generated component from another application unless the source information for the component also exists in the application.

You must generate the new item after the copy has finished. The copy process does not generate the new item automatically. This allows you to modify the new item before you generate it.

The copy does not occur until you press the **Commit Data** function key.

13-8 Utilities

Catalog Deletions Screen

This screen allows you to delete components within an application.

Screen Image

The screenshot shows a terminal window titled "Catalog Deletions" with the following content:

```
Developer: Catalog Deletions deleting
1 Field Specs.      10 Record Layouts  16 Processes
2 Ranges          11 Files           17 Functions
3 Tables          12 Messages       18 Decision Tables
4 Variables       13 Help Screens   19 Reports
5 Calculated Items 14 Screens        20 SQL Select Lists
6 Numeric Constants 15 Function Keys  21 SQL Logic Blocks
7 Alpha-Numeric Constants
8 Scratch-Pad Fields
9 Application Titles
22 Work Areas

Item Type Number [ ]
Item Name to Delete [ ]

Description

Last modification:      Date      Time

Printing Copying Catalog 14* 40 System Commit Help Previous
Display      Display Keys   Data  Menu
```

Field Descriptions

Item Type Number. Enter the number that corresponds to the item type to be deleted.

Item Name to Delete. Enter the name of the item to be deleted.

Description. Display-only field showing the details from the description fields of the item to be deleted.

Function Keys

f1) Printing Cancels the current screen and displays the catalog printing screen.

f2) Copying Cancels the current screen and displays the catalog copying screen.

- f4** **Catalog Display** Cancels the current screen and displays the catalog display screen.

Additional Information

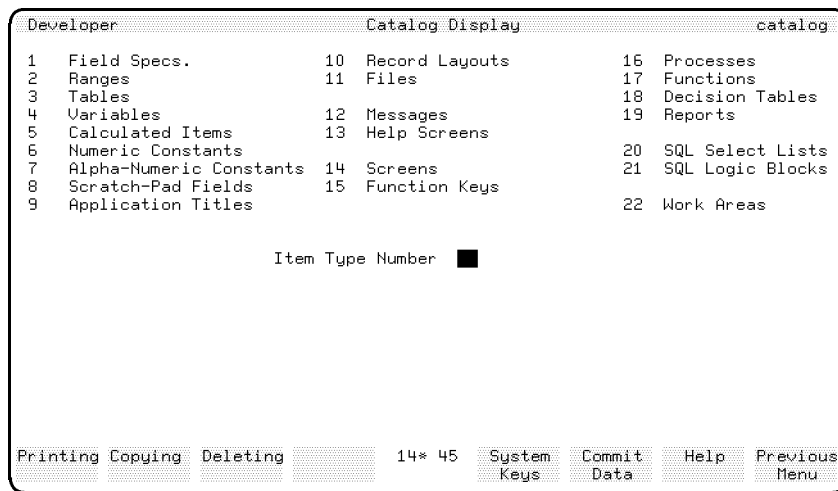
HP HP ALLBASE/4GL does not delete the item until you press the **Commit Data** function key.

In some cases, HP HP ALLBASE/4GL cannot recover space in its internal name lists after you delete an item. To recover the name list space, you must generate the entire application by selecting the *All* option on the generates menu. This menu is accessible from the developer main menu.

Catalog Display Screen

This screen allows you to display the names of existing application components on the terminal screen.

Screen Image



The screenshot shows a terminal window titled 'Developer Catalog Display catalog'. It displays a list of application components with their corresponding item type numbers. The list is as follows:

Item Type Number	Component Name
1	Field Specs.
2	Ranges
3	Tables
4	Variables
5	Calculated Items
6	Numeric Constants
7	Alpha-Numeric Constants
8	Scratch-Pad Fields
9	Application Titles
10	Record Layouts
11	Files
12	Messages
13	Help Screens
14	Screens
15	Function Keys
16	Processes
17	Functions
18	Decision Tables
19	Reports
20	SQL Select Lists
21	SQL Logic Blocks
22	Work Areas

Below the list, there is a prompt: 'Item Type Number' followed by a solid black square. At the bottom of the screen, there are several menu options: 'Printing Copying Deleting', '14* 45', 'System Keys', 'Commit Data', 'Help', and 'Previous Menu'.

Field Description

Item Type Number. Enter the number for the item type you want to display, and then press the **Commit Data** function key.

HP HP ALLBASE/4GL displays the names of the existing components of the selected type.

If the list of components extends to more than one screen, press **Return** to display the next page of the list.

Function Keys

- f1** **Printing** Cancels the current screen and displays the catalog printing screen.
- f2** **Copying** Cancels the current screen and displays the catalog copying screen.
- f3** **Deleting** Cancels the current screen and displays the catalog deletions screen.

Additional Information

HP ALLBASE/SQL Table and Select List Display

Internally, HP HP ALLBASE/4GL regards SQL tables and select lists as being files. When you select the *Files/SQL Tables* option (option number 11) on this screen, HP HP ALLBASE/4GL displays the names of all files, SQL tables, and select lists defined for the application.

Select option number 20 (*SQL Select Lists*) to obtain a list of select list names only.

Additional Utilities

HP HP ALLBASE/4GL includes a number of utility programs. These programs are summarized below.

HP4STOA	This utility extracts the source information for an application from the HP HP ALLBASE/4GL S-files and writes the information to an ASCII text file.
HP4ATOS	This utility takes the source code for an application from an ASCII text file and writes the information into a set of S-files.
HP4REMK	This utility rebuilds the indexes for the HP HP ALLBASE/4GL system files (S-files).
HP4REOD	This utility rebuilds the HP HP ALLBASE/4GL S-file data files.
HP4TUPLD	This utility extracts dictionary definitions from an HP TurboIMAGE/iX schema and writes the information to an ASCII text file in a format that can be used by HP4ATOS.

The HP4TUPLD utility is only useful if you are using the HP TurboIMAGE/iX data manager for your applications. Refer to chapter 10, HP TurboIMAGE/iX for details about HP4TUPLD.

The other utilities are described below.

HP4STOA and HP4ATOS

The *HP4STOA* (S-files to ASCII) utility extracts clean source code from an application's S-files and writes the source to an ASCII text file. It reports the inconsistencies it has encountered to \$STDLIST.

The *HP4ATOS* (ASCII to S-file) utility allows you to load the clean source code from the ASCII text file into a set of S-files.

To use these utilities, you must exit from HP HP ALLBASE/4GL and run the utilities directly from MPE/iX.

System Variables

To use HP4STOA and HP4ATOS, you must set the HP4SPATH system variable. The HP4SPATH system variable identifies the location of the group and account containing the HP HP ALLBASE/4GL S-files.

Keywords File

The file *HP4GLKWD.PUB.SYS* contains the names of all keywords, objects, and symbols (used to mark the start and end of objects) used by HP4STOA and HP4ATOS. To use these utilities, *HP4GLKWD.PUB.SYS* must exist.

Running HP4STOA

The syntax for running HP4STOA from the MPE/iX prompt is:

```
HP4STOA " [options] [> FILENAME]"
```

or

```
RUN HP4STOA.PUB.SYS [; INFO = "options"] [; STDLIST = FILENAME]
```

Options can be any of the options listed below. Options may be combined, but all objects will be unloaded in the order specified.

HP4STOA sends its output to \$STDLIST unless you redirect output to a file. *FILENAME* can be any name you choose. If you redirect output to a file, you must create the file before running HP4STOA, with an adequate record size and an adequate number of records specified.

Command Line Options.

Option	Description
-a [<i>application_name</i>] [: <i>dev_sec_code</i>]	The application name or version name to be unloaded. If a development security code exists, it must accompany the application or version name. You must specify an application name, except when you wish to unload system class objects. In this case, do not specify an application name, but specify <i>administ</i> as the user name. Refer to pages 13-18 for a full list of system class objects.

13-14 Utilities

- b** Prints out only the names and types of all objects that are unloaded. If you don't select this option, all object details are output.
- c** As much data as possible about any S-file inconsistencies will be printed. This option cannot be used with the **-b** option.
- d** Checks all default values and gives warnings for unusual default settings. This option will produce many warnings when run on older S-files.
- e** Disables error checking. Only use this option with error-free S-files. This option speeds up output.
- i** Messages regarding S-files with incomplete descriptions are not displayed. This option is best used when unloading run-time S-files.
- k** Ignores namelists in S-files for non-list objects. Instead it creates its own list by scanning the S- files for appropriate records. HP4STOA does not differentiate between select lists and files, so this option will not build the namelist for SQL select lists.
- n** No output is produced to an ASCII file. Details are reported to \$STDLIST. Use this option to check for S-file inconsistencies.
- o** *object type* [, *object list*] Allows you to specify objects that you wish to unload. You may specify the type of object by using an object type code (refer to page 13-16). To specify that objects matching particular character strings (regular expressions) are unloaded, list the character strings. This list of strings is called the object list. If you don't include an object list, HP HP ALLBASE/4GL will unload all objects within the specified object type. You can also use an asterisk to specify 'all objects'.
- u** [*user_name*] [*:user_password*] Specifies the name of a registered user of the application. If a password exists for the user, it must be included with the user name. You must always

specify a user name when you run HP4STOA. When you specify system class objects in an HP4STOA command you must specify *administ* as the user. Refer to page 13-18 for a full list of system class objects.

- w** Disables warning message output.
- v** *object type* [,*object list*] Allows you to specify objects that you do not wish to unload. You may specify the type of object by using an object type code (refer to page 13-16). You may specify that objects matching particular character strings (regular expressions) are not unloaded. To do so, list the character strings in an object list. If you don't include an object list, all objects within the specified object type will not be unloaded. You can also use an asterisk to specify 'all objects'.
- z** Produces dummy dates and times in the output.

An example of running HP4STOA is:

```
HP4STOA "-udevelopr:password -aexample -od* -olfp > CLEANFIL"
```

or

```
RUN HP4STOA.PUB.SYS ; INFO = "-udevelopr:password -aexample -od*  
-olfp" ; STDLIST = CLEANFIL
```

Both commands load all clean dictionary objects, functions and processes from the application *example* into the CLEANFIL ASCII file.

Object Types. An object type normally consists of two letters representing a class and type of object. The codes for all object types are listed below.

In addition, you may combine object types by listing the class letter and specific types. For instance, *-olfp* specifies functions and processes, *-ol** specifies all logic classes, and *-o*f* specifies functions and field specifications.

If no object types or classes are specified, the entire application's S-files are loaded into an ASCII file.

13-16 Utilities

Dictionary Class

da	Application title (prompt)
dc	Constant (alphanumeric)
dd	Database item: file,SQL, table
df	Field specification
dg	Range
dh	Help screen
dl	SQL select list
dm	Message
dn	Numeric
dr	Record
ds	Scratch pad
dt	Table
du	Calculated item
dv	Variable
dw	Work area

Screen Class

sk	Function (soft) keys
ss	Screen

Logic Class

ld	Decision table
lf	Function
lp	Process
ls	SQL logic block

Report Class

rr	Report
----	--------

System Class

- xa Application headers
- xb HP TurboIMAGE/iX database names
- xd Terminal display attributes
- xm System master
- xp HP TurboIMAGE/iX database access parameters
- xs Synonyms
- xt Master titles
- xu Users
- xv Version headers

When you run HP4STOA, an exit value is returned to indicate the success of the program. For a list of these values, refer to the Exit Values section in this chapter.

Running HP4ATOS

HP4ATOS allows you to load a clean application from an ASCII file into a set of S-files.

The syntax for running HP4ATOS at the MPE/iX prompt is:

```
HP4ATOS " [options] < FILENAME "
```

or

```
RUN HP4ATOS.PUB.SYS [; INFO = "options" ] [; STDIN = FILENAME]
```

Options can be any of the options listed below. Options may be combined, but all objects will be loaded in the order specified. If no options are specified, the entire contents of the ASCII file are loaded into S-files.

FILENAME is the name of the ASCII file you wish to load into S-files.

Command Line Options.

Option	Description
-a [<i>application_name</i>] [: <i>dev_sec_code</i>]	The application name or version name to be loaded. If a development security code exists, it must accompany

13-18 Utilities

the application or version name. You *must* specify an application name, except when you wish to load system class objects into the S-files. In this case, do not specify an application name, but specify *administ* as the user name. Refer above for a full list of system class objects.

- e** Ensures that objects that already exist in the S-files are not overwritten.
- f** Forces the original date and time fields of an object to be stored. If this option is not specified, all dates and times will be stored as the current date and time.
- n** No output is produced to the S-files, only to the screen. Use this option to check ASCII files for inconsistencies. If you use this option to check ASCII files of applications that do not exist in the S-files, errors may occur because HP4ATOS assumes that previous objects in the ASCII file have been loaded.
- u** [*user_name*]
[:*user_password*] Specifies the name of a registered user of the application. If a password exists for the user, it must be included with the user name. You must always specify a user name. If you wish to load system class objects into S-files, you must specify *administ* as the user, and you must not specify an application. Refer to page 13-18 for a full list of system class objects.

An example of running HP4ATOS is:

```
HP4ATOS "-udevelopr -aexample < CLEANFIL"
```

or

```
RUN HP4ATOS.PUB.SYS ; INFO = "-udevelopr:password -aexample "; STDIN =  
CLEANFIL
```

This loads the clean S-file information from CLEANFIL into the current S-files for the *example* application. HP4ATOS produces warning messages when an item in the ASCII text file corresponds in name to an item in the S-files, but the definitions of the items differs. Warning messages are also produced for other errors found in the definitions of items. For a list of exit values returned by HP4ATOS, refer to Exit Values below.

Exit Values

HP4STOA and HP4ATOS both produce one of the following exit values after being run. These values indicate the status of the completed job:

- 0 Job completed correctly with no inconsistencies.
- 1 Command line is incorrect. Check that you have the correct command format.
- 2 S-files or ASCII files contain inconsistencies.
- 3 Operating system failure. Perhaps there was insufficient memory available.

HP4REMK

The HP4REMK utility remakes the indexes for the HP HP ALLBASE/4GL system files (S-files).

To use this utility, you must exit from HP HP ALLBASE/4GL and ensure that no other users are currently accessing the system.

You must set the MPE/iX variable HP4SPATH before you can use the HP4REMK utility. The HP4SPATH variable identifies the location of the group and account containing the HP HP ALLBASE/4GL S-files. For example, if the S-files are in the HP4S.HP4GL group and account, you must set the HP4SPATH variable as follows:

```
SETVAR HP4SPATH "HP4S.HP4GL"
```

To run HP4REMK, enter the command HP4REMK at the MPE/iX prompt. The HP4REMK utility then displays a message asking you to enter the number of the file to be remade. The HP HP ALLBASE/4GL S-files are numbered from 1 to 11 inclusive. The following table indicates the main contents of each S-file.

13-20 Utilities

S-file	Contents
s01	Terminal attributes
s02	End users, developers, and groups
s03	Application and version definitions
s04	Descriptions
s05	Messages, master titles, and prompts
s06	Help screens
s07	Field specifications
s08	System records and some list items
s09	List items (constants, variables, numerics, calculated items, validation items, data files, scratch-pad fields).
s10	Source
s11	Generated items

You can also run the HP4REMK program by specifying the file numbers on the command line in the format:

```
HP4REMK "[-1] [-s] [n[-m]]"
```

The *-l* option produces a long listing of each block that is read and remade. The *-s* 'silent' option runs HP4REMK without the progressive count of blocks read.

In this command, *n* and *m* are numbers between 1 and 11 indicating the S-file or range of S-files to be remade.

HP4REOD

Caution



You must use the HP4REMK utility to rebuild the HP ALLBASE/4GL S-file indexes before you attempt to use the HP4REOD utility. If you don't remake the file indexes before attempting to reorder the files, you may corrupt the HP ALLBASE/4GL S-files.

The HP4REOD utility compacts the HP HP ALLBASE/4GL S-files to recover disk space occupied by logically deleted records.

To use the HP4REOD utility, exit from the HP HP ALLBASE/4GL system and ensure that no other users are accessing the system.

To run HP4REOD, you must set the MPE/iX variable HP4SPATH. (refer to HP4REMK above.) After you have used the HP4REMK utility to rebuild the S-file indexes, start the HP4REOD utility by entering the command **HP4REOD** at the MPE/iX prompt.

HP4REOD then displays a message asking you to enter the number of the file to be reordered. The HP HP ALLBASE/4GL S- files are numbered from 1 to 11 inclusive.

You can also run the HP4REOD program by specifying the file numbers on the command line in the format:

```
HP4REOD "[-s] [n[-m]]"
```

The *-s* 'silent' option runs HP4REMK without the progressive count of blocks read.

In this command, *n* and *m* are numbers between 1 and 11 indicating the S-file or range of S-files to be reordered.

Checking S-files for Inconsistencies

To identify and correct any S-file inconsistencies, complete the following instructions. These instructions assume that you are familiar with the HP4STOA, HP4ATOS, HP4REMK, and HP4REOD utilities, which you will need to use in these procedures.

Caution



Be sure to use the version of these utilities that corresponds to the version of your S-files. Using a mixed version could result in inconsistencies being reported that don't exist and in some inconsistencies not being reported.

Each step provides a template for the command you will need to enter, but you must replace the slanted text parameters with the suitable names. For more

13-22 Utilities

information about the parameters, refer to the description of the appropriate utility, in this chapter.

Before you start, you should create a file to which you will redirect HP4STOA output. You must create the file before running HP4STOA, with an adequate record size and an adequate number of records specified.

Caution

Backup your S-file group before attempting to correct your S-files. This ensures that if you inadvertently destroy your S-files, they can be recovered.

1. At the MPE/iX prompt, use HP4STOA with the `-n` option to produce a list of S-file inconsistencies in your application.

```
RUN HP4STOA.PUB.SYS ; INFO = "-uuser_name:user_password  
-aapplication_name:dev_sec_code -n" ; STDLIST = FILENAME
```

At this point you may decide to delete the inconsistencies, or you may wish to repair the inconsistencies. If you wish to repair the inconsistencies, refer to Repairing Inconsistencies later in this chapter.

If you decide to delete the inconsistencies, you should make a backup of your application. If you do not make a backup of your application, any objects with inconsistencies will be irretrievable.

2. Use HP4STOA without the `-n` option to create an ASCII file containing the consistent portions of the application. The inconsistent objects are not unloaded.

```
RUN HP4STOA.PUB.SYS ; INFO = "-uuser_name:user_password  
-aapplication_name:dev_sec_code" ; STDLIST = FILENAME
```

3. Log in to HP HP ALLBASE/4GL as the administrator, and delete the application using the administrator deletions screen. All inconsistent objects are deleted with the application.
4. Redefine the application using the application definition screen.

5. Use HP4ATOS to load the clean application in the ASCII file back into the application's S-files.

```
RUN HP4ATOS.PUB.SYS ; INFO = "-uuser_name:user_password  
-aapplication_name:dev_sec_code" ; STDIN = FILENAME
```

6. Log in to HP HP ALLBASE/4GL as a developer and regenerate all objects in your application.
7. Then use HP4REMK and HP4REOD to remake and reorder the indexes for the S-files.

```
HP4REMK 1 -11
```

```
HP4REOD 1 -11
```

8. You may now wish to recreate and regenerate any inconsistent objects that were deleted from the application. You should also correct and regenerate any objects that caused a regeneration error.

Repairing Inconsistencies

Many of the error messages given by HP4STOA are self-explanatory and easy to fix. For example, the message **ERROR: Bad scroll bottom specified, object screen, test_scr** indicates that the scroll bottom line value given for the *test_scr* screen is incorrect. To correct this error, correct the value and regenerate the screen. Other examples of this type of error include variables, ranges, or calculated items with illegal numbers of decimal places. Correct these types of errors in HP HP ALLBASE/4GL and recommit and regenerate the relevant objects.

Some errors are less specific, and usually indicate a severe inconsistency. An example of this type of error is **ERROR: Object not complete: S-file corruption, object report, testrep**.

For each object that causes an error of this type:

1. Log in to HP HP ALLBASE/4GL as a developer, and print out as many details about the object as possible, using the catalog printing screen. This will ensure that if you cannot repair the object, you will have the information to recreate the object later.

13-24 Utilities

2. If a message is displayed suggesting that an object is undefined, move to the screen used to define the object and try to display the object. If the object is retrieved, use the screen printing facility to print out as much information as possible. If the object still cannot be retrieved, enter a new header or description (if appropriate). Then return to the Catalog Printing screen and try again to print the object details.
3. If the object can be retrieved, you may be able to repair it. Recommit header and detail information, and then generate the object (if appropriate).
4. If you cannot retrieve the object at all, delete it using the Catalog Deletion screen. If you cannot delete it, use the *-vobject_name* option when you use HP4STOA to ignore the object. When you delete the entire application, and reload it from the ASCII file using HP4ATOS, the item will not be present. You can recreate the object later in the S- files.
5. In some rare cases, you may be able to call up an object's details but you may not be able to recommit them. If this occurs, print out the object's details and delete the object. Then recreate the object (and regenerate it if appropriate).

After correcting as many inconsistencies as you can, run HP4STOA to check that the inconsistencies have been fixed. Continue the cycle of running HP4STOA and then repairing inconsistencies until all inconsistencies are removed.

FINAL TRIM SIZE : 7.0 in x 8.5 in

Application Generation and Testing

Application Generation

The following components of HP ALLBASE/4GL applications must be generated before they can be executed:

- Record layouts.
- Data screens and windows.
- Processes.
- Functions.
- Decision tables.
- Calculated items.
- Reports.
- SQL select lists.
- SQL logic blocks.
- Messages.

Generating application components converts your input to executable run-time code. In the generation process, HP ALLBASE/4GL converts the application component to an executable form, and resolves references to other application components.

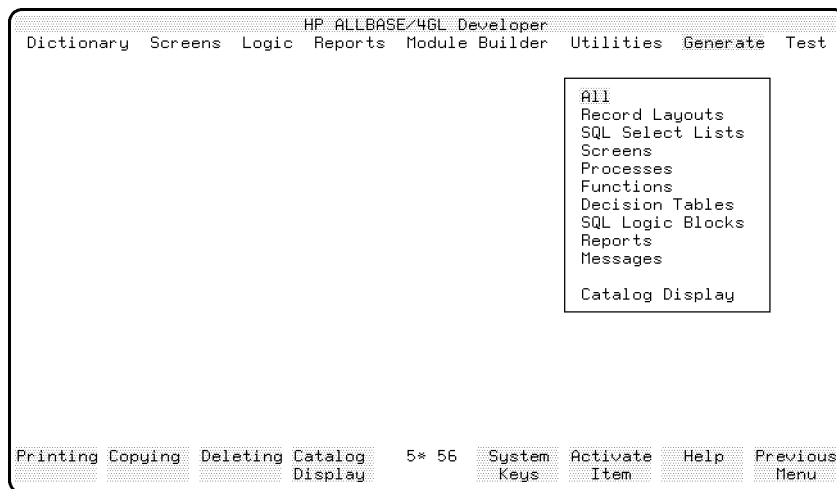
To generate components of an application, select the *Generate* option on the developer main menu. This selection displays the application generation menu. Selections on this menu allow you to generate some or all components of an application.

You can also generate application components by pressing the **Generate** function key that appears on some HP ALLBASE/4GL developer screens.

Messages and calculated items are generated automatically when you press the **Commit Data** function key on the appropriate dictionary screens.

The generation process uses an external program. HP ALLBASE/4GL calls this program automatically when you generate an application component.

Screen Image



Menu Selections

This screen allows you to generate specific application components, or the entire application. Highlight the component type that you wish to generate, and then activate the highlighted item by pressing **Return** or pressing the **Activate Item** function key.

Catalog Display. This selection displays the catalog display screen, allowing you to display the names of existing application components. Refer to chapter 13 for details.

14-2 Application Generation and Testing

Function Keys

- (f1) **Printing** Cancels the current screen and displays the catalog printing screen.
- (f2) **Copying** Cancels the current screen and displays the catalog copying screen.
- (f3) **Deleting** Cancels the current screen and displays the catalog deletions screen.
- (f4) **Catalog Display** Cancels the current screen and displays the catalog display screen.

Additional Information

The selections on this screen allow you to generate groups of components in an application. Typically, you would use this screen to generate a group of application components after you have made a change to a component such as a field specification or a record layout, and you want to make the change effective throughout the application. To generate an individual component, use the **Generate** function key on the appropriate item screen.

The generation process does not start until you activate the highlighted item on the menu. Once generation starts, HP ALLBASE/4GL displays a series of progress messages indicating the status of the generation process.

If any errors are detected during generation, the generate program displays a warning message, and generation pauses until you press **Return**. The generation process restarts after about 20 seconds if the system does not receive any input.

Any changes you make to an application component that requires generation will not be effective until you generate the component.

The following application components don't need to be generated:

- Menus.
- Field specifications.
- Validation ranges.
- Validation tables.

- File definitions.
- Work area definitions.
- Variables.
- Numeric and alphanumeric constants.
- Scratch-pad field name declarations.
- Function key sets.
- Help screens.
- Application titles.

Changes to any of the above components except field specifications are automatically effective throughout the application.

Note

If you make a change to a dictionary field specification or a record layout, you must generate all other application components that use the field specification or record layout before the change becomes effective.

During generation, HP ALLBASE/4GL checks references to some other application components and displays error messages if the referenced components don't exist. However, HP ALLBASE/4GL does not check references to the following items:

- Data screens and windows.
- Menus.
- Processes.
- Functions.
- Decision tables.
- Reports.
- SQL select lists.
- SQL logic blocks.
- Messages.

14-4 Application Generation and Testing

- Help screens.
- Function keys sets.

As a result, HP ALLBASE/4GL may successfully generate an application, and then display error messages at run-time if the application contains references to any of these items that have not been defined.

SQL Component Generation

For some commands in SQL logic blocks, HP ALLBASE/4GL creates stored sections in the database environment. These sections are stored in a module in the database environment.

The module has the name *owner_group.name* where *owner_group* is the SQL owner group defined on the application definition screen in the administrator, and *name* is the name of the application or version.

When you select the *All* option or the *SQL Logic Blocks* option on this screen, HP ALLBASE/4GL issues a DROP MODULE command, and then rebuilds the database module with the newly generated SQL sections. HP ALLBASE/4GL also builds a file containing the definition of the database module and the definitions of all SQL tables required by the application. The name of this file is the name of the application or version. The file is created in the account and group specified by the current value of the variable HP4DBMPATH. If HP4DBMPATH is not defined, HP ALLBASE/4GL does not create the database module file.

HP ALLBASE/4GL uses the variable HP4DBMSIZE to determine the file limit for the database module file. HP ALLBASE/4GL uses a default limit of 4095 records if HP4DBMSIZE is not defined.

Generate Error Log File

HP ALLBASE/4GL logs all error messages initiated during generation to a temporary file in the temporary domain of your log-on group and account. The formal file designator for this file is HP4GNERR.

If desired, you can use a file equation to direct the generate log to a permanent disk file or a suitable printer.

Application Testing

HP ALLBASE/4GL allows you to test an application from within the developer. You can also test the application as an end user.

To test an application from within the developer, select the *Test* option on the developer main menu. This selection runs the application as it would appear to the end user. HP ALLBASE/4GL returns you to the developer main menu when the application finishes, or terminates the application if a fatal error occurs.

To test the application as an end user, you must return to the HP ALLBASE/4GL sign-on screen, and then sign on to your application as an end user. To do this, you must have an HP ALLBASE/4GL end user name, and this name must be included in the list of users for your application. If you have signed on to the application as an end user, HP ALLBASE/4GL returns you to the sign-on screen at the end of the application. If a fatal error occurs in end-user mode, HP ALLBASE/4GL returns you to MPE/iX.

Using Databases during Application Testing

If the application uses HP ALLBASE/SQL or HP TurboIMAGE/iX databases, HP ALLBASE/4GL automatically connects to the database during application testing when you execute a process or function in the application that requires access to the database. HP ALLBASE/4GL releases from the database when you exit from application testing mode.

Menu Bypass Facility

While you're running an application in the application testing mode, you can use the menu bypass facility to execute any component in the application. To use the bypass facility, enter / at any menu in the application. HP ALLBASE/4GL displays a message requesting you to enter an action. To enter an action, enter an action prefix followed by a hyphen and the name of the action, and then press **Return**. HP ALLBASE/4GL executes the action as though it was selected directly from a menu.

14-6 Application Generation and Testing

The action prefixes are:

Prefix	Action
B-	Background process
D-	Menu or data screen
F-	Function
P-	Process
R-	Report
H-	Help screen
X-	External program

Training Mode

While you're testing an application, you can enable the *Training Mode* if you don't want to write data to the application's data files. The application data files are still available for reference purposes.

To enable the training mode, press the **System Keys** function key, and then press the **More Keys** function key. You can now activate training mode by pressing the **Training Mode** function key. An asterisk in the function key display indicates that training mode is active.

Using Trace Mode

HP ALLBASE/4GL has a trace mode that allows you to run an application on a step-by-step basis. When trace mode is active, the system displays the current logic block and step on the bottom two lines of the screen. The system pauses at each step for about three seconds, and then proceeds to the next step.

You can only use trace mode while you are running the application from within the developer using the application testing option. You cannot use trace mode if you are running the application as an end user.

To activate the trace mode, press the **System Keys** function key, and then press the **More Keys** function key. You can now activate trace mode by

pressing the **Trace Mode** function key. To cancel trace mode, press the **Trace Mode** function key again. An asterisk in the screen display for the **Trace Mode** function key indicates that trace mode is active.

Trace mode can also be activated by moving a non-zero value into the *LTRACE communication area from within a logic block. To turn trace mode off, move a zero into *LTRACE. See chapter 4 for more information on the *LTRACE communication area.

While you're using trace mode, HP ALLBASE/4GL records a trace log file. The trace log file is a file in the temporary domain of your log-on group and account. The formal file designator for this file is HP4TRACE. The default for this file is a file in the temporary domain of your current log-in group and account.

If desired, you can use a file equation to direct the generate log to a permanent disk file or a suitable printer.

If required, you can set the trace message delay to any value (including zero seconds) by using the *-m* option on the HP ALLBASE/4GL command line. The command line becomes:

```
HP4GL "-in"
```

where *n* is an integer number ranging from 0 to 9 to specify the message delay in seconds.

Unloading Applications

Unloading completed applications for use on another HP ALLBASE/4GL system is a system administration task. Refer to the *HP ALLBASE/4GL Developer Administration Manual* for details of the application loading and unloading utilities.

14-8 Application Generation and Testing

A

Native Language Support

HP ALLBASE/4GL Collating Sequence

HP ALLBASE/4GL uses NLS (native language support) to control the collating sequence. This sequence depends on the setting of the NLATALANG jcw, or job control word. If this jcw is not set, then the “American” NLS collating sequence is used. If this sequence is not present, then N-computer is used.

FINAL TRIM SIZE : 7.0 in x 8.5 in

HP ALLBASE/4GL Versions

Versions

This appendix describes the procedures for creating and maintaining versions of HP ALLBASE/4GL applications.

A version is a set of additional parameters that operate in conjunction with a base application. When HP ALLBASE/4GL runs a version, it effectively runs a modification of the base application. Typically, you would use a version to provide an application that has been modified to suit the needs of a particular end user, or group of end users.

A version is **not** a modified copy of the base application. Instead, it's a set of parameters that represent the difference between the base application and the modification that you want to present for the selected users. A version cannot be run unless the base application exists on the same HP ALLBASE/4GL system. Also, versions cannot be unloaded or loaded without the base application.

Defining Versions

Defining a version and the list of eligible developer names and end user names for the version is a system administrator task. The system administrator can assign a development security code for a version.

The *HP ALLBASE/4GL Developer Administration Manual* describes these procedures.

Developing Versions

For the most part, the techniques for developing versions with HP ALLBASE/4GL are identical to the techniques for developing any other application. All the facilities of the HP ALLBASE/4GL developer are available to you while you are developing a version.

To develop a version of an application, sign on to HP ALLBASE/4GL as normal, and then enter the version name in the *Application or Version* field on the HP ALLBASE/4GL sign-on screen.

While you are developing a version, all the components of the base application are accessible, although they appear in secured form. Although base application components are accessible while you are developing a version, a component does **not** become part of the version unless you press the **Commit Data** function key. There is no need to commit a component from the base application in a version unless you modify the component.

To define a modified component for a version, call up the component of the base application on the appropriate definition screen and enter **N** in the *Secured* field if the item is secured. You can now make any changes that are required, and commit the component as part of the version by pressing the **Commit Data** function key.

You can also use the appropriate definition screens to define new components for the version. The catalog copying screen in the utilities menu allows you to copy components from other applications or versions.

When you define a new or modified component for a version, (for example, adding a further option to a menu and the necessary screen and logic to support the option), HP ALLBASE/4GL stores the additional information with the version. If you modify an existing item to create a new item for the version (for example, adding some further fields to an existing screen) HP ALLBASE/4GL takes a copy of the original item from the base application and stores the modified item with the version.

With the exceptions listed below, any subsequent changes made to items in the base application are automatically reflected in versions of the application, provided a modified copy of the same items doesn't already exist in the version. If you have defined a modified item in the version, changes to the same item in the base application have no effect on the item in the version.

B-2 HP ALLBASE/4GL Versions

Limitations

Under some circumstances, changes to the following items in the base application will **not** be reflected in versions of the application:

- Validation ranges.
- Validation tables.
- Alphanumeric constants.
- Numeric constants.
- Variables.
- Calculated items.
- Data file definitions.
- Work area definitions.
- Scratch-pad name declarations.

HP ALLBASE/4GL stores these items in common internal lists. If you make a change to any **one** of these items in any particular type, HP ALLBASE/4GL creates a copy of all of the items of the same type in the version. Subsequent changes to these items in the base application will not be reflected in the version.

Changes to these items will only be reflected in versions of the base application if you have not made any changes to *any* items of the same type in the version.

Note



A version may contain generated components (for example, functions and processes) that refer to dictionary items in the base application. Changes to the dictionary items in the base application will not be reflected in generated components in the version unless you generate the components again.

Unloading and Loading Versions

Unloading and loading versions is a system administrator task. A version cannot be unloaded or loaded without the base application. Up to 10 versions can be unloaded or loaded with the base application.

The *HP ALLBASE/4GL Developer Administration Manual* describes the procedures for unloading and loading applications and versions.

Version Data File Reformat

HP ALLBASE/4GL provides a mechanism for automatically reformatting KSAM data files in the HP ALLBASE/4GL run-time environment. Data file reformatting may be necessary if you have changed the format of a record layout for a data file by changing the number of fields, changing the order of fields, or changing the details of fields.

The HP ALLBASE/4GL run-time environment data file reformatting system relies on the structure file (*xxxxS*, where *xxxx* is the name of the data file) associated with each data file, and a file-structure file (in the group identified by the current value of the MPE/iX variable HP4FSPATH) created by the run-time application unloading procedure. During the subsequent application loading procedure, HP ALLBASE/4GL compares the existing file format, as described in the existing structure file with the new file format contained in the file-structure file. If the two formats are different, HP ALLBASE/4GL reformats the existing data files in the run-time environment.

The automatic data file reformatting system cannot simultaneously reformat data file for both a base application and a version. To overcome this limitation, the run time file reformatting system only reformats the data files for the version if the version file format is different from the base application file format. To reformat data files for the base application only, the application must be unloaded by itself without any versions.

This means that you may need to unload (and subsequently reload) an application twice if a version of the application uses data files with a different format. By initially unloading and then loading the application without the version, you can reformat the base application data files. By subsequently unloading, and then reloading the application with the version, you can reformat the data files for the version.

B-4 HP ALLBASE/4GL Versions

Don't attempt to unload more than one version with an application if all the versions use different file formats.

Refer to the *HP ALLBASE/4GL Developer Administration Manual* for more information about run-time data file reformatting.

FINAL TRIM SIZE : 7.0 in x 8.5 in

C

Data Manager Error Messages

This appendix lists the error messages that may be generated by the MPE/iX file system, the KSAM file manager, HP ALLBASE/SQL, or HP TurboIMAGE/iX while you run an application.

Whenever an error occurs, HP ALLBASE/4GL displays both the error description and its HP ALLBASE/4GL error number. Some file errors cause HP ALLBASE/4GL to abort the application. These errors generally indicate a problem with the file or the MPE/iX file system and should be investigated carefully.

This appendix lists the file errors in numerical sequence.

Wherever an error message contains variable information such as the file name, its description is listed as part of the message.

MPE/iX Miscellaneous Errors

These errors are caused by problems with the MPE/iX file system. Some of these errors are fatal and cause HP ALLBASE/4GL to abort the current application.

This appendix does not list all the MPE/iX errors. If you need further information about MPE/iX errors, consult the appropriate MPE/iX reference manual.

19000 HP ALLBASE/4GL data error - Access to file <file name>
has failed

Error type message. This error is caused by an MPE/iX or HP ALLBASE/4GL file system problem that has been detected by the HP ALLBASE/4GL data manager.

- 19002** File: <file name> does not exist
Error type message. Check the name, group, and account that you have specified for the application data file.
- 19009** Invalid file number for: <file name>
Abort type message. This error indicates a problem with the HP ALLBASE/4GL file system. It should be documented and reported to HP.
- 19013** Access permission denied to file: <file name>
Warning message. You have attempted to access a file in a way that contravenes the file's access restrictions. For example, you may be trying to modify a file to which the application has read-only access.
- 19024** File: <file name> cannot be opened - too many open files
Abort type message. Each MPE/iX system has a limit to the number of files which may be opened by any user. If the application attempts to open too many data files this error will occur. Before this error is encountered HP ALLBASE/4GL tries to close all its own files first to enable the application to continue. If this error does occur, you can use the logic command FILE *CLOSE to close each file that is not needed at the time. If you close a file that is needed again later in the application, HP ALLBASE/4GL automatically re-opens the file.
- 19027** Unable to write to file: <file name> - It is too big
Abort type message. You have reached the file limit defined for the file.

C-2 Data Manager Error Messages

Data File and Data Set Errors

These errors are caused by problems encountered by the KSAM file manager, the serial file data manager, or the HP TurboIMAGE/iX data manager while attempting to read or update an application's data files. Some errors are fatal and cause HP ALLBASE/4GL to abort the current application. Where a non-fatal condition is encountered, the error number is placed into the communication area field *IOSTATUS. You can interrogate the contents of *IOSTATUS from the application logic.

The FILE logic command allows you to specify an optional error command that is executed if an error occurs while the file command is being executed. If you specify an error command for a KSAM file, error messages associated with errors 19 100, 19 110, and 19 111 will not be displayed on the terminal screen. All other KSAM errors are displayed.

If you don't specify an error command in the FILE command, all error messages are displayed.

The messages in this section may occur for any one of the data managers listed above. However, there are errors that only occur for the HP TurboIMAGE/iX data manager. These errors are listed on page C-11.

- 19000** HP ALLBASE/4GL data error - access to file <file name>
has failed
- Error type message. This error is caused by an MPE/iX or HP ALLBASE/4GL file system problem that has been detected by the HP ALLBASE/4GL data manager.
- 19100** The primary key value already exists, and duplicates are
not permitted for file <file name>
- Error type message. You are trying to add a record to a file and the record has the same primary key value as a record already on the file. That is, a duplicate key error has occurred. This error will only occur when the file's primary key has been defined as having unique values.
- 19101** Internal HP ALLBASE/4GL error - attempt to access
unopened file <file name>

- Abort type message. You are trying to access a file that has not been opened by HP ALLBASE/4GL. This error should be documented and reported to HP.
- 19102** Internal HP ALLBASE/4GL error - Bad argument to call for file: <file name>
- Error type message. This indicates an HP ALLBASE/4GL data manager error. It should be documented and reported to HP.
- 19103** The key descriptor of <file name> is outside defined range
- Error type message. One or more elements that make up the key description is outside the range of acceptable values for that key. The file may need reformatting.
- 19105** Data storage error - the contents of file: <file name> are corrupt.
- Abort type message. The format of the data file has been corrupted, possibly as the result of a system failure.
- 19106** Data storage error - Unable to update index of file: <file name>
- Abort type message. This error may occur during data file creation or data file reformatting while an application is under development. HP ALLBASE/4GL requires exclusive access to the file during file creation or reformatting.
- 19107** Unable to access record in file: <file name> - It is locked
- Error type message. This is a warning message only and indicates that there has been an attempt to access a record that is locked by another user of that file. HP ALLBASE/4GL tries a number of times to see if the record is still locked before this error is returned.
- 19108** It is not possible to re-add the index in file: <file name>

C-4 Data Manager Error Messages

Abort type message. This error may occur during data file creation or data file reformatting while an application is under development. This error should be documented and reported to HP.

19109 Data storage error - unable to delete Primary key index for file: <file name>

Warning type message. This error may occur during data file reformatting while an application is under development.

19110 Data storage warning - beginning or end of file <file name>

Warning type message. This message indicates that the end or beginning of the file has been reached during a FILE *NEXT or a FILE *PREVIOUS logic command.

19111 Data storage warning - key value specified does not exist in file <file name>

Warning type message. This message indicates that no record could be found that matched the *KEY= argument in a FILE *READ, FILE *FIND, or a VALIDATE logic command.

19112 Data storage error - record has not been read for file <file name>

Error type message. For KSAM files and HP TurboIMAGE/iX data sets, the FILE *MODIFY command can only operate on the most recent record accessed. For KSAM files and HP TurboIMAGE/iX data sets that have non-unique primary keys, the FILE *DELETE command can only operate on the most recent record accessed.

19113 File <file name> is exclusively locked

Error type message. The file is currently being accessed by another user who has exclusive access to the file. This error may occur if the file is being reformatted.

19114 File Name <file name> is too long

Data Manager Error Messages C-5

- Warning type message. The HP ALLBASE/4GL file name (excluding the group and account name if specified) cannot be longer than seven characters.
- 19130** Unable to lock file <file name>. The file or a record is already locked
- Error type message. The current HP ALLBASE/4GL process is attempting to access a file under MODE *LOCK, and another user has already locked the file or a record in the file.
- 19451** The file <file name> does not exist
- Error type message. The KSAM file manager or the HP TurboIMAGE/iX data manager has detected that a file does not exist. Check the file, group, and account names.
- 19990** HP ALLBASE/4GL data error - unable to restore current record in file <file name>
- Error type message. This message may occur if the pointer position for the current key has been lost and the current process is attempting a *MODIFY, *DELETE, *NEXT, or *PREVIOUS file operation. This situation may occur if the value of *INDEXNO has been changed, or another user has deleted the current record.
- 19991** HP ALLBASE/4GL Record Layout size and physical record size differ
- Warning type message. This error may occur if the HP ALLBASE/4GL record layouts for the file have been changed and the file has not been reformatted. Use the data file reformat screen to reformat the file.
- 19992** Internal HP ALLBASE/4GL error - internal lock flag error
- Abort type message. This is an internal HP ALLBASE/4GL error. It should be documented and reported to HP.
- 19993** The specified *INDEXNO is greater than the number of indexes for file <file name>

C-6 Data Manager Error Messages

- Abort type message. The current value of *INDEXNO is higher than the total number of keys defined for the file.
- 19994** HP ALLBASE/4GL Revision error - the correct user data manager for the file type has not been installed
- Abort type message. This is an internal HP ALLBASE/4GL error. It should be documented and reported to HP.
- 19995** *WRITE operation may not be used because the current index of file <file name> permits duplicate values.
- Error type message. If the file has been defined as allowing duplicate values for the primary key, then any updates to the file should be performed by using the *INSERT argument to add a new record and the *MODIFY argument to change one that already exists.
- 19996** You cannot change file <file name>. It is in read mode
- Error type message. You are trying to alter the contents of a file which has not been specified as being updated in the current process. Check the MODE command and ensure that the argument is *WRITE, *LOCK, or *UNLOCK, and you have specified the correct file name.
- 19997** Internal HP ALLBASE/4GL error - invalid file operation
- Error type message. This is an internal HP ALLBASE/4GL data manager error. It should be documented and reported to HP.
- 19998** Internal HP ALLBASE/4GL error - invalid data pool number
- Abort type message. This is an internal HP ALLBASE/4GL error. It should be documented and reported to HP.
- 19999** Record Length for File: <file name> is greater than the physical record size
- Abort type message. This error may occur when the total length of the default record layout of the file is greater than the length of the record layout defined when the file was created.

Data Manager Error Messages C-7

This situation may occur if you modify a record layout without reformatting the data file.

HP ALLBASE/SQL Messages

HP ALLBASE/SQL can detect errors at the following times:

- During generation of SQL logic blocks.
- While you are using the developer application.
- While an HP ALLBASE/4GL application is being run.

HP ALLBASE/4GL can display two different types of messages related to SQL applications. Errors can be detected directly by HP ALLBASE/SQL or by the HP ALLBASE/4GL-HP ALLBASE/SQL interface. All errors detected by the HP ALLBASE/4GL-HP ALLBASE/SQL interface have error numbers between 60 000 and 61 000.

HP ALLBASE/4GL displays all error messages, except those that are detected at run time, in the message window on the current screen.

If HP ALLBASE/SQL detects an error at run time, HP ALLBASE/4GL places the text of the error message in the communication area field *ERROR, and places the error number plus 100 000 in the communication area field *IOSTATUS.

If the SQL command that caused the error does not contain an optional error command, the text of the error message is displayed on the screen.

In some cases, a single SQL statement may cause more than one error message. Under these conditions, the first error message and error number are loaded into *ERROR and *IOSTATUS respectively at the time of error detection.

If the SQL command contains an optional error command, the error handling logic can invoke the SQLEXPLAIN command in an SQL logic block to retrieve the text and number of the next error. The SQLEXPLAIN command returns the text of the error to the *ERROR communication area field, and the error number to *IOSTATUS. (Note that you cannot specify a host variable with the SQLEXPLAIN command in an SQL logic block.)

C-8 Data Manager Error Messages

The SQLEXPLAIN command returns a null string after the last error has been returned.

Subtract 100 000 from the HP ALLBASE/4GL error message number to determine the HP ALLBASE/SQL error message number. Refer to the *HP ALLBASE/SQL Message Manual* for the causes and suggested remedial actions for HP ALLBASE/SQL error messages.

HP ALLBASE/4GL-HP ALLBASE/SQL Interface Error Messages

The HP ALLBASE/4GL-HP ALLBASE/SQL interface can display the following error messages.

- 60000** This application has no SQL database specified.
Error type message. The application is attempting to access an HP ALLBASE/SQL database, but no database environment has been specified on the administrator application definition screen.
- 60001** Connecting to SQL database '<database name>'
Information type message. HP ALLBASE/4GL displays this message when an application first accesses an HP ALLBASE/SQL database. Once the application has connected successfully to the database, the connection is retained until the user terminates the application.
- 60010** Unable to access SQL message catalog.
Error type message. HP ALLBASE/4GL has not been able to find or access the HP ALLBASE/SQL message catalog.
- 60011** You have an SQL transaction open, continue(y/n) [n]?
Query type message. You have tried to invoke HP ALLBASE/QUERY while an HP ALLBASE/SQL transaction is still open. If you enter Y, HP ALLBASE/4GL will display error number 60012 to determine whether you wish to commit the SQL transaction. If you enter N, HP ALLBASE/4GL will cancel the action and HP ALLBASE/QUERY will not be invoked.

- 60012** Do you want to perform an SQL COMMIT WORK (y/n) [y]
Query type message. Enter Y to terminate and make permanent all HP ALLBASE/SQL work since the previous explicit or automatic BEGIN WORK. If you enter N, all of this work will be rolled back.
- 60110** Data storage warning - end of file: <filename>
Warning type message. This message occurs when a FILE *NEXT command encounters the end of file for the active set retrieved by a SELECT command. If the SELECT command does not retrieve any records, the first FILE *NEXT command returns this error condition.
- 60112** Data storage error - record has not been read for file: <filename>
Error type message. You cannot use the FILE logic command to delete a record, and you cannot use the WHERE CURRENT OF clause in an SQL logic block, unless you read a record to position the cursor.
- 60150** The requested file operation cannot be performed on SQL select lists
Error type message. You cannot modify records or insert records in SQL select lists.
- 60151** The requested FILE *DELETE <filename> operation is out of sequence or cannot be performed at this time
Error type message. To delete a record with a FILE *DELETE command, you must:
- Use a SELECT command in an SQL logic block to open a cursor.
 - Use a FILE *NEXT command to position the cursor on the record to be deleted.
- 60152** The requested FILE *FIRST <filename> operation is out of sequence or cannot be performed at this time

C-10 Data Manager Error Messages

Error type message. This message may be displayed while HP ALLBASE/4GL is reporting from an HP ALLBASE/SQL database. The start of report function must include a SELECT command in a SQL logic block to open a cursor, but it should not include a FILE *NEXT command. The report generator issues an implicit FILE *FIRST command when it starts creating the report. (Note that the FILE *FIRST command is not permitted in HP ALLBASE/4GL logic blocks.)

60153 The requested FILE *NEXT <filename> operation is out of sequence or cannot be performed at this time

Error type message. You must use a SELECT command in an SQL logic block to open a cursor before you can use the FILE *NEXT command.

60997 HP ALLBASE/4GL internal error - invalid SQL table/select list operation.

Error type message. This is an internal HP ALLBASE/4GL error condition. It should be documented and reported to HP.

HP TurboIMAGE/iX Data Set Errors

Errors may be returned from the HP TurboIMAGE/iX data manager. The text of these messages may be found in the *ERRORDM communication area field. The error message number is contained in the *IOSTATDM communication area field. This error message is taken from the status array returned by HP TurboIMAGE/iX after an HP TurboIMAGE/iX library procedure is executed. The *IMSTAT communication area field contains the entire status array. For information about these types of errors, refer to the *HP TurboIMAGE/iX Database Management System Reference Manual*.

When an HP TurboIMAGE/iX error occurs, HP ALLBASE/4GL may also display an HP ALLBASE/4GL error message. For these errors, refer to page C-3.

The most common error messages that are specific to the HP TurboIMAGE/iX data manager are listed below:

Data Manager Error Messages C-11

- 19115** Data storage error - beginning or end of chain in data set <data set name>.
- Error type message. This error indicates that the beginning or end of a detail data set chain was reached during a *NEXT or *PREVIOUS file operation.
- 19131** Unable to lock database. The database or part of it is already locked.
- Error type message. You cannot lock a database because another user has already locked the database or some part of it. You must wait until the lock on the database is released before you can lock, and access, the part of the database that you attempted to lock.
- 19500** Database subsystem access disabled for data set: <data set name>.
- Error type message. The named data set cannot be accessed because access to the appropriate database has been disabled from within DBUTIL. Access to the database must be enabled before you can access the data set.
- 19505** *LOCKWAIT has bad value for data set: <data set name>.
- Error type message. The communication area field *LOCKWAIT contains a value lower than -1. This is an invalid value. Change the value in *LOCKWAIT.
- 19520 - 19532** <HP TurboIMAGE/iX intrinsic name> intrinsic failed for data set <data set name>.
- Error type message. These errors indicate that the named HP TurboIMAGE/iX library procedure (intrinsic) has failed. Details about the failure can be obtained from the status array returned from HP TurboIMAGE/iX to the *IMSTAT communication area field. The error number returned from HP TurboIMAGE/iX is contained in the *IOSTATDM communication area field.

The following table indicates the HP ALLBASE/4GL error number related to each library procedure.

C-12 Data Manager Error Messages

Error Number	Library Procedure
19520	DBBEGIN
19521	DBCLOSE
19522	DBDELETE
19523	DBEND
19524	DBFIND
19525	DBGET
19526	DBINFO
19527	DBLOCK
19528	DBMEMO
19529	DBOPEN
19530	DBPUT
19531	DBUNLOCK
19532	DBUPDATE
19540	Record <record name> has no TurboIMAGE attributes , regenerate it. Error type message. You are attempting to use a record created in an earlier release of HP ALLBASE/4GL in connection with an HP TurboIMAGE/iX data set. Regenerate the record to allow HP ALLBASE/4GL to create data manager specific attributes for each record field. After regenerating the record, the error should no longer occur.
19550	Bad TurboIMAGE parameters for data conversion in data set: <data set name>. Error type message. The data manager specific attributes set for a field specification within the named data set are not valid.
19551	Conversion from TurboIMAGE data to Internal data failed for data set: <data set name>.

Data Manager Error Messages C-13

Error type message. This is a run-time error that occurs during conversion of a field in an HP ALLBASE/4GL record buffer to HP TurboIMAGE/iX data. This error indicates that the HP TurboIMAGE/iX data manager specific attributes for a field within the named data set are incompatible with the field specification details.

19552 Conversion from Internal data to TurboIMAGE data failed for <data set name>.

Error type message. This is a run-time error that occurs when HP ALLBASE/4GL attempts to convert HP TurboIMAGE/iX data to an HP ALLBASE/4GL record buffer. This error indicates that the HP TurboIMAGE/iX data manager specific attributes for a field within the named data set are incompatible with the field specification details.

19553 Conversion between internal data types, during TurboIMAGE lock descriptor creation, failed.

Error type message. This is a run-time error that occurs when a lock descriptor in a DM IMAGE *LOCK command cannot be converted to HP TurboIMAGE/iX data. This error indicates that the HP TurboIMAGE/iX data manager specific attributes for a field within the named data set are incompatible with the field specification details.

45073 Logging error - cannot close file, transaction still in progress.

Error type message. This error occurs when HP ALLBASE/4GL attempts to close an HP TurboIMAGE/iX data set while a transaction is still in progress.

C-14 Data Manager Error Messages

D

Operating System Environment

This appendix describes the MPE/iX operating system environment established for HP ALLBASE/4GL. The description is under the following subheadings:

- HP ALLBASE/4GL Files.
- MPE/iX Account Structure.
- MPE/iX Formal File Designators.
- MPE/iX Variables.
- S-file Size Limits.
- Multiple HP ALLBASE/4GL Systems.
- Start-Up Command File.
- Screen Image Printing.
- Hardware and MPE Clock Settings.
- HP ALLBASE/4GL Terminals.

For the majority of installations, the default HP ALLBASE/4GL environment should not require any modification.

In general, you should not make any changes to the default operating system environment for HP ALLBASE/4GL. Incorrect changes may prevent HP ALLBASE/4GL from operating correctly and may even prevent you from running the system at all. Don't change the MPE/iX environment for HP ALLBASE/4GL, unless you are fully experienced in using the MPE/iX operating system and have a good understanding of the environment needed to support HP ALLBASE/4GL.

Note

You may adversely affect future product support and your ability to use future product enhancements if you make any changes to the operating system environment for HP ALLBASE/4GL.

HP ALLBASE/4GL Files

HP ALLBASE/4GL uses a number of different types of files. They are:

- S-files.
- Program files.
- Command files.
- Application databases.
- Terminal support files.

S-Files

The HP ALLBASE/4GL S-files are a set of 11 pairs of files. They contain the definition of any completed applications on the system, and the application developers' input for applications under development. The S-files are named S01D to S11D, and S01I to S11I.

Each S-file consists of a pair of files. One file is a data file (identified by the *D* suffix on the name), and the other file is an index file (identified by the *I* suffix on the name). Each index file is matched to the corresponding data file and must not be copied separately, or deleted. If necessary, you can use the HP4REMK utility program to rebuild the S-file indexes. Refer to Using Remake, for information about the HP4REMK utility.

One set of S-files can support a number of applications and a number of users working simultaneously.

In effect, a set of S-files is an HP ALLBASE/4GL system. One MPE/iX system can support multiple sets of HP ALLBASE/4GL S-files. Each S-file

D-2 Operating System Environment

set requires a system administrator, and can support different system-wide configurations, user name lists, and applications.

Note

The developer environment S-files are not the same as the run-time environment S-files. The two types of S-files are not interchangeable.

All the files in each set of S-files must be in the same MPE/iX group and account.

Copying S-Files

If required, you can use more than one set of S-files on the MPE/iX system. If you do create more than one set of S-files, each set must reside in a separate MPE/iX group.

Caution

Always use the command file HP4SCOPY.HP4GL.SYS to create new copies of the HP ALLBASE/4GL S-files. Attempting to use FCOPY directly will waste disk space.

Each of the HP ALLBASE/4GL S-files has a file limit of 2 000 000 records to allow for expansion. Since the FCOPY command always creates a file with 32 extents, the first extent will occupy up to 250 000 sectors of disk space. To avoid using excessive disk space, the HP4SCOPY command file uses the BUILD command to create suitable files, and then copies the S-files into the new files.

To use the HP4SCOPY command file, enter the following command at the MPE/iX prompt:

$$\text{HP4SCOPY.HP4GL.SYS src=src_grp [dest=dest_grp] \left[\text{overwrite} = \begin{Bmatrix} \text{Y} \\ \text{N} \end{Bmatrix} \right]$$

In this command, the term *src_grp* is the group containing the files you wish to copy, and *dest_grp* is the group you want to copy the files to.

For the HP ALLBASE/4GL developer environment, the MPE/iX sub-system installation procedure installs a set of S-files in the HP4GL.SYS group. In general, you should leave the S-files in the HP4GL.SYS group in their original

state. This ensures that you have a “clean” set of S-files if you need to establish a new HP ALLBASE/4GL environment.

For the HP ALLBASE/4GL run-time environment, the S-files are installed in the HP4GLR.SYS group. The HP4SCOPY command file is also in the HP4GLR.SYS group.

Refer to S-File Size Limits on page D-12 for more information about S-file sizes.

Program Files

The HP ALLBASE/4GL program files are the executable program files that run HP ALLBASE/4GL. Only one set of program files is required on each MPE/iX system, as more than one set of S-files can use the same set of program files simultaneously. The HP ALLBASE/4GL program files reside in the PUB group of the SYS account.

Some of the program files in the developer environment have an equivalent program file in the run-time environment. They differ slightly in functionality and therefore require a different name. To differentiate the developer version of a program file from its run-time equivalent, the run-time program file has an “R” appended to its name. Where this occurs it is indicated by the following notation:

filename[R]

The HP ALLBASE/4GL program files are listed in the table below.

D-4 Operating System Environment

Table D-1. Program Files in the Developer Environment

Name	Purpose	Stand-Alone Use
HP4ARPT	Administrator reports	No
HP4ATOS	ASCII to S-file conversion program	Stand-alone only
HP4CHKAP	Application consistency checker	Yes
HP4DRPT	Developer reports	No
HP4FUTL	File create/delete/reformat utility	No
HP4GEN	Generate program	No
HP4GLB	Main HP ALLBASE/4GL program	Yes
HP4LD	Application loading program	No
HP4REMK	S-file index rebuilding program	Stand-alone only
HP4REOD	S-file reorder/compacting program	Stand-alone only
HP4RP	Report painter	No
HP4SORT	Report sorting program	No
HP4SP	Screen painter	No
HP4STOA	S-file to ASCII conversion program	Stand-alone only
HP4TUPLD	HP TurboIMAGE/iX database definition loading program	Stand-alone only
HP4ULD	Application unloading program	No

Table D-2. Program Files in the Run-Time

Name	Purpose	Stand-Alone Use
HP4ARPTR	Administrator reports	No
HP4CHAPR	Application consistency checker	Yes
HP4FUTLR	File create/delete/reformat utility	No
HP4GLBR	Main HP ALLBASE/4GL program	Yes
HP4LDR	Application loading program	No
HP4REMKR	S-file index rebuilding program	Stand-alone only
HP4REODR	S-file reorder/compacting program	Stand-alone only
HP4SORTR	Report sorting program	No

MPE/iX Command Files

HP ALLBASE/4GL uses a number of MPE/iX command files. These files are used at the following times:

- During the product installation process.
- To start HP ALLBASE/4GL from the MPE/iX prompt.
- To create new copies of the HP ALLBASE/4GL system files.

The command files are located in the PUB.SYS group and the HP4GL[R].SYS group.

The command files are listed in the following table.

D-6 Operating System Environment

Table D-3. Command Files in the PUB.SYS Group

Name	Purpose	Stand-Alone Use
HP4GL	Start-up command file for HP ALLBASE/4GL developer	Yes
HP4GL2	Used by HP4GL command file	No
HP4GLR	Start-up command file for HP ALLBASE/4GL run-time environment	Yes
HP4GL2R	Used by HP4GLR command file	No

Table D-4. Command Files in the HP4GL[R].SYS Group

Name	Purpose	Stand-Alone Use
HP4BLD[R]	Builds HP ALLBASE/4GL environment during installation	Yes
HP4BLD2	Used by HP4BLD[R]	No
HP4SCOPY	Copies HP ALLBASE/4GL S-files (Used by HP4BLD[R])	Yes

Application Databases

HP ALLBASE/4GL provides facilities for applications to access HP ALLBASE/SQL and HP TurboIMAGE/iX databases. The databases used in an application can be located anywhere in the MPE/iX file system.

HP ALLBASE/4GL also provides the facilities needed to create and maintain KSAM data files and serial data files from within HP ALLBASE/4GL.

KSAM data files do not need to be unique to an HP ALLBASE/4GL application or even a set of HP ALLBASE/4GL S-files. However, we recommend that only applications in one set of S-files use any given set of data files.

Non-HP ALLBASE/4GL applications can use HP ALLBASE/4GL KSAM data files, however the HP ALLBASE/4GL transaction control system may not operate correctly if HP ALLBASE/4GL and non-HP ALLBASE/4GL applications use these data files simultaneously.

Terminal Support Files

Special files are required to provide the correct support for HP ALLBASE/4GL terminals. These files must be in the HP4TERM.SYS group and account.

Refer to HP ALLBASE/4GL Terminals on page D-19 for more information about these files.

MPE/iX Account Structure

The HP ALLBASE/4GL installation procedure requires you to create an MPE/iX account enabling it to create the default HP ALLBASE/4GL account structure. This section describes the structure created by the installation procedure.

The account for the HP ALLBASE/4GL system contains the following groups:

Developer Environment	Run-Time Environment	Purpose
HP4APPN	HP4APPNR	Unloaded application definitions
HP4DATA	HP4DATAR	KSAM and serial data files
HP4DBM	HP4DBMR	HP ALLBASE/SQL database modules
HP4FS	HP4FSR	KSAM file create/delete/reformat files
HP4S	HP4SR	HP ALLBASE/4GL S-files
HP4SQL	HP4SQLR	HP ALLBASE/SQL databases
HP4TI	HP4TIR	HP TurboIMAGE/iX databases

The developer and run-time environments use essentially the same groups. The only difference is the last character of the run-time environment group names. In the run-time environment, the group names all end with “R”. This is indicated by the following notation:

GROUP[R]

D-8 Operating System Environment

The following paragraphs explain the purpose of these groups.

HP4APPN[R] Group

This group is used for the files containing unloaded HP ALLBASE/4GL applications. The HP ALLBASE/4GL administrator application unloading facility creates files in this group to store the definitions of unloaded applications.

The administrator application loading utility reads the definition of the application to be loaded from a file in this group.

HP ALLBASE/4GL uses the MPE/iX variable HP4APPNPATH to identify the name and location of the group for the unloaded application definition file. The HP ALLBASE/4GL administrator must be able to create files in this group.

HP4DATA[R] Group

This group is available for KSAM and serial data files. You can use a different name for the group if desired, or locate the data files elsewhere in the MPE/iX file system.

By default, HP ALLBASE/4GL uses the MPE/iX variable HP4DATAPATH to identify the name and location of this group. HP4DATAPATH can be reset if another group is to be used to contain the HP TurboIMAGE/iX databases.

HP4DBM[R] Group

This group is used for the database module files for applications and versions.

HP ALLBASE/4GL uses the MPE/iX variable HP4DBMPATH to identify the name and location of the group containing the database module files. Application developers and the HP ALLBASE/4GL administrator must be able to create files in this group.

HP4FS[R] Group

This group is required if your applications use KSAM data files. It contains the file-structure files that are created when application developers create or reformat KSAM data files.

It is also used for file-structure files that may be required when the HP ALLBASE/4GL administrator loads or unloads a run-only application. These files are part of the automatic data file reformatting system for run-only applications.

HP ALLBASE/4GL uses the MPE/iX variable HP4FSPATH to identify the name and location of this group. Application developers and the HP ALLBASE/4GL administrator must be able to create files in this group.

HP4S[R] Group

This group is used for the HP ALLBASE/4GL S-files. You can use a different name for this group if desired. HP ALLBASE/4GL uses the MPE/iX variable HP4SPATH to identify the name and location of this group.

HP ALLBASE/4GL can access S-files in any part of the MPE/iX file system. All HP ALLBASE/4GL users must have read, write and lock access permissions on the S-files.

HP4SQL[R] Group

This group is optional and is available for HP ALLBASE/SQL application databases. You can use a different name for the group if desired, or locate the HP ALLBASE/SQL databases elsewhere in the MPE/iX file system.

By default, HP ALLBASE/4GL uses the MPE/iX variable HP4SQLPATH to identify the name and location of this group. HP4SQPPATH can be reset if another group is to be used to contain the HP ALLBASE/SQL database.

D-10 Operating System Environment

HP4TI[R] Group

This group is optional and is available for HP TurboIMAGE/iX application databases. You can use a different name for the group if desired, or locate the HP TurboIMAGE/iX databases elsewhere in the MPE/iX file system.

By default, HP ALLBASE/4GL uses the MPE/iX variable HP4TIPATH to identify the name and location of this group. HP4TIPATH can be reset if another group is to be used to contain the HP TurboIMAGE/iX databases.

Formal File Designators

HP ALLBASE/4GL can receive inputs from a number of different sources, and direct output to a number of different destinations. The following table lists the formal file designators for the sources of inputs and the output destinations, the default devices, and the corresponding formal file designators. Each of these files can be equated using an MPE/iX file equation.

Formal File Designator	Default Device	Comments
HP4KYIN	\$STDIN	If HP4KYIN is equated, keystrokes are read from the specified file. The file must use single byte ASCII records.
HP4SCOUT	\$STDLIST	If HP4SCOUT is equated, screen output is written to the specified file. The default is a variable length record ASCII file with a maximum length of 1024 bytes.
HP4KYOUT	none	If HP4KYOUT is equated, keystrokes are logged to the specified file. Must be single byte ASCII records.
HP4RFLOG	DEV=DISC;TEMP	The log file produced by the KSAM file reformat facility. The default is a fixed length ASCII file with 256 byte records, and a file limit of 4000 records.
HP4GNERR	DEV=DISC;TEMP	Log file for all generate errors produced in a development session. The default is a fixed length ASCII file with 80 byte records, and a file limit of 1 000 000 records.
HP4TRACE	DEV=DISC;TEMP	Log file of all messages displayed while trace mode is active during a development session. The default is a fixed length record ASCII file with 160 byte records, and a file limit of 1 000 000 records.
HP4REP	DEV=LP	The destination for reports from the developer and administrator applications, and screens printed via the system printer. The default record size depends on the report being printed.

D-12 Operating System Environment

MPE/iX Variables

HP ALLBASE/4GL uses the following MPE/iX variables to identify the location of various items when a user signs on to HP ALLBASE/4GL.

HP ALLBASE/4GL also uses MPE/iX job control words (JCWs) to select the message catalog to suit the user's language, and set the collating sequence for KSAM files.

The HP ALLBASE/4GL start-up command file automatically sets appropriate values for the following variables and job control words.

- **HP4APPPATH** This variable identifies the group and account containing any unloaded applications.
- **HP4DATAPATH** This variable identifies the group and account containing the KSAM and serial data files for an application.
- **HP4DBMPATH** This variable identifies the group and account containing any database module files created for an unloaded application.
- **HP4FSPATH** This variable identifies the group and account containing the file-structure files. These files are required when application developers create or reformat KSAM data files, or the administrator loads or unloads a run-only application requiring reformatting of KSAM data file reformatting.
- **HP4SPATH** This variable identifies the group and account containing the HP ALLBASE/4GL S-files.
- **HP4SQLPATH** This variable identifies the group and account containing the DBECon file for the HP SQL database environment used by an application that accesses HP ALLBASE/SQL.
- **HP4TERM** This variable identifies the user's terminal type. The value of this variable determines the name of the termdata file used by HP ALLBASE/4GL to drive the terminal. Refer to HP ALLBASE/4GL Terminals on page D-19 for more information about terminals.
- **HP4TIPATH** This variable identifies the group and account containing the HP TurboIMAGE/iX databases used by an application.
- **NLDATALANG** The KSAM data manager uses this variable to determine the collating sequence during the data file creation process. By default, this job control word is set to 001 (American).

- **NUSERLANG** HP ALLBASE/4GL uses this job control word to select the appropriate message catalog. By default, this job control word is set to 001 (American). Refer to Message Catalogs on page D-14 for information about the messages system.

HP ALLBASE/4GL uses the following MPE/iX variables for various MPE/iX and HP ALLBASE/4GL activities. HP ALLBASE/4GL assigns default values to some of these variables - these values are noted below. Values can be assigned to these variables via a UDC or command file, or directly from the MPE/iX command interpreter.

- **HP4BG** This variable contains a job logon stream (including passwords) to allow HP ALLBASE/4GL to create and stream jobs to run background processes.
- **HP4DBMSIZE** If set, this variable indicates the file limit for the database module files created by HP ALLBASE/4GL. The database module file is a binary file, with 500 byte records. The default value is 4095.
- **HP4INV_PRT_CHAR** This variable specifies the character used when screen printing to indicate a space character in an inverse video field. The default value is the DEL character, which prints a checkerboard pattern on some printers. In some circumstances the DEL character may be interpreted incorrectly and may cause unreliable printing. If this occurs, change the value of this variable to a printable ASCII character, such as a hash (#).
- **HP4SCREEN_PRINT** This defines the destination printer that will print HP ALLBASE/4GL screen images. Refer to Screen Image Printing on page D-18 for more details.

D-14 Operating System Environment

S-File Size Limits

The source and generated code for all HP ALLBASE/4GL applications is stored in the S-files. As you create further applications or increase the size of an application, there is a greater demand for storage space in the S-files.

The MPE/iX operating system uses files of a predetermined maximum size. The HP ALLBASE/4GL installation procedure establishes an arbitrary limit of 2 000 000 records for each S-file. Under normal circumstances, this limit is far in excess of the required file size.

Note



Under the MPE/iX operating system, the amount of disk space used by a file depends on the number of records in the file, and not the file limit. Reducing the file limit for the HP ALLBASE/4GL S-files will not have any effect on disk space usage.

If the file limits for any S-files have been reduced, it is possible to reach a situation where the S-files cannot accommodate new records. At the start of each session, and whenever you attempt to write a record to the S-files, HP ALLBASE/4GL checks the amount of space available in the S-files. HP ALLBASE/4GL issues a warning message, and then aborts if you attempt to write a record to the S-files, and there is insufficient space available in the files.

If the S-file size limits have been reduced, the available space can become exhausted for the following reasons:

- The files contain a large number of logically deleted records.
- The number of active records in the file is close to the file limit.

When HP ALLBASE/4GL deletes a record from the S-files, the record is marked as logically deleted, but it is not deleted physically. Records are logically deleted from the S-files when an application component is deleted using the *Deletions* screen in the developer utilities menu, or an entire application is deleted using the administrator deletions utility. The HP4REMK and HP4REOD utility programs allow you to remake the S-file indexes, and rebuild the S-file data files. This process purges logically deleted records.

If removing logically deleted records does not provide sufficient space in the S-files, you must increase the file limits.

Using Remake and Reorder

The HP4REMK[R] and HP4REOD[R] utility program have the following purpose:

- **HP4REMK[R]** This program is known as “remake”. It remakes the index portion of an S-file. It reorganizes the “I” suffix file by balancing the index tree, reclaiming any dead space, and checking for invalid indexes.
- **HP4REOD[R]** This program is known as “reorder”. It reorders the data portion of an S-file. It reorganizes the “D” suffix file by recreating it in index sequence and purging any logically deleted records.

Caution



Always run HP4REMK on an S-file set before running HP4REOD. Make sure that all users have signed off from HP ALLBASE/4GL before you run these programs.

Before you run these programs you must set the MPE/iX variable HP4SPATH to indicate the group and account in which the S-files reside. For example, if the S-files are in the HP4S.HP4GL group and account, you must set the HP4SPATH variable as follows:

```
SETVAR HP4SPATH "HP4S.HP4GL"
```

To run the programs, exit from HP ALLBASE/4GL, and enter the following commands at the MPE/iX prompt:

```
HP4REMK ' [n[-m]] '
```

```
HP4REOD ' [n[-m]] '
```

where *n* and *m* are numbers between 1 and 11 indicating the S-file, or range of S-files that you wish to remake or reorder. If you do not specify a value on the command line, the program will ask you which file to remake or reorder.

Increasing the S-File Limits

If running HP4REMK and HP4REOD does not reclaim sufficient free space in the S-files, the files concerned must be enlarged.

D-16 Operating System Environment

You can enlarge the files with the HP4REMK and HP4REOD utilities, or you can use MPE/iX commands directly.

To enlarge the S-files using HP4REMK and HP4REOD, the commands are:

```
HP4REMK '-1 dddd [n[-m]]'
```

```
HP4REOD '-1 dddd [n[-m]]'
```

In these commands, *dddd* is the file limit for the enlarged files, and *n* and *m* are values between 1 and 11 to indicate the file to be enlarged. Note that you must set the MPE/iX variable HP4SPATH before you execute these commands.

Alternatively, you can use the following MPE/iX commands to enlarge the files:

```
FILE *TEMPFILE;DISC=nnnnn  
FCOPY FROM=Snn?;TO=*TEMPFILE;NEW  
PURGE Snn?  
RENAME TEMPFILE,Snn?
```

In these commands, *nnnnn* is the file limit for the enlarged file, and *Snn?* is the S-file that you want to enlarge.

Message Catalogs

The messages displayed by HP ALLBASE/4GL are stored in a message catalog in the PUB.SYS group and account. The catalog is stored in a keyed file called *HP4nnn* in the developer environment, and *HP4Rnnn* in the run-time environment. The key file is called *HP4nnnK* or *HP4RnnnK* in the developer and run-time environments respectively. In these file names, the term *nnn* is a three digit number identifying the user's current language.

HP ALLBASE/4GL uses the job control word (JCW) NLUSERLANG to determine which message catalog to use. If this job control word is not defined, HP ALLBASE/4GL uses the message catalog HP4[R]001 (American) by default.

Multiple HP ALLBASE/4GL Systems

One MPE/iX system can support a number of HP ALLBASE/4GL systems simultaneously, and each HP ALLBASE/4GL system in turn can support a number of users and applications. Typically, you might want to use multiple HP ALLBASE/4GL environments if you are using the same MPE/iX system for developing multiple applications.

If you need to use multiple HP ALLBASE/4GL environments on your MPE/iX system, you don't need to create multiple copies of the entire HP ALLBASE/4GL system. The only parts you need to duplicate are the HP ALLBASE/4GL S-files. More than one HP ALLBASE/4GL developer environment can share the same program files, message catalogs and terminal support files.

Note that the program files and message catalogs are not the same for the developer and run-time environments. A developer environment and a run-time environment cannot share program files or message catalogs. The terminal support files are identical for both the developer environment and the run-time environment.

If you do use more than one set of HP ALLBASE/4GL S-files, they must be in separate groups. The S-file groups can be in the same account, or in different accounts.

When you use multiple S-file sets, the MPE/iX variable HP4SPATH must be set correctly for each user. You will also need to set the HP4DATAPATH variable, the HP4TIPATH variable, and the HP4SQLPATH variable for each user, according to the S-file set being used.

The following table is an example showing the values of the MPE/iX environment variables required to access the two HP ALLBASE/4GL environments. This table assumes you have copied the HP ALLBASE/4GL S-files to the HP4S groups in two accounts called HP4DEVEL and HP4TEST.

D-18 Operating System Environment

User	Variable	Typical Value
Developer	HP4SPATH	HP4S.HP4DEVEL
	HP4DATAPATH	DEVDATA.HP4DEVEL
	HP4SQLPATH	DEVSQL.HP4DEVEL
	HP4TIPATH	DEVTL.HP4DEVEL
Test user	HP4SPATH	HP4S.HP4TEST
	HP4DATAPATH	TESTDATA.HP4TEST
	HP4SQLPATH	TESTSQL.HP4TEST
	HP4TIPATH	TESTTL.HP4TEST

Run-Only Applications

If required, completed applications can be unloaded from the development S-files and loaded into the test environment S- files as run-only applications. This reduces disk usage since run-only applications do not require the developer source information to be present. Unloading applications in run-only form also ensures that they cannot be modified.

Start-up Command File

HP ALLBASE/4GL is supplied with a command file HP4GL (or HP4GLR for the run-time environment) to start HP ALLBASE/4GL from the MPE/iX command interpreter.

This command file allows you to specify a number of options on the command line. These options include the commands for log-on bypass, or customized log on. You can also specify values for the MPE/iX variables used by HP ALLBASE/4GL by entering the values on the command line. Refer to chapter 2 for details of the log-on bypass facility and the customized log-on screen facility.

The command line to start HP ALLBASE/4GL is:

```
:HP4GL ["options"] [variable=value] . . .
```

The *variable* term on this command line allows you to set values for the following MPE/iX variables or job control words used by HP ALLBASE/4GL.

- HP4APPNPATh
- HP4DATAPATH
- HP4DBMPATH
- HP4FSPATH
- HP4SPATH
- HP4SQLPATH
- HP4TERM
- HP4TIPATH
- NLUSERLANG (job control word)
- NLDATA LANG (job control word)

If any of these variables or job control words are set when you start HP ALLBASE/4GL, the command file uses the current value for the variable or JCW.

Values specified on the command line only apply for the current HP ALLBASE/4GL session. The existing values (if any) for these variables and job control words are restored when you exit from HP ALLBASE/4GL.

If required, you can use a logon UDC to set values for these variables and job control words.

If the variables and job control words are undefined when you start HP ALLBASE/4GL, the command file assumes the following default values.

D-20 Operating System Environment

Variable	Default Value (Developer)	Default Value (Run-Time)
HP4APPNPATH	HP4APPN	HP4APPNR
HP4DATAPATH	HP4DATA	HP4DATAR
HP4DBMPATH	HP4DBM	HP4DBMR
HP4FSPATH	HP4FS	HP4FSR
HP4SPATH	HP4S	HP4SR
HP4SQLPATH	HP4SQL	HP4SQLR
HP4TERM	HP	HP
HP4TIPATH	HP4TI	HP4TIR
NLUSERLANG	1 (American)	1 (American)
NLDATALANG	1 (American)	1 (American)

These default values correspond to the group structure that is created for the standard HP ALLBASE/4GL installation.

Screen Image Printing

HP ALLBASE/4GL allows you to print an image of a screen by pressing **CTRL** and **P** followed by **Return** while the screen is displayed. The screen printing system uses the HP-UX variable HP4SCREEN_PRINT to determine the destination printer.

Note



You may also need to alter the HP4INV_PRT_CHAR MPE/iX variable. Refer to the description of this variable under MPE/iX Variables for details.

HP4SCREEN_PRINT Variable

If the HP4SCREEN_PRINT variable is not defined, HP ALLBASE/4GL prints screen images to device class LP. (The formal designator is HP4REP.)

If HP4SCREEN_PRINT exists as an empty string (that is, ""), HP ALLBASE/4GL attempts to print screen images to a slave printer attached to the user's terminal. The terminal configuration must be correct for slave printing to operate. Refer to Terminal Configurations for more details.

Note

If HP4SCREEN_PRINT is a non-null value, the screen image will not be printed.

The **To System** function key in the screen printing function key set also uses the HP4SCREEN_PRINT environment variable.

The **To Local** function key always attempts to print the screen image to a terminal slave printer.

Hardware and MPE Clock Settings

HP ALLBASE/4GL uses the standard C programming language library routines to determine the time. These routines rely on the system hardware clock (ISL time), rather than the local MPE/iX time setting.

HP ALLBASE/4GL assumes the system hardware clock is set to Greenwich Mean Time (GMT) and uses the MPE/iX variable TZ to determine the offset for the local time zone. If the TZ variable is not defined, HP ALLBASE/4GL computes its correct value by obtaining the difference between the values of the hardware clock and the local MPE/iX time setting.

Do not set TZ manually unless you are fully familiar with the way it is specified.

D-22 Operating System Environment

HP ALLBASE/4GL Terminals

HP ALLBASE/4GL runs on most HP TERM0 standard terminals, and supports features such as the HP line drawing character set, touchscreen operation, and color. HP ALLBASE/4GL requires at least the terminal functionality provided by the HP 2622 terminal.

HP ALLBASE/4GL uses a file known as a terminal support file (a *termdata* file) containing the information necessary to drive the terminal. HP ALLBASE/4GL uses the value of the HP4TERM variable as the name of the terminal support file to retrieve.

If the HP4TERM variable is not set, HP ALLBASE/4GL assumes minimal terminal capability equivalent to an HP 2622 terminal.

Terminal Support Files

The following terminal support files are supplied with HP ALLBASE/4GL:

Table D-5. Terminal Support Files

hp	hp2392t	hp2397at	hp2625
hpcl	hp2392a	hp2622	hp2626
hpclt	hp2392at	hp2622a	hp2626a
hpl	hp2393	hp2622p	hp2626p
hplt	hp2393a	hp2623	hp2627
hpt	hp2394	hp2623a	hp2627a
hp150	hp2394a	hp2623p	hp2627p
hp2382	hp2397	hp2624	hp2628
hp2382a	hp2397t	hp2624a	hp70092
hp2392	hp2397a	hp2624p	hp70094

These files are located in the HP4TERM.SYS group and account.

HP ALLBASE/4GL uses the value of the HP4TERM variable as the name of the appropriate terminal support file.

If you want to use a terminal that does not appear in this list, set the HP4TERM variable to one of the values shown in the following table.

File Name	Terminal Type
hp	Basic HP terminal
hpcl	HP terminal with color and line drawing character set
hpclt	HP terminal with color, line drawing character set, and touchscreen operation
hpl	HP terminal with line drawing character set
hplt	HP terminal with line drawing character set and touchscreen operation
hpt	HP terminal with touchscreen operation

Touchscreen Terminals

If your terminal supports touchscreen operation, set the HP4TERM variable to one of the options with a “t” in their suffix. Touchscreen operation is available for all HP 150 terminals.

Terminal Initialization and Resetting

When HP ALLBASE/4GL starts up or returns from a temporary exit, it sets the terminal into a mode compatible with the MPE/iX format mode. HP ALLBASE/4GL resets the terminal to its initial state whenever it temporarily exits via an EXTERNAL *REFRESH call, when invoking a temporary MPE/iX command interpreter, or when calling any other products such as HP ALLBASE/QUERY or HP ISQL.

HP ALLBASE/4GL makes extensive use of terminal tabstops to maximize the efficiency of terminal output. It initializes the terminal tabstop settings by copying the file TABSET.HP4TERM.SYS to the terminal. This file contains the escape sequences necessary to clear any current settings and make the appropriate new settings. Do not delete or rename the TABSET file.

D-24 Operating System Environment

Terminal Configurations

HP ALLBASE/4GL supports the HP Roman8 extended character set. To use this character set, your terminal must be in 8-bit ASCII mode.

For HP 150 and HP2394 terminals, an additional parameter, Tab = Spaces, must be set to NO.

If you wish to print HP ALLBASE/4GL screens, you may also need to alter the HP4INV_PRT_CHAR variable to a printable ASCII character. In some circumstances, the default value of DEL causes unreliable printing.

To ensure reliable operation of a slave printer with an HP terminal, you must use the correct terminal configurations.

The following tables show the suggested configuration for an HP 700/92 or an HP 2392 terminal.

Datacomm Configuration

Baudrate	9600	Parity/DataBits	None/8
EnqAck	NO	Asterisk	OFF
Chk Parity	NO	SR(CH)	LO
RecvPace	Xon/Xoff	CS(CB)Xmit	NO
XmitPace	Xon/Xoff		

External Device Configuration

BaudRate	9600	Parity/DataBits	None/8
PrinterNulls	000	PrinterType	ROMAN8
SRRXmit	NO	XmitPace	Xon/Xoff
SRRInvert	NO	CS(CB)Xmit	NO

Terminal Configuration

Datacomm/ExtDev	PORT1/PORT2	Keyboard	USASCII
Terminal Id	2392A	Language	ENGLISH
LocalEcho	OFF	CapsLock	OFF
Start Col	01	Bell	ON
XmitFunctn(A)	YES	SPOW(B)	NO
InhEolWrp(C)	NO	Line/Page(D)	LINE
InhHndShk(G)	NO	Inh DC2(H)	NO
Esc Xfer(N)	YES		
FldSeparator	U/S	BlkTerminator	R/S
ReturnDef	C/R	TermMode	HP
Tab = Spaces	NO		

D-26 Operating System Environment

HP ALLBASE/4GL Limits

Item Limits

HP ALLBASE/4GL places some limits on the number of items in an application. The following table summarizes these limits.

Item Type	Limit per Application
Alpha-numeric constants	255
Application titles	4000
Calculated items	255
Decision tables	4000
Databases	255 *
Field specifications	4000
File definitions	255
Function key sets	4000
Functions	4000
Help screens	4000
Master titles (per set of S-files)	4000
Messages	4000
Numeric constants	255
Processes	4000
Record layouts	4000
Reports	4000
Scratch-pad fields	99
Screens	4000
SQL logic blocks	4000
Validation ranges	255
Validation tables	255 **
Variables	255
Work areas	999

- (* Each application may only access one HP ALLBASE/SQL database)
- (** Depends on table size. Total table text cannot exceed 32 kilobytes)

Other Limits

HP ALLBASE/4GL also imposes some limits on the number of components in certain item types. These limits are summarized below.

Fields per record layout	255
Record layouts per file	12
Key fields per record layout	16
Characters per record field	999
Characters per key field	120
Characters per record layout	2048
Fields per screen	99
Characters per screen field	80
Lines per report definition *	255
Characters per report field *	255
Fields per report line *	127
Characters per report line	255
Characters per logic block	400
Databases per set of S-files	4000

(* Depends on system memory configuration.)

In some cases, logically deleted items still occupy space in the HP ALLBASE/4GL internal name lists. If deleted items occupy a significant amount of name list space, some of the limits listed above may be reduced.

To recover this space, generate the entire application by selecting the *All* option on the generates menu. HP ALLBASE/4GL compacts the internal name list when it generates an entire application.

E-2 HP ALLBASE/4GL Limits

Application Portability

Introduction

This appendix describes some issues that application developers should consider when writing HP ALLBASE/4GL applications for use on more than one operating system.

Portability Mechanism

In general, applications are portable across the operating systems that support HP ALLBASE/4GL.

To transport an application from one operating system to another, you must unload the application from the source system and copy the required files onto the target file system. You can then use the HP ALLBASE/4GL application loading utility to load the application into the target system.

The files involved in transporting applications are:

- The unloaded application file.
- The database module files.
- The data file-structure file.

The names for these files are shown in the following table.

File Type	Operating System	File Name
Application Definition File	HP-UX MPE/iX	<i>\$HP4SPATH/name.uld</i> NAME.!HP4APPNPATH
Database Module File	HP-UX MPE/iX	<i>\$HP4SPATH/name.dbm</i> NAME.!HP4DBMPATH
File Structure File	HP-UX MPE/iX	<i>\$HP4SPATH/name.ufs</i> NAME.!HP4FSPATH

Application Definition File

The administrator application unloading utility unloads the definition of an application (and up to 10 versions of the application) into a single application definition file. The application definition file can contain the developer source code, or can contain a run-only application. The application source code is not required to transport run-only applications between operating systems.

An application definition file created on one operating system can be loaded into an HP ALLBASE/4GL system running under a different operating system provided you observe the precautions that are outlined in the following paragraphs.

Under the MPE/iX operating system, the application definition file must be a variable length ASCII file with a maximum record length of at least 1 kilobyte.

When shipping an application definition file from HP-UX to MPE/iX, the file can be shipped as a fixed length record binary file.

Under the MPE/iX operating system, HP ALLBASE/4GL uses the current value of the variable HP4APPNPATH to determine the name of the group and account used for application definition files. Under the HP-UX operating system, HP ALLBASE/4GL creates or reads application definition files in the directory containing the HP ALLBASE/4GL S-files.

Refer to chapter 12 for more information about application definition files.

F-2 Application Portability

Database Module Files

A database module file contains the definition of the stored module for an application or a version. HP ALLBASE/4GL creates a database module file when you generate either an entire application, or the SQL logic blocks in an application or version that uses HP ALLBASE/SQL. HP ALLBASE/4GL creates one database module file for an application, and one for each version of the application. A database module file is not required for an application that does not access an HP's database.

The HP ALLBASE/4GL application loading utility automatically installs the database module in the application database environment during the application loading procedure.

The database module files are portable between the HP-UX operating system and the MPE/iX operating system. Under the MPE/iX operating system, the database module file must be a fixed length binary file with a record length of 500 bytes.

Under the MPE/iX operating system, HP ALLBASE/4GL uses the current value of the variable HP4DBMPATH to determine the name of the group and account used for database module files. Under the HP-UX operating system, HP ALLBASE/4GL creates or reads database module files in the directory containing the HP ALLBASE/4GL S-files.

Refer to the *HP ALLBASE/4GL Developer Administration Manual* for more information about database module files.

File Structure File

The file structure file is only required if you are porting run-only applications that access KSAM or KSAM data files, and the files need to be reformatted to reflect changes to the record layouts. This file is not required for applications that use HP ALLBASE/SQL databases or HP TurboIMAGE/XL databases.

HP ALLBASE/4GL builds a file structure file during the unloading process for a run-only application. During the application loading process, HP ALLBASE/4GL uses the file structure file to reformat the application data files if required.

Under the MPE/iX operating system, the file structure file must be a fixed length binary file with a maximum record length of at least 8 kilobytes.

Under the MPE/iX operating system, HP ALLBASE/4GL uses the current value of the variable HP4FSPATH to determine the name of the group and account used for file structure files. Under the HP-UX operating system, HP ALLBASE/4GL creates or reads file structure files in the directory containing the HP ALLBASE/4GL S-files.

Refer to the *HP ALLBASE/4GL Developer Administration Manual* for more information about run-time application data file reformatting.

Application Design Considerations

The following paragraphs outline some application design considerations that will help you develop applications that are readily portable between operating systems. These considerations are described under the following headings:

- Application and Version names.
- Screen field behaviour.
- Data manager differences.
- Reporting environment.
- Using external programs.
- Administrator functionality.

Application and Version Names

Under the HP-UX operating system, application and version names can contain alphabetic characters, numbers, and the underscore character (_). The name must start with an alphabetic character.

Under the MPE/iX operating system, application and version names must be valid MPE/iX file names. (HP ALLBASE/4GL uses the application name as the name of the database module file, and the application definition file.) MPE/iX application and version names should only use the characters *A* to *Z* and *0* to *9*.

HP ALLBASE/4GL is case sensitive with respect to application and version names, although MPE/iX files names are not case sensitive. This means that

F-4 Application Portability

you must be careful about the application and version names you choose under MPE/iX, or clashes may occur. For example, HP ALLBASE/4GL would regard the two application names *ACCOUNTS* and *accounts* as being distinct. Under MPE/iX, the database module files and application definition files for both these application would have the name *ACCOUNTS*.

Screen Field Behaviour

Under the HP-UX operating system, HP ALLBASE/4GL terminal input and output is character based. Under the MPE/iX operating system, terminal input and output is field or line based. As a result, the user interface and screen field behaviour is somewhat different between the operating systems.

Under MPE/iX, the user must press **Return** or **Tab** to terminate a field entry. No other characters can signify to the host system that data entry for a field is complete. In contrast, under a character based input and output system (such as HP-UX), any action that moves the cursor off a field terminates data entry for that field.

Auto-Flow Fields

Under character based input and output systems, HP ALLBASE/4GL allows you to define a screen field with *automatic flow to next* behaviour. When the user enters sufficient characters to fill the field, HP ALLBASE/4GL automatically commits the field, performs the normal field commit processing such as executing the field function, and then continues to process the next field on the screen.

This feature is not available under the MPE/iX operating system. If you do set the *Automatic Flow to Next* field to Y, the setting is ignored. The user must always press **Return** or **Tab** to terminate data entry for a field.

Justification and Case Shifting

Under the HP-UX operating system, HP ALLBASE/4GL performs field justification and case shifting (for forced uppercase U edit code fields) automatically during data entry. Under the MPE/iX system, justification and case shifting cannot occur until the user commits the field. The data is displayed correctly after the field is committed.

Error Field Highlighting

If a user enters invalid data in an input field, HP ALLBASE/4GL automatically highlights the field. By default, the highlight is flashing inverse video.

Under the HP-UX operating system, HP ALLBASE/4GL automatically removes the error highlight when the user starts reentering data in the field. Under the MPE/iX operating system, the error highlighting remains until the user commits the field again by pressing **(Return)** or **(Tab)**.

Data Manager Differences

The functionality of the HP ALLBASE/SQL database system is the same on both the HP-UX and MPE/iX operating systems.

HP TurboIMAGE/XL databases can only be used on MPE/iX operating systems.

HP ALLBASE/4GL also lets you use KSAM (Indexed Sequential Access Method) or KSAM (Keyed Sequential Access Method) data files. These data management systems provide similar functionality, although some important differences do exist. These differences are described below under the following headings:

- External file names.
- Limits.
- Locking mechanism.
- Data file recovery.

These differences only apply to the KSAM and KSAM data managers.

External File Names

The ISAM data manager under MPE/iX does not allow the external name of a file to be longer than seven characters, since the eighth character is used as a suffix to identify the file type. The following table shows the external names of the files for KSAM and KSAM data files.

F-6 Application Portability

	ISAM	KSAM
Data File	<i>name.dat</i>	name
Index or Key File	<i>name.idx</i>	nameK
Structure File	<i>name.str</i>	nameS

Under the MPE/iX operating system, external file names can use the characters A to Z, and 0 to 9. Note that MPE/iX is not case sensitive with respect to file names. By choosing suitable external names for data files you can ensure that application definitions will be portable across operating systems.

Limits

The ISAM data manager allows up to 16 key fields on a file record. The KSAM data manager under the HP-UX operating system allows up to 19 key fields on a file record.

Locking Mechanism

The MPE/iX KSAM data manager only allows locking at the file level. This means that only one user can access a given file at any time to add, modify or delete a record. (Under some circumstances, multiple users can access the same file in a read-only mode.)

Under the HP-UX operating system, the KSAM data manager provides record level locking. This allows multiple users to access the same file at the same time, while still maintaining full data integrity.

The KSAM data manager under HP-UX provides a MODE *UNLOCK for accessing files and a FILE *UNLOCK command. When a file is opened under this mode, all record locks obtained by the current process are retained until the process terminates, the file is explicitly closed, or the locks are released using the FILE *UNLOCK command. These commands allow multi-user access to a file, yet ensure transaction integrity for a transaction that involves more than one record in a given file. The KSAM data manager does not provide this feature.

Under the MPE/iX system, the MODE *WRITE, and MODE *UNLOCK commands perform the same action as the MODE *LOCK command.

Refer to chapter 12 for more information about data file locking.

Data File Recovery

Under the HP-UX operating system, HP ALLBASE/4GL provides a roll-forward data file recovery system for KSAM data files. This system is not available under other operating systems.

Refer to the *HP-UX Developer Administration Manual* for more information about the data file recovery system.

HP ALLBASE/4GL also provides a transaction control system that allows a developer to roll-back a partially complete transaction. (This functionality is provided by the TRANSACT logic command.) The operation of this system is similar under both the MPE/iX operating system and the HP-UX operating system.

Reporting Environment

Some differences exist in the way that HP ALLBASE/4GL directs reports to printers on different operating systems.

Under the HP-UX operating system, HP ALLBASE/4GL directs reports to one of four system printers, the user's terminal, or a slave printer attached to an HP terminal. The system administrator must specify the device names for the system printers on the system-wide definition screen in the administrator application.

Under the MPE/iX operating system, HP ALLBASE/4GL writes reports to a formal file designator rather than a physical device. The user can then use a file equation to direct the report to the required device or device class. By default, reports are directed to device class LP. (Under MPE/iX HP ALLBASE/4GL can also display reports on the user's terminal or print them on a slave printer.)

Reports developed under one operating system can be transported to other operating systems without any changes. There is no need to generate the report after it has been loaded into the target system. When you transport a report developed under HP-UX to an MPE/iX system, HP ALLBASE/4GL

F-8 Application Portability

uses the HP-UX output file name as the name of the formal designator for the report. Regardless of the HP-UX printer number used for the report, HP ALLBASE/4GL directs the report to a file designator.

If you transport a report developed on an MPE/iX system to an HP-UX system, HP ALLBASE/4GL uses the formal file designator name for the report as the name of the HP-UX output file for the report. HP ALLBASE/4GL assumes a value of 1 for the HP-UX printer number.

Reports that are directed to the user's terminal or a terminal slave printer can be transported between operating systems without any changes.

Using External Programs

HP ALLBASE/4GL allows an application to call an external program. The external program can receive parameters from HP ALLBASE/4GL, and return values to HP ALLBASE/4GL. The external program must be written in a language that is supported on the host operating system, and must be compiled and linked to run on that target system.

The mechanism for returning values from an external program to HP ALLBASE/4GL differs between MPE/iX and other operating systems.

Under the HP-UX system, the external program returns values to HP ALLBASE/4GL by writing them on file descriptor number 3. Under MPE/iX, the external program can return values by writing them to the message file HP4EXTP using the FWRITE intrinsic.

Refer to the description of the EXTERNAL command in chapter 12 for more information about using external programs.

Administrator Functionality

Under the HP-UX operating system, the HP ALLBASE/4GL administrator application provides facilities for file backup, restoration and recovery. It also provides facilities for printer and report control.

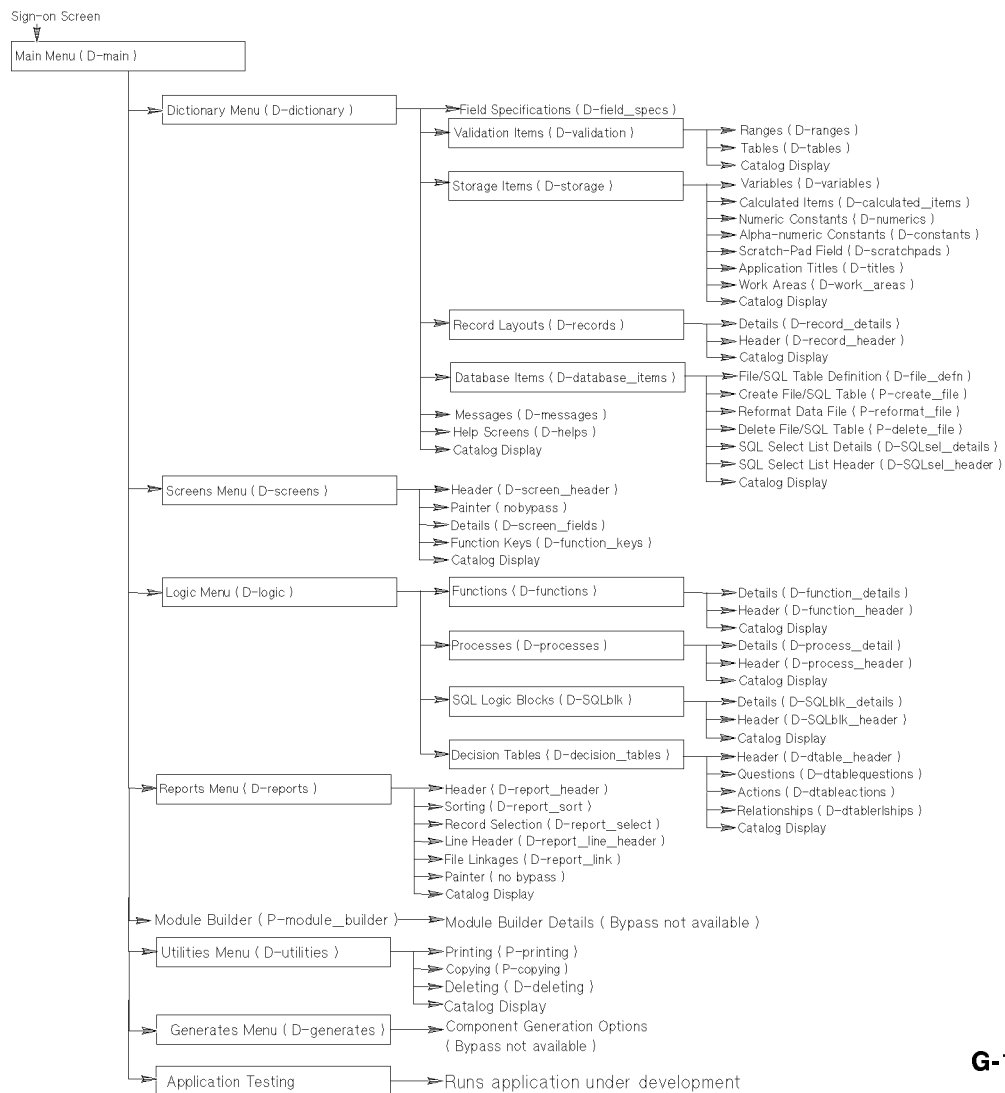
These facilities are not provided under other operating systems.

Refer to the *HP-UX HP ALLBASE/4GL Developer Administration Manual* for more information about these facilities.

FINAL TRIM SIZE : 7.0 in x 8.5 in



Developer Reference Chart



FINAL TRIM SIZE : 7.0 in x 8.5 in

Glossary

This glossary explains the meaning of some of the terms and words used with HP ALLBASE/4GL. In some cases, the terms used in HP ALLBASE/4GL may differ in meaning from the same terms used in a conventional programming environment.

A

Abort. A message type that informs the user that a serious error has been detected. The HP ALLBASE/4GL session terminates after the user acknowledges the message.

***ABSENT.** VALIDATE command argument that is used to check if the specified KSAM file record field does not exist.

administ. The name of the administrator application. It is a reserved name.

Administrator. The person who controls the HP ALLBASE/4GL site configuration and controls access by developers and users.

administrator. The application used by the system administrator to control the system-wide site configuration. Note that this name has a lowercase "a".

***ALL.** OFF and ON logic command argument for referencing all user switches 1 through 8.

***ALPHA.** IF and IFLOOP logic command argument used for checking if a field contains only alphabetic characters.

B

BACKGRND. Logic command for executing a process in background mode. Only an HP ALLBASE/4GL process can be executed in this manner. The process may execute any facility except displaying a screen. A background

process cannot send messages to its originating terminal or accept input from the terminal.

***BLANK.** IF and IFLOOP command argument used to check if a field contains only space characters.

***BUFFER.** FILE command argument used to clear a file record buffer. This command also unlocks a file record that has been locked to the file buffer.

***BYPASS.** A switch used by the report generator. When the switch is set *on*, it indicates that a record is not to be processed.

C

CALC. Logic command for performing a mathematical operation.

Calculated Item. A variable that is evaluated each time it is referenced. In the simple form, the calculated item is the result obtained by evaluating a CALC logic command. In the extended form, HP ALLBASE/4GL executes a function each time the calculated item is referenced. In this case, the calculated item function assigns a value to the calculated item.

CALL. Logic command that provides dynamic loading of routines written in the C programming language.

CHECK. Logic command for matching data against an indexed file record or table. The CHECK command places the matching position number into the communication area field *PASS.

CLEAR. Logic command for clearing all or part of the screen or scratch-pad.

***CLOSE.** FILE command argument used to close the nominated file.

***COMMIT.** Command argument used with the EXIT command in an after function on a data screen input field, a prior function on a display-only field on a data screen, or a function called from a function key on a data screen. This argument terminates the current screen in the same way as the I-COMMIT internal routine.

Communication Area Field. A field addressable by HP ALLBASE/4GL and the logic of the application. Communication area fields can be either alterable or read only. Any non-numeric communication area field can be

Glossary-2

referenced using the substring conventions. Numeric communication area fields are held as numbers and cannot be referenced by substring.

***COUNT(n).** Alterable communication area fields used primarily in the report generator for counting the number of times a nominated report line is printed. HP ALLBASE/4GL has five line counter fields, identified as *COUNT(1) through *COUNT(5). These fields are treated as numeric.

***CROSS(n).** Alterable communication area fields used primarily in the report generator for across-the-page totalling of numeric fields. HP ALLBASE/4GL has five cross add fields, *CROSS(1) through *CROSS(5). These fields are treated as numeric.

D

Data item. The smallest accessible data element in an HP TurboIMAGE/iX database. Within HP ALLBASE/4GL, data items are defined as field specifications.

Data entry. An ordered set of related data items within an HP TurboIMAGE/iX database. Within HP ALLBASE/4GL, data entries are defined as record layouts.

Data set. A collection of data entries, or records, within an HP TurboIMAGE/iX database.

Data screen. The normal HP ALLBASE/4GL screen on which the user can enter data, and on which the system displays information. Data screens can contain a scrolling area or a window or both. The SCREEN logic command or the D-action on a menu initiates display of a data screen.

***DATE.** Read-only communication area field containing the current date. This field may be referenced by substring.

Date Format. Each HP ALLBASE/4GL site has one date format (specified by the System Administrator) that applies to all developers and users. The format is either the US date format MM/DD/YY, or the European date format DD/MM/YY. Regardless of the date format, HP ALLBASE/4GL stores all dates on file records in the format YYMMDD. The chosen system-wide data format controls the presentation of dates on screens and reports.

DECISION. Logic command for executing a decision table.

Decision table. A decision table consists of up to 8 questions, 31 different combinations of outcomes for the questions, and up to eight actions. For each of the combinations of outcomes, the decision table specifies which actions are to be performed, and the sequence in which they are performed.

DEFINE. Logic command that allows you to define a macro in a logic block. The DEFINE command substitutes an expression for a one to four character string surrounded by % signs.

***DELETE.** FILE command argument that specifies file record deletion.

developr. The name of the developer application.

DISPLAY. Logic command for clearing and displaying data within the scroll area of a data screen.

DM IMAGE *CLOSE. Logic command used to close HP TurboIMAGE/iX databases.

DM IMAGE *LOCK. Logic command for locking an HP TurboIMAGE/iX database, or specified data sets or records within an HP TurboIMAGE/iX database.

DM IMAGE *MODE. Logic command to specify the access mode used to access parts of one or more HP TurboIMAGE/iX databases. The command does *not* actually open any data sets.

DM IMAGE *UNLOCK. Logic command for releasing all locks held on one or more HP TurboIMAGE/iX databases.

E

Edit code. A code assigned to a field specification, a screen field, or a report field that defines the type of data stored or displayed in the field. Refer to chapter 3 of this manual for an explanation of the HP ALLBASE/4GL edit codes.

***ENDFIELD.** A switch used by the screen processing logic. A function called from a function key can set this switch *on*. If *ENDFIELD is *on*, normal input processing for the current field is terminated, and processing continues at the next field on the screen when the function exits.

Glossary-4

***ENDLINE.** A switch used by the report generator. When this switch is *on*, it indicates that no more components of the current line group are to be printed, or no more records are to be read from the current link file.

ENTER. Logic command specifying the step number of the next command step to be executed in the current logic block. The ENTER command is equivalent to an unconditional GOTO command.

***ENTERED.** A switch used by the screen processing logic. This switch is *off* if the user has not entered data into a screen field, and is set *on* after the user enters data.

Error. A message type that informs the user of an error condition that must be corrected.

EXIT. Logic command that terminates the current logic block.

EXTERNAL. Logic command that passes control to an external program written in a language other than HP ALLBASE/4GL.

F

FIELD. Logic command for altering the behavior or attributes of a field on the current screen.

***FIELDNO.** An alterable communication area field that contains the current screen field number. It is treated as a numeric.

Field Specification. A dictionary entry that defines the characteristics of a field. A field specification defines the editing and data validation characteristics of a field.

FILE. Logic command for specifying reading from, writing to, or deleting from a file.

***FILENAME.** Communication area field that contains the external name for the current or most recently accessed KSAM file or serial file. This field is also used for dynamic naming of serial files. You can specify the external name for a file as *FILENAME, and then move the appropriate name into *FILENAME to specify the file to be accessed at run time.

***FIND.** FILE command argument used for locating a record on an KSAM data file. The FILE *FIND command positions the file pointer at the first

record with a key value equal to or greater than the value in the key field of the record buffer, or the value specified by the *KEY= argument.

***FIRST.** FILE command argument used for specifying that the first record on a serial or KSAM file is to be read.

***FUNCTION.** Alterable communication area field that contains the name of the function being executed (or blank if not currently executing a function).

Function. HP ALLBASE/4GL logic entity. A function is similar to a subroutine in a conventional language system. It consists of between 1 and 99 lines of logic commands.

G

Generate. Validate and transform HP ALLBASE/4GL source parameters to executable format.

H

Hash. The character #. This character is used in the MATH and MATHLOOP commands to specify that a number is to be treated as an absolute number.

HP ALLBASE/SQL. A relational database management system which operates on both the HP-UX and MPE XL operating systems.

HP TurboIMAGE/iX. A network database management system that operates on the MPE XL operating system.

I

IF. Logic command for performing conditional tests.

IFLOOP. Logic command for repeated conditional testing.

***INDEX=.** Command argument for the CHECK, FILE, and VALIDATE commands. This argument specifies the index to be used to access the file for the particular file access. Using *INDEX= with these commands does not change the current value in *INDEXNO.

***INDEXNO.** Alterable communication area field that contains the number of the file index to be used for subsequent accesses of an application data file. This field is treated as numeric.

Glossary-6

***INSERT.** FILE command argument used for specifying that the current contents of the file buffer are written to the file.

***IOSTATUS.** Alterable communication area field that contains the return status code from the data manager after accessing an application data file. This field contains the value zero if no error is encountered. If a file error does occur, the HP ALLBASE/4GL data manager writes a file error code to this field. Refer to Appendix C, Data Manager Error Messages for a summary of the file error codes. This field is treated as numeric.

J

***JOINER=.** LINK and LINKLOOP command argument for specifying insertion of a character or string of characters between each field to be concatenated.

Justification. Alignment of data on a boundary or margin. HP ALLBASE/4GL provides automatic justification of data within the left and right boundaries (minimum and maximum length) of the field. The data may be justified on the first character position (left justified) or the last character position (right justification). Data may also be centered within the field, or no justification need be performed.

K

***KEY=.** CHECK, FILE, and VALIDATE command argument specifying the key value used to access KSAM files.

KEYS. Logic command to display a set of function keys on the screen.

L

***LAST.** FILE command argument specifying that the last record in an indexed file is to be read.

LENGTH. Logic command for calculating the length of the contents of a field.

LINK. Logic command for concatenating nominated fields.

LINKLOOP. Logic command used for concatenating a number of consecutive fields.

Literal. A string of characters surrounded by quotes.

Logic block. The commands that make up a process or function. Each logic block contains from 1 to 99 lines of logic commands.

M

***MAP.** CLEAR command argument that specifies that the screen or scratch-pad field position is to be mapped (within the specified screen or the scratch pad).

MATH. Logic command for performing an arithmetic operation.

MATHLOOP. Logic command for performing repeated arithmetic operations.

Mess. A message type that is displayed for information only and does not require a response from the user.

MESSAGE. Logic command for displaying a message on the screen.

MODE. Logic command used for specifying the names of the files that are being used in the current process. This command can only be used in a process. Files can be defined as being updated or locked to the current process. Files that are not defined as being updated are available for reference only. The MODE command remains in effect until the next process is executed.

***MODIFY.** FILE command argument specifying that the contents of the file buffer are to be used to update or change an existing record on a fixed length record serial file or KSAM file. This record must be the last one read for that file.

***MOREREC.** A switch used in conjunction with variable length record serial files. If HP ALLBASE/4GL reads sufficient characters to fill the file record buffer without encountering a newline character when reading from a serial file with variable length records, the data manager sets the switch *MOREREC *on*. If HP ALLBASE/4GL encounters a newline character before the buffer is filled, or the number of characters read exactly matches the length of the buffer, *MOREREC is set *off*.

MOVE. Logic command for copying contents of one field (or literal) to another field. The MOVE command can also copy the contents of one buffer to another.

Glossary-8

MOVELOOP. Logic command for repeated copying of one set of fields to another.

N

***Name Recall.** A context-sensitive recall facility that allows developers to scroll through the names of items that have been defined for a field type, when that field is the active field on the screen.

***NEWTIE.** An alterable communication area field that may be set to contain the next screen field number to be selected. Moving a value into *NEWTIE performs the same function as the logic command TIE. This field is numeric.

***NEXT.** FILE command argument specifying that the next record on the file is to be read.

NOTE. Logic command that allows the insertion of a comment in a logic block. HP ALLBASE/4GL ignores NOTE command steps when it executes the logic block.

***NULL.** IF and IFLOOP command argument for testing the status of the null indicator variable associated with fields on record buffers for HP ALLBASE/SQL tables and select lists. Also used as a command argument for the MOVE command to set the indicator variables to the null status.

***NUMERIC.** IF and IFLOOP command argument used for checking if a field contains only numeric characters. (That is, 0 through 9, +, -.)

O

***OFF.** IF and IFLOOP command argument used for testing if a switch is currently *off*.

OFF. Logic command that sets a switch to its *off* state.

***ON.** IF and IFLOOP command argument used for testing if a switch is currently *on*.

ON. Logic command that sets a switch to its *on* state.

P

***P.** CLEAR command argument indicating that the field or fields to be cleared are scratch-pad fields. By itself, the *P argument indicates that all scratch-pad fields are to be cleared.

Pad character. A specified character used to fill a field. Typically a space or 0.

***PAGELINE.** Alterable communication area field containing the number of the current print line on the current page of a report. This field is treated as numeric.

***PAGENO.** Alterable communication area field containing the current page number of a report. This field is treated as numeric.

***PASS.** Alterable communication area field used to pass arguments and results. For example, the CHECK command places the matching position for data in a file or table into *PASS. This field may be referenced by substring.

***PRESENT.** VALIDATE command argument used to check if the specified KSAM data file record field exists.

***PREVFLD.** Communication area field containing the number of the last screen field to be successfully committed.

***PREVIOUS.** FILE command argument specifying that the previous record on a fixed length record serial file or KSAM file is to be read.

PRINT. Logic command to print a defined report line group from within a function that has been called during the generation of a report. The PRINT command is only available during report generation, and is ignored at any other time.

PROCEED. Logic command to execute a process logic block.

Process. An HP ALLBASE/4GL process is similar to a program in a conventional language system. Each process logic block contains from 1 to 30 lines of logic commands.

***PROCESS.** Alterable communication area field name containing the name of the current process.

Glossary-10

Q

Query. A message type that requires the user to enter a response.

R

Range. A validation range specifying the lower and upper limits for the contents of a field.

***RANGE=.** Report generator record selection option that specifies the name of an HP ALLBASE/4GL range to be used for testing the contents of a specified field when selecting records for reporting.

***READ.** FILE command argument specifying that the record with the given key is to be read.

***RECNO.** Communication area field that contains the number of the record just read or written for a fixed length record serial data file. The value in *RECNO is undefined after HP ALLBASE/4GL accesses any other data file type.

***REFRESH.** SHOW command argument specifying that the contents of the nominated fields are to be obtained from the main or default data movement field.

*REFRESH is also a command argument for the EXTERNAL command. If *REFRESH is specified, HP ALLBASE/4GL restores the terminal configurations and refreshes the screen on return from an external program.

***REPORT.** Alterable communication area field containing the name of the report being executed (or blank if not currently executing a report).

REPORT. Logic command to execute a report

***RESET=.** DISPLAY command argument that clears all lines from the nominated line to the end of the scroll area before any further display within the scroll area.

***ROUTINE.** Alterable communication area field which contains the name of the current external routine (or blank if not currently in a called routine).

***ROWCOUNT.** Communication area field containing a number indicating the number of rows that HP ALLBASE/SQL processes as the result of a command that changes an SQL table.

S

***S.** CLEAR command argument indicating that the field or fields to be cleared are screen fields. When used by itself, *S indicates that all fields on the screen are to be cleared.

Schema

Scratch Pad. The HP ALLBASE/4GL scratch pad contains up to 99 scratch-pad fields. The scratch-pad fields are available for temporary storage of data. The scratch pad is dynamic and each scratch-pad field takes on the attributes of the data moved into it.

***SCREEN.** Alterable communication area field containing the name of the current screen (or blank if no screen).

SCREEN. Logic command to display a screen.

SCROLL. Logic command that displays data in the scroll area of a screen.

SELECT. Logic command that executes one of a series of commands from a selection list. The system executes the command identified by the value contained in the communication area field *PASS.

Select list. A select list is a “virtual” SQL table. The columns of a select list are taken directly from an existing table or view, or computed using an expression or aggregate function. HP ALLBASE/4GL automatically builds a record layout for each select list.

SERIES. Logic command that executes the specified command step number, or range of command steps in the current logic block. After the nominated commands have been executed, the command immediately following the SERIES command is then executed.

SHOW. Logic command that displays the current contents of a specified screen field or fields. When used with the *REFRESH option, the SHOW command refreshes the screen buffer from the main or default data movement fields.

***SHOWING.** A switch used by the screen processing logic. When the SHOW command is displaying the contents of a field, *SHOWING is set to *on*.

Glossary-12

SQL Logic Block. Used for passing SQL commands from HP ALLBASE/4GL to HP ALLBASE/SQL. An SQL logic block can contain up to eight SQL commands. These commands are passed to HP SQL when the SQL logic block is invoked by an SQL command in an HP ALLBASE/4GL logic block.

Subscript. An integer or integers used to refer to a specific occurrence of a file record field when the field is specified as occurring more than once. HP ALLBASE/4GL subscripts are enclosed in parentheses.

Substring. A portion of the contents of a field. A substring is referenced by giving its starting character position in the field, a comma, and the length (in characters) of the substring, all enclosed in square brackets. For example, [3,2] is a substring of length 2, starting at character position number 3.

***SUITE.** Read-only communication area field containing the name of the application being run. If the current application is a version derived from a base application, *SUITE contains the name of the base application.

T

***TABLE=.** CHECK command argument that identifies the table against which checking is to be performed.

TIE. Logic command for specifying the next screen field to be selected. Used where the next screen field to be accessed is not necessarily the next field in sequence on the screen.

***TIME.** Read-only communication area field containing the current time. This field may be referenced by substring.

TOP. Logic command for specifying that step number 1 of the logic block is the next step to be executed.

***TOTALS(n).** A set of 16 alterable communication area field used primarily in the report generator for totalling the values of numeric fields. The fields are identified as *TOTALS(1) to *TOTALS(16). These fields are treated as numeric.

Training mode. Operating mode used to prevent the updating of application data files from a specific user. While in training mode, all

requests to update data files are ignored. The system administrator can set training mode *on* permanently for any user.

TRANSACTION. Logic command that defines blocks of logically related file transaction commands.

U

UPDATE. Logic command that updates all KSAM and serial files whose buffer contents have been modified. This command has the same effect as issuing a FILE *WRITE command for each relevant file.

***USER.** Read-only communication area field containing the name of the current user.

V

VALIDATE. Logic command used to check if the contents of a specified field are missing (*ABSENT) or present (*PRESENT) on the relevant KSAM file record.

Version. An application that has been modified from a base application for a particular end user or group of end users.

***VERSION.** A read-only communication area field that contains the name of the current version being run.

VISIT. Logic command for executing a function logic block.

W

Warn. A message type that beeps the terminal bell to alert the user, but does not require a response.

WINDOW. Logic command to display a window on the current screen.

Window. Screen that is overlaid on the current screen starting at the line number defined for the current screen. Window screens operate exactly as normal data screens. However, they may not contain windows or scroll areas. Window screens cannot be displayed independently of a normal data screen.

***WRITE.** FILE command argument used for specifying that the contents of the file buffer are to be written to a serial file or KSAM file.

Glossary-14

X

(no entry)

Y

(no entry)

Z

ZIP. Logic command that performs no operation and allows the next command to be executed.

FINAL TRIM SIZE : 7.0 in x 8.5 in

Index

A

abbreviations
 in logic, 12-31
 in menu selections, 2-9

aborting the screen painter, 6-62

abort messages, 6-15

absolute operands, 12-8, 12-134, 12-139

access class, HP ALLBASE/SQL tables, 9-23

access class, SQL tables, 5-52

access mode, 7-13

account structure (MPE/iX), D-8

accumulating totals in reports, 7-10

across the page totals in reports, 7-56

action prefixes, 3-21

action prefixes, menu bypass, 14-6

actions
 menu, 6-1, 6-51

activate item function key, 6-40

administrator application, 1-3

after function if blank, 6-34

after functions, 6-7, 6-16

after link read function, 7-19, 7-43, 7-46

after-print function, 7-21, 7-41

after selection function, 7-3, 7-34, 7-44

alphanumeric constants screen, 5-33

alterable communication area fields, 4-1

“American” NLS collating sequence, A-1

another report function key, 7-54

application
 generation, 14-1, E-2

 generation menu, 14-2

 portability, F-1

 security, 1-14

 testing, 14-6

 title referencing, 5-38

 titles, 3-19, 12-4

 titles screen, 5-37

 unloading, 14-8

application data files, 1-15

application definition file, F-2

argument by reference with the CALL command, 12-13

argument by value with the CALL command, 12-13

argument type with the CALL command, 12-13

automatic data movement, 6-18

automatic data validation, 6-17

automatic field commit, 6-33

automatic flow to next, 6-33, F-5

automatic screen field numbering, 6-27

automatic SQL commands, 9-6

available space function key, 6-58

B

BACKGRND command, 12-5

background process
 limitations, 12-7
 message file, 12-6

base application, B-1

batch processing, see BACKGRND command, 12-5

Index-1

- before-print function, 7-19, 7-40
- *BEGIN argument, TRANSACT command, 12-184
- BEGIN WORK command, 9-7
- blank fields on screens, 6-34
- blank when zero, 5-6
- block copying in screen painter, 6-59
- block deletion in screen painter, 6-60
- block move in screen painter, 6-59
- both functions, 6-7
- bottom of page line group, 7-21
- buffer
 - data set record, 12-99
 - file record, 5-56, 12-72, 12-85
 - screen field, 6-4
 - select list, 12-91
 - SQL table, 12-91
 - work area, 5-40
- *BYPASS, 4-13, 7-8, 7-19, 7-21, 7-34, 7-41, 7-44
- bypassing fields on screens, 6-12
- bypassing file linkages, 7-19
- bypassing link file records, 7-44
- bypass selection on menus, 2-11

C

- CALC command, 12-8
- calculated item
 - defined, 1-6
 - definition screen, 5-27
 - extended, 5-27
 - function, 5-27
 - generation, 5-30, 14-1
 - referencing, 3-17
 - usage, 5-30
- calculations in reports, 7-11
- CALL command, 12-16
 - argument type, 12-13
 - C language pointer, 12-13
 - reference, 12-13
 - syntax, 12-18

- terminal setting, 12-12
- value, 12-13
- cancelled transactions, 12-185
- catalog
 - copying screen, 13-7, B-2
 - deletions screen, 13-9
 - display screen, 13-11
 - item details, 13-4
 - printing screen, 13-4
- century overflow errors, 12-28
- changing screen literals, 6-50
- CHECK command, 12-19
- C language pointer with the CALL command, 12-13
- clashing items on screens, 6-56
- “clean” S-files, D-3
- CLEAR command, 12-23
- clearing data set buffers, 12-99
- clearing file buffers, 12-72
- clearing screen fields, 6-27, 6-33, 6-39, 12-24
- clearing the message display, 5-78
- clearing the scroll area, 12-34
- clearing work areas, 5-40
- clear on commit, 6-27
- *CLOSE, 12-71, 12-98
- CLOSE CURSOR command, 9-7, 12-92
- closing data sets, 12-98
- closing files, 12-71, 12-72
- closing HP TurboIMAGE/iX databases, 10-7, 12-36
- closing HP TurboIMAGE/iX data sets, 10-8
- clustering index, 5-44
- collating sequence, 7-32, 12-113, 12-119, A-1
- color terminals, 6-54
- column entry mode, 6-60
- column heading line group, 7-21
- column names, 5-53
- command file, startup, D-19

Index-2

command line options, 2-25
 commit actions, screen and field, 6-11
 commit data action, 2-18, 6-18
 commit data function key, 6-39
 commit details function key, 7-53
 *COMMIT in EXIT command, 6-18
 committing screens, 6-39, 12-52
 COMMIT WORK command, 9-7, 12-92
 *COMMS argument, EXTERNAL
 command, 12-56
 communication area, 4-1
 communication area field referencing,
 3-19
 communication area fields
 *COUNT, 4-1
 *CROSS, 4-2
 *DATE, 4-11
 *DBENAME, 4-11
 *ERROR, 4-2
 *ERRORDM, 4-3, 12-106
 *FIELDNO, 4-3, 6-8
 *FILENAME, 4-4
 *FUNCTION, 4-5
 *IMSTAT, 4-5, 12-106
 *INDEXNO, 4-5
 in report generation, 7-8
 in screen processing, 6-8
 *IOSTATDM, 4-6, 12-106
 *IOSTATUS, 4-6
 *KEYS, 4-11
 *LOCKWAIT, 4-7, 12-77, 12-109
 *LTRACE, 4-7
 *MSGDELAY, 4-8
 *NEWTIE, 4-8, 6-8
 *PAGELINE, 4-8
 *PAGENO, 4-8
 *PASS, 4-8
 *PREVFLD, 4-11, 6-8
 *PROCESS, 4-9
 *RECNO, 4-9
 *REPORT, 4-9
 *ROUTINE, 4-9
 *ROWCOUNT, 4-9
 *SCREEN, 4-10, 6-8
 *SUIE, 4-12
 *TIME, 4-12
 *TOTALS, 4-10
 *USER, 4-12
 *VERSION, 4-12
 concatenating fields, see LINK(LOOP)
 command, 12-127, 12-130
 configuration, terminal, D-25
 CONNECT command, 9-6
 connecting to a database, 14-6
 constant referencing, 3-18
 control break, 7-4, 7-10, 7-18, 7-32
 control break levels, 7-16
 copying fields on reports, 7-54
 copying screen items, 6-58
 copying S-files, D-3
 *COUNT, 4-1, 7-8, 7-40
 counting report lines, 7-40
 creating serial files, 5-59, 12-83
 creating temporary serial files, 5-59
 critical linkages, 7-7, 7-37, 7-44, 7-45
 *CROSS, 4-2, 7-8, 7-21, 7-56
 cross add fields, 7-56
 C type report lines, 7-5, 7-52, 7-54
 cursor commands (SQL), 9-7
 cursor control in report painter, 7-48
 cursor control keys, 2-9, 2-17
 cursor movement, screen painter, 6-48
 cursor position after delete (SQL), 9-16
 cursor position (SQL), 9-16
 cursors, declaring and opening (SQL),
 12-89
 customized applications, see versions,
 B-1

D

D1 line group, 7-6, 7-17
 D-action, 6-9

- database
 - HP TurboIMAGE/iX, 10-2
- database environment, 5-52
- database items menu, 5-48
- database module file, 14-5, D-14, F-3
- database sections, 9-4
- data display, scrolling on screen, 12-167
- data, entering and editing, 2-15
- data file
 - application, D-7
 - creation, 5-59
 - creation screen, 5-61
 - definition screen, 5-50
 - dynamic naming, 5-55
 - group and account, 5-55
 - locking, 12-76
 - opening, 12-71
 - protection during testing, 14-7
 - reformat, B-4
 - reformat screen, 5-64
 - temporary, 5-56
- data file recovery, F-8
- data files, 1-15
- data manager, 1-2
- data manager specific field attributes, 5-10
- data movement, 6-16, 6-18
- data movement fields
 - control of, 6-3
 - default, 6-6, 6-32
 - definition, 6-4
 - others, 6-6, 6-32
 - primary, 6-5, 6-31
 - refresh, 12-176
- data movement in SHOW command, 6-21
- data screen by module builder, 8-21
- data screen processing logic, 6-3
- data screens, 6-1, 6-49
- data set
 - indexes, 5-44
 - locking, 12-106
 - opening, 12-98
- data set error handling, 12-105
- data set indexes, 12-98
- data set operation, 12-100
- data set record buffers, 12-99
- data set reformatting, 5-67
- data types, 5-8
- data validation
 - by secondary file, 8-9
 - decimal places, 6-15
 - default error message, 6-16
 - edit code, 6-15
 - error detection, 6-16
 - error messages, 6-15
 - in after functions, 6-16
 - number of characters, 6-15
 - range, 5-7, 5-18
 - range checking, 5-19
 - screen field processing, 6-3, 6-17
 - table, 5-7, 5-20
 - table checking, 5-21
 - validation ranges or tables, 6-15
- *DATE, 4-11
- DATE command, 12-26
- date errors, 12-26, 12-28
- date literals, 12-4, 12-26
- dates in logic commands, 12-4
- day of the week, 12-26, 12-28
- DBCHANGE function key, 6-40
- DBEFileset, 5-52
- *DBENAME, 4-11
- DB IMAGE *MODE command, 12-98
- DBUTIL function key, 6-40
- debugging, see trace mode, 14-7
- decimal places, MATH(LOOP)
 - command, 12-137, 12-141
- DECISION command, 12-30
- decision table
 - usage, 1-10
- DECLARE CURSOR command, 9-6

Index-4

DECLARE CURSOR for **SELECT**
 command, 9-9
 default data movement field, 6-4, 6-6,
 6-12, 6-32, 12-176
 default function keys
 actions, 6-43
 definitions, 6-43
 system keys, 6-43
 default help screen, 5-80
 default_keys, 6-43
 default record, 5-44, 5-54, 5-56, 12-69,
 12-81, 12-95
DEFINE command, 8-28, 12-31
DELETE command (SQL), 9-16
 deleted items, D-15, E-2
DELETE WHERE CURRENT
 command, 9-18
 deleting fields on reports, 7-54
 deleting HP ALLBASE/SQL data via
 a cursor, 9-18
 deleting HP ALLBASE/SQL data
 without a cursor, 9-20
 deleting screen items, 6-59
 description fields, 6-52
 developer main menu, 2-7
 developer user name, 2-3
 development security code, 1-14, 2-4
 dictionary fields
 on reports, 7-54
 on screens, 6-31
 specifications, 6-52
 dictionary menu, 5-2
 dictionary select list name, 9-6
 dictionary table name, 9-6
 difference between dates, 12-27
 disabled logic commands, 12-2
DISPLAY command, 12-33
 display highlighting, 6-56
 display highlighting, special text items,
 6-53
 displaying data in a scroll area, 12-33

display logic function keys, 6-38
 display-only field, 6-12
 display scrolling in report painter, 7-49
DM IMAGE *CLOSE command, 12-36
DM IMAGE *LOCK command, 12-37
DM IMAGE *MODE command, 12-107
 done list, 8-19
DROP MODULE command, 14-5
 duplicate key errors, 12-73, 12-74
 duplicate keys, 5-45
 dynamic file naming, 4-4, 5-55, 12-84

E

echo, 12-64
 edit codes, 3-21, 5-5
 edit code validation, 6-15
 editing fields, 2-15
 eight-bit characters, 2-6
***END** argument, **TRANSACTION** command,
 12-184
***ENDFIELD**, 4-13, 6-8
***ENDLINE**, 4-13, 7-7, 7-8, 7-19
 end of file errors, 12-75
 end of link line group, 7-7, 7-19, 7-43,
 7-46
 end-of-report function, 7-9, 7-18, 7-29
ENTER command, 12-50
 enter details function key, 7-53
***ENTERED**, 4-13, 6-7, 6-8, 6-12, 6-34
***ERROR**, 4-2
***ERRORDM**, 4-3, 12-106
 error handling
 CALC command, 12-9
 DATE command, 12-26
 FILE command, 12-70, 12-74, 12-81,
 12-90, 12-96, 12-105
 files, 12-84
 IF command, 12-114
 IFLOOP command, 12-120
 MATH command, 12-136
 MATHLOOP command, 12-140

- VALIDATE command, 12-190
- error log file, generate errors, 14-5
- error messages
 - during generation, 14-3
 - HP ALLBASE/SQL, C-8
 - HP TurboIMAGE/iX data sets, C-3
 - HP TurboIMAGE/iX files, C-11
 - KSAM files, C-3
 - MPE/iX, C-1
 - serial files, C-3
- error processing, 6-15
- error type message, 5-18, 5-20, 6-15
- EXIT command, 12-52
- EXIT *COMMIT command, 6-11, 6-18, 12-52
- exiting from HP ALLBASE/4GL, 2-24, 6-40
- extended calculated item, 5-27
- EXTERNAL command, 12-53
- external names for files, F-6
- external programs
 - operating system dependencies, F-9
 - passing parameters, 12-53

F

- fatal error, 14-6
- FETCH command, 9-9, 12-93
- field and screen commit actions, 6-9
- FIELD command, 12-64
 - echo, 12-64
 - no echo, 12-64
- field commit action, 2-16, 6-11, 6-15
- field commit action, new commit, 6-15
- field commit, automatic, 6-33
- field committed in error, 6-15
- field details screen, screens menu, 6-30
- field display logic, 6-12
- field editing, 2-15
- *FIELDNO, 4-3, 6-8, 6-12, 6-16
- field numbering, on windows, 12-193
- field referencing rules, 3-5

- field specification names, 6-52
- field specifications, 1-5
- field specifications screen, 5-4
- field types, on screens, 6-48
- file
 - application definition, F-2
 - creation screen, 5-61
 - data, D-7
 - database module, D-14, F-3
 - data file recovery, F-8
 - deleted records in S-files, D-15
 - dynamic naming, 5-55, 12-84
 - error handling, 12-74, 12-84, 12-90
 - error messages, 12-74, 12-105
 - errors, common, 12-90
 - external names, 5-55, F-6
 - file structure file, F-3
 - indexes, 5-44, 12-71
 - key field limits, F-7
 - linkage number, 7-35
 - linkages, 7-12, 7-19, 7-37
 - linkages in reports, 7-6
 - linkages screen, 7-42
 - locking, F-7
 - locking for HP TurboIMAGE/iX data sets, 12-106
 - locking for KSAM data files, 12-76
 - naming, dynamic, 5-55, 12-84
 - opening, 12-71
 - operations, 12-72
 - position pointer, 12-73
 - program, D-4
 - record buffer referencing, 3-12
 - record buffers, 5-56, 12-72
 - record field references, 3-11
 - serial file creation, 5-59
 - S-file indexes, D-2
 - S-files, D-2
 - S-files, copying, D-3
 - temporary, 5-56, 12-83
 - temporary serial file creation, 5-59

Index-6

- terminal support, D-23
- transaction management, F-8
- FILE command
 - *DELETE operation, 9-16
 - HP TurboIMAGE/iX, 10-8
 - (HP TurboIMAGE/iX), 12-94
 - (KSAM), 12-68
 - (serial), 12-80
 - (SQL), 12-88
- file creation screen, 5-61
- file creation, temporary files, 5-56
- file designator, report, 7-2, 7-26
- file equations, 5-60
- file error messages, C-1
- file errors, common, 12-75
- file errors, common conditions, 12-84
- FILE *FIRST command, 7-12
- FILE (HP TurboIMAGE/iX) command, 12-94
- file linkages, 7-19
- file linkages, limitations, 7-46
- file locking
 - DM IMAGE *LOCK command, 12-37
- *FILENAME, 4-4, 5-55, 12-72, 12-83
- file reformat, in run-time environment, 5-67
- file size, S-files, D-15
- file/SQL table definition, 5-50
- file/SQL table deletion screen, 5-68
- file transaction reversal, 12-185
- fixed function switches, 4-13
- floating point arithmetic, 12-137, 12-141
- forcing a screen commit, 6-18, 12-52
- formal file designators
 - HP4GNERR, 14-5, D-12
 - HP4KYIN, D-11
 - HP4KYOUT, D-12
 - HP4REP, D-12
 - HP4RFLOG, D-12
 - HP4SCOUT, D-11
 - HP4TRACE, 14-8, D-12
- full index, 13-5
- *FUNCTION, 4-5
- function
 - after, 6-7
 - after selection, 7-34
 - both, 6-7
 - calculated item, 5-27
 - called from function key, 6-15, 6-18
 - definition of, 1-9
 - end of report, 7-9, 7-29
 - in reports, 7-8
 - on show, 6-22
 - post-link read, 7-43
 - prior, 6-7
 - screen field, 6-7, 6-34
 - show, 6-7
 - start-of-report, 7-16
 - start of report, 7-9, 7-29
- function keys
 - action prefixes, 6-37
 - activate item, 6-40
 - another report, 7-54
 - available space, 6-58
 - backtab, 6-39
 - commit data, 6-39
 - commit details, 7-53
 - default definitions, 6-43
 - definition screen, 6-36
 - display logic, 6-38
 - enter details, 7-53
 - for window, 12-193
 - help screen, 2-22, 5-81
 - introductory description, 2-11
 - main menu, 6-41
 - more keys, 2-14
 - Op System, 6-42
 - previous keys, 6-41
 - previous menu, 6-41
 - report painter, 7-49
 - restore original, 6-51
 - screen painter, 6-47, 6-58

- sign-on screen, 6-41
- special text, 6-53
- start again, 6-39
- switches, 6-37
- tab, 6-42
- window keys, 7-49
- windows, 12-193

function key sets

- name recall, 2-13

G

- generated components, 14-1
- generate error log file, 14-5
- generate error messages, 14-3
- generate program, 14-2
- generating application components, 1-13
- generation
 - applications, E-2
 - calculated items, 14-1
 - messages, 5-78, 14-1
 - program, 14-2
 - reports, 7-15
 - screens, 6-23
 - SQL commands, 9-6
 - SQL logic blocks, 9-21
- global_keys_1, 6-43
- global_keys_2, 6-44
- global_keys_2 for test mode, 6-44
- global_keys_3, 6-45
- GOTO, see ENTER command, 12-50
- grand total line group, 7-18
- group structure (MPE/iX), D-8

H

- header screens, 13-5
- help screens
 - default, 5-80
 - display, 5-80
 - for window, 12-194
 - function keys, 2-22, 5-81

- I-HELP routine, 6-40
- introduced, 2-22
- message help, 5-77
- name, 5-7
- screen, 5-79

H line groups, 7-18

host variable references, 3-20, 5-72

HP4APNPATh variable, 2-26, D-13, D-19

HP4ATOS utility, 13-13

HP4BG variable, D-14

HP4DATAPATH MPE/iX variable, 5-55

HP4DATAPATH variable, 2-26, D-13, D-19

HP4DBMPATH variable, 2-26, 9-21, 14-5, D-13, D-19

HP4DBMSIZE variable, 9-21, D-14

HP4FSPATH variable, 2-26, 5-63, B-4, D-13, D-19

HP4GNERR formal file designator, 14-5, D-12

HP4INV_PRT_CHAR variable, D-14

HP4KYIN formal file designator, D-11

HP4KYOUT formal file designator, D-12

HP4REMK utility, 13-20, D-16

HP4REOD utility, 13-21, D-16

HP4REP formal file designator, D-12

HP4RFLOG formal file designator, D-12

HP4SCOUT formal file designator, D-11

HP4SCREEN_PRINT variable, 6-41, D-14, D-22

HP4SPATH variable, 2-26, 13-20, D-13, D-16, D-19

HP4SQLPATH variable, 2-26, D-14, D-19

HP4STOA utility, 13-13

HP4TERM variable, 2-5, 2-26, D-14, D-19, D-23

Index-8

HP4TIPATH variable, 2-26, 10-2, D-14, D-19
 HP4TRACE formal file designator, 14-8, D-12
 HP4UPLD, 10-2
 HP ALLBASE/4GL administrator, 1-3
 HP ALLBASE/4GL terminology, 1-13
 HP ALLBASE/QUERY, 2-6
 HP ALLBASE/QUERY function key, 6-40
 HP ALLBASE/SQL
 error messages, C-8
 interface with HP ALLBASE/4GL, 1-11
 reports, 7-12
 table locking, 12-92, 12-145
 transaction management, 9-22
 HP ISQL function key, 6-40
 hpqm user name, 2-6
 HP QUERY/XL function key, 6-40
 HP TurboIMAGE/iX
 closing databases, 10-7
 closing data sets, 10-8
 data set reformatting, 5-67
 error messages, C-3
 interface with HP ALLBASE/4GL, 1-12
 intrinsics, 10-8
 library procedures, 10-8
 names, 3-4
 opening databases, 10-7
 opening data sets, 10-8
 reports, 7-13
 status array, 10-10
 upload utility, 10-2
 HP TurboIMAGE/iX database, 10-2
 HP TurboIMAGE/iX data set definition, 5-50
 HP TurboIMAGE/iX DBchange/XL function key, 6-40
 HP TurboIMAGE/iX DBUTIL function key, 6-40
 HP TurboIMAGE/iX file error messages, C-11
 HP TurboIMAGE/iX interface, 10-1
 HP TurboIMAGE/iX search items, 5-44
 HP TurboIMAGE/iX-specific field attributes, 5-10
 HP TurboIMAGE/iX terminology, 1-13
 HP TurboIMAGE\XL
 uploading dictionary definitions, 10-4

I

I-BACKTAB internal routine, 6-39
 I-CLEAR internal routine, 6-39
 I-COMMIT action, on menus, 6-40
 I-COMMIT internal routine, 6-11, 6-18, 6-39
 I-EXIT internal routine, 6-40
 IF command, 5-9, 12-111
 I-HELP internal routine, 6-40
 I-hpqm_call internal routine, 6-40
 I-image_dbchg_call, 6-40
 I-image_dbutil_call, 6-40
 I-image_query_call, 6-40
 I-isql_call internal routine, 6-40
 I-logon internal routine, 6-41
 I-MAINMENU internal routine, 6-41
 implicit SQL commands, 9-6
 *IMSTAT, 4-5, 12-106
 include in SHOW, 6-20, 6-33
 increasing S-file size, D-15
 indexes, files, 5-44
 index files, for S-files, D-2
 index list, 13-4
 index name, 12-71, 12-98
 *INDEXNO, 4-5, 12-70, 12-71, 12-96, 12-98
 index numbers, HP TurboIMAGE/iX data sets, 12-98
 index numbers, KSAM files, 12-71

- input field logic, 6-14
- input fields, 6-2, 6-12
- input fields on screens, 6-52
- input validation, 6-17
- INSERT command (SQL), 9-11
- inserting HP ALLBASE/SQL data, 9-10
- internal routines
 - I-BACKTAB, 6-39
 - I-CLEAR, 6-39
 - I-COMMIT, 6-39
 - I-EXIT, 6-40
 - I-HELP, 6-40
 - I-hpqm_call, 6-40
 - I-image_dbchg_call, 6-40
 - I-image_dbutL_call, 6-40
 - I-image_query_call, 6-40
 - I-isql_call, 6-40
 - I-logon, 6-41
 - I-MAINMENU, 6-41
 - I-PREVKEYS, 6-41
 - I-PREVMENU, 6-41
 - I-PRINT, 6-41
 - I-REFRESH, 6-42
 - I-RESET, 6-42
 - I-SHELL, 6-42
 - I-SLAVE_PRINT, 6-42
 - I-TAB, 6-42
 - I-TRACEMODE, 6-42
 - I-TRAIN, 6-42
- intrinsic
 - HP TurboIMAGE/iX, 10-8
- *IOSTATDM, 4-6, 12-106
- IOSTATDM communication area field, 10-10
- *IOSTATUS, 4-6, 7-45, 7-46, 12-105
 - KSAM files, 12-74
 - serial files, 12-84
 - SQL files, 12-90
- I-PREVKEYS internal routine, 6-41
- I-PREVMENU internal routine, 6-41

- I-PRINT internal routine, 6-41
- I-REFRESH internal routine, 6-42
- I-RESET internal routine, 6-42
- I-SLAVE_PRINT internal routine, 6-42
- I-TAB internal routine, 6-42
- item limits, E-1
- item naming by module builder, 8-25
- item referencing rules, 3-5
- items referenced during generation, 14-4
- item types on screens, 6-48
- I-TRACEMODE internal routine, 6-42
- I-TRAIN internal routine, 6-42

J

- jew for native language support, A-1
- job control words
 - NLDATALANG, 2-26, D-14, D-19
 - NLUSERLANG, 2-26, D-14, D-19
- Job control words
 - NLUSERLANG, D-17
- justification code, 3-23, 5-6
- justification of screen fields, F-5

K

- keyed access
 - module builder capabilities, 8-2
- key fields, 5-44, F-7
- *KEYS, 4-11
- KEYS command, 12-123
- KSAM data files, D-7
- KSAM file error messages, C-3
- KSAM file external names, 5-55, 5-63

L

- layout keys in screen painter, 6-58
- LENGTH command, 12-125
- length of logic step, 12-4
- library procedures
 - HP TurboIMAGE/iX, 10-8
 - status array, 10-10
- limitations

Index-10

- background processes, 12-7
- DEFINE command, 12-31
- on SQL commands, 9-4
- SELECT command, 9-6
- limits, E-1
- limits, S-file sizes, D-15
- line drawing character set, 6-55
- line group
 - bottom of page, 7-21
 - C, 7-5, 7-52, 7-54
 - column heading, 7-21
 - D1, 7-6, 7-17
 - end of link, 7-7, 7-19, 7-43, 7-46
 - grand total, 7-18
 - H, 7-18
 - identifiers, 7-5
 - in reports, 7-4, 7-38
 - link detail, 7-6
 - link from, 7-43
 - link print, 7-7, 7-19, 7-43
 - subheading, 7-4, 7-32
 - subtotal, 7-32
 - T, 7-10, 7-18
 - TF, 7-9, 7-18
 - top of page heading, 7-21
 - total, 7-4
- line header screen, 7-38
- line numbers, 7-39
- line printing logic, 7-21
- line types, reports, 7-4
- linkages
 - critical, 7-44
 - in reports, 7-6
 - limitations, 7-46
 - non-critical, 7-44
 - number, 7-35
 - record selection, 7-7
- LINK command, 12-127
- link detail line group, 7-6
- link file
 - initial read, 7-19
 - read failure, 7-45
 - reading, in reports, 7-43
 - selection criteria, 7-37
- link from line group, 7-43
- LINKLOOP command, 12-130
- link numbers, 7-42
- link print line group, 7-7, 7-19, 7-43
- link read function, 7-43
- literal fields on reports, 7-52
- literals
 - in logic commands, 12-4
 - on reports, 7-6, 7-52
 - on screens, 6-2, 6-49, 6-50
 - referencing rules, 3-18
 - used as dates, 12-4, 12-26
- *LOCK, 12-71, 12-98
- *LOCK argument, MODE command, 12-107, 12-147
- locked file errors, 12-75
- locked record errors, 12-74
- locking
 - HP ALLBASE/SQL tables, 12-92
 - HP TurboIMAGE/iX data sets, 12-106
 - KSAM data files, 12-76
 - KSAM files, F-7
 - serial data files, 12-85
- LOCK TABLE command, 9-24
- log file
 - background process messages, 12-6
 - generate errors, 14-5
 - trace mode, 14-8
- logic commands
 - BACKGRND, 12-5
 - CALC, 12-8
 - CALL, 12-16
 - CHECK, 12-19
 - CLEAR, 12-23
 - DATE, 12-26
 - DECISION, 12-30
 - DEFINE, 12-31
 - disabled, 12-2

DISPLAY, 12-33
 DM IMAGE *CLOSE, 12-36
 DM IMAGE *LOCK, 12-37
 ENTER, 12-50
 EXIT, 12-52
 EXTERNAL, 12-53
 FIELD, 12-64
 FILE (HP TurboIMAGE/iX), 12-94
 FILE (KSAM), 12-68
 FILE (serial), 12-80
 FILE (SQL), 12-88
 IF, 12-111
 IFLOOP, 12-117
 KEYS, 12-123
 LENGTH, 12-125
 LINK, 12-127
 LINKLOOP, 12-130
 MATH, 12-134
 MATHLOOP, 12-138
 MESSAGE, 12-143
 MODE, 12-145
 MOVE, 12-149
 MOVELOOP, 12-152
 NOTE, 12-156
 OFF, 12-157
 ON, 12-158
 PRINT, 12-160
 PROCEED, 12-161
 REPORT, 12-163
 SCREEN, 12-165
 SCROLL, 12-167
 SELECT, 12-169
 SERIES, 12-172
 SHOW, 12-175
 SQL, 12-178
 step length, 12-4
 synonyms, 12-2
 TIE, 12-181
 TOP, 12-183
 TRANSACT, 9-23, 12-184
 UPDATE, 12-188

using literals, 12-4
 VALIDATE, 12-189
 VISIT, 12-192
 WINDOW, 12-193
 ZIP, 12-196
 logic command windows, 12-3
 logic constructs
 SQL logic blocks, 9-3
 logic structures, 1-8
 login, MPE/iX, 2-1
 *LTRACE, 4-7

M

macros in logic blocks, 12-31
 macros in module builder, 8-27
 main menu, 2-7
 main menu function key, 6-41
 manual screen field numbering, 6-27,
 6-60
 *MAP option, CLEAR command, 12-24
 marking blocks in screen painter, 6-59
 master title referencing, 3-19
 master titles, 5-38, 12-4
 MATH command, 12-134
 MATHLOOP command, 12-138
 menu
 application generation, 14-2
 database items, 5-48
 description of, 6-1, 6-48
 dictionary, 5-2
 main, 2-7
 record layout, 5-41
 reports, 7-24
 screen development, 6-24
 storage items, 5-22
 utilities, 13-1
 validation items, 5-15
 menu action items, 6-1, 6-51
 menu action prefixes, 6-48
 menu bypass, 2-11, 14-6
 menu item selection, 2-8

Index-12

- menu selection by abbreviation, 2-9
- message catalog, D-17
- MESSAGE command, 12-143
- messages
 - clearing the message window, 5-78
 - construction, 5-77
 - data validation, 6-15
 - definition screen, 5-75
 - display, 5-78
 - during screen input, 6-15
 - from background process, 12-6
 - generation, 5-78, 14-1
 - help screen, 5-77
 - query, 12-144
 - response item, 5-76
 - severity codes, 5-75
 - type codes, 5-75, 12-143
- MODE command, 12-71, 12-76, 12-83, 12-89, 12-107, 12-145
- MODE *UNLOCK command, 12-77, 12-147
- MODE *WRITE command, 12-77
- module
 - capabilities, 8-2
 - construction, 8-19
 - data screen layout, 8-21
 - file maintenance, 8-7
 - item names, 8-25
 - keyed access, 8-6
 - macros, 8-27
 - scanning, 8-4
 - templates, 8-24, 8-26, 8-30
- module file, database, 14-5
- more keys function keys, 2-14
- *MOREREC, 4-14, 12-82
- MOVE command, 12-149
- MOVELOOP command, 12-152
- moving fields on reports, 7-54
- moving screen items, 6-58
- MPE/iX
 - command interpreter, 6-42
 - error messages, C-1
 - file names, 5-55
 - login, 2-1
- MPE/iX variables, D-13
 - HP4APPNPATh, 2-26, D-13, D-19
 - HP4BG, D-14
 - HP4DATAPATH, 2-26, 5-55, D-13, D-19
 - HP4DBMPATH, 2-26, 9-21, 14-5, D-13, D-19
 - HP4DBMSIZE, 9-21, D-14
 - HP4FSPATH, 2-26, 5-63, B-4, D-13, D-19
 - HP4INV_PRT_CHAR, D-14
 - HP4SCREEN_PRINT, 6-41, D-14
 - HP4SPATH, 2-26, 13-20, D-13, D-16, D-19
 - HP4SQLPATH, 2-26, D-14, D-19
 - HP4TERM, 2-26, D-14, D-19
 - HP4TIPATH, 2-26, D-14, D-19
- *MSGDELAY, 4-8
- multiple databases, 10-2
- multiple HP ALLBASE/4GL systems, D-18
- multiple record layouts, 5-57, 12-72, 12-99
- multiple selection criteria, 7-37

N

- name recall, 2-13
 - name completion function, 2-19
 - option for activating, 2-26
 - search mode, 2-19
 - tab key, 2-21
- name substitution, 8-28, 8-29
- naming rules, 3-2, 12-1
- Native Language Support, A-1
- N-computer, A-1
- new commit, 6-15
- newline character, 12-82
- *NEWTIE, 4-8, 6-8, 6-16

NLDATALANG, A-1
 NLDATALANG job control word, 2-26,
 D-14, D-19
 NLS collating sequence, A-1
 NLUSERLANG job control word, 2-26,
 D-14, D-17, D-19
 no echo, 12-64
 *NOLOCK argument, 12-76
 non-critical linkages, 7-7, 7-44, 7-46
 non-generated components, 14-3
 no-op command, see ZIP, 12-196
 NOTE command, 12-156
 *NULL, 5-8
 null indicator variables, 5-8
 null message, 5-78
 null values, 5-8, 12-55
 numeric constant referencing, 3-18
 numeric constants screen, 5-31
 numeric fields on reports, 7-53
 numeric overflow on reports, 7-53

O

object types, 13-16
 OFF command, 12-157
 ON command, 12-158
 on-line help, 2-22
 OPEN CURSOR command, 9-7, 9-9
 opening a cursor, 12-89
 opening data sets, 12-98
 opening files, 12-71
 opening HP TurboIMAGE/iX databases,
 10-7
 opening HP TurboIMAGE/iX data sets,
 10-8
 operand treated as absolute value, 12-8,
 12-134, 12-139
 operating system file names, 5-55
 operating system interface, 1-2
 Op System function key, 6-42
 optional error commands, 12-114, 12-120

other data movement fields, 6-4, 6-6,
 6-16, 6-32
 output fields on reports, 7-52
 output fields on screens, 6-52
 output file, report, F-8
 overlining numbers on reports, 7-40,
 7-56

P

pad character, 5-7, 5-25
 page details for reports, 7-28
 *PAGELINE, 4-8
 *PAGENO, 4-8
 painter, report, 7-47
 paper type for reports, 7-28
 parameter passing, to external program,
 12-53
 parameters, for external program, 12-54
 partially complete transactions, 12-185
 *PASS, 3-17, 4-8, 5-36, 12-19, 12-169
 passing parameters to an external
 program, 12-54
 password
 MPE/iX, 2-1
 user, 2-3
 perform on SHOW, 6-34
 permitted SQL commands, 9-4
 pointer, file position, 12-73
 portability of applications, F-1
 prefixes
 function key actions, 6-37
 menu actions, 6-48
 system items, 6-51
 *PREVFLD, 4-11, 6-8
 previous keys function key, 6-41
 previous menu function key, 6-41
 primary data movement field, 6-4, 6-5,
 6-12, 6-16, 6-32, 12-176
 primary report file, 7-1, 7-27
 PRINT command, 7-6, 12-160
 printers, 7-28

Index-14

- printing problems, D-25
- printing screens, 2-23, 6-41, D-21
- prior functions, 6-7
- PROCEED command, 6-7, 12-161
- *PROCESS, 4-9
- processes, 1-9
- processing logic
 - data screens, 6-3
 - reports, 7-15
- program files, 1-16, D-4
- protecting data files during testing, 14-7

Q

- qualifier, in field references, 3-8
- query message response item, 5-76
- query messages, 12-144
- quotes in literals, 3-18

R

- range validation, 5-19
- reading link files, in reports, 7-43
- reading serial files, 12-82
- read-only data sets, 12-98
- read-only files, 12-71
- *RECNO, 4-9, 12-83
- record buffer
 - data sets, 12-99
 - files, 12-72
 - select list, 12-91
 - serial files, 12-85
 - SQL table, 12-91
- record layout
 - details screen, 5-43
 - header screen, 5-46
 - menu, 5-41
 - select list, 5-72
- record locking, 12-77
- record not found errors, 12-75
- record not read errors, 12-75
- record selection in reports, 7-2, 7-16
- record sorting in reports, 7-3, 7-16

- reference by name in TIE command, 6-17
- reference by name, screen fields, 6-31
- referencing by name, 12-1
- referencing rules
 - calculated items, 3-17
 - communication area fields, 3-19
 - constants, 3-18
 - field, 3-5
 - file record buffers, 3-12
 - file record fields, 3-11
 - HP ALLBASE/SQL names, 3-3
 - HP TurboIMAGE/iX names, 3-4
 - in select lists, 5-72
 - item names, 3-2, 3-5
 - literals, 3-18
 - master titles, 3-19
 - naming rules, 12-1
 - numeric constants, 3-18
 - scratch-pad fields, 3-16
 - screen fields, 3-13
 - subscripts, 3-9
 - substrings, 3-7
 - titles, 3-19
 - variables, 3-19
 - work area buffers, 3-16
 - work area fields, 3-15
- *REFRESH, 6-21
- refreshing the screen, 6-42
- *REFRESH option, EXTERNAL command, 12-56
- REFRESH option with the CALL command, 12-17
- relational operators, 12-113, 12-119
- RELEASE command, 9-6
- releasing from a database, 14-6
- remake utility, D-16
- renumbering screen fields, 6-60
- reorder utility, 13-21, D-16
- *REPORT, 4-9, 7-8
- report

- calculations, 7-11
- column heading line, 7-16
- communication area fields, 7-8
- completion, 7-15
- completion logic, 7-18
- control breaks, 7-4
- control logic, 7-15
- definition, 1-10
- displayed on terminal, 7-28
- field overflow, 7-53
- fields suppressed if unchanged, 7-56
- file designator, 7-2, 7-26
- file equations, 7-27
- file linkages, 7-6, 7-12
- file, primary, 7-1
- functions, 7-8
- generation, 7-15
- header screen, 7-26
- HP ALLBASE/SQL databases, 7-12
- HP TurboIMAGE/iX databases, 7-13
- initialization, 7-15
- line group identifiers, 7-5
- line group logic, 7-15, 7-19
- line groups, 7-4, 7-38
- line header screen, 7-38
- line numbers, 7-39
- line printing, 7-40
- line types, 7-4
- menu, 7-24
- output, 7-2, 7-26
- output fields, 7-52
- output file, F-8
- page details, 7-28
- page heading lines, 7-16
- painter, 7-47
- paper type, 7-28
- primary file, 7-27
- printers, 7-28
- print line buffer, 7-40
- processing logic, 7-15
- record processing, 7-15, 7-17
- record selection, 7-2, 7-16
- record sorting, 7-16
- saving output on disk, 7-27
- selection criteria screen, 7-34
- sorting, 7-3
- sorting fields, 7-31
- sorting levels, 7-31
- sorting screen, 7-31
- stationery, 7-28, 7-30
- subheading lines, 7-16
- switches, 7-8
- totalling facilities, 7-9
- totalling matrix, 7-9
- REPORT command, 7-16, 12-163
- report painter
 - cursor control, 7-48
 - display scrolling, 7-49
 - field descriptions, 7-55
 - function keys, 7-49
 - moving copying or deleting fields, 7-54
 - prompt lines, 7-48
 - saving reports, 7-54
 - window, 7-47
 - window keys function keys, 7-49
- resetting the terminal, 6-42
- response item, messages, 5-76
- restore original function key, 6-51
- retrieving HP ALLBASE/SQL data, 9-8
- returning parameters from an external program, 12-55
- ROLLBACK WORK command, 12-184, 12-186
- roman8 character set, 2-6
- *ROUTINE, 4-9
- *ROWCOUNT, 4-9
- running HP4TUPLD, 10-3
- run-only applications, D-19
- run-time code, 14-1
- run-time data file reformat, 5-67, B-4

Index-16

run-time environment, 9-7
run-time error messages, 14-5

S

SAVEPOINT command, 12-186

savepoints, 12-186

saving reports, 7-54

saving screens, 6-62

scratch-pad field referencing, 3-16, 12-1

scratch-pad fields, 1-6, 3-2

scratch-pad fields screen, 5-35

*SCREEN, 4-10, 6-8

screen

- alphanumeric constants, 5-33

- application generation menu, 14-2

- application titles, 5-37

- calculated items, 5-27

- catalog copying, 13-7

- catalog deletions, 13-9

- catalog display, 13-11

- catalog printing, 13-4

- database items menu, 5-48

- data file creation, 5-61

- data file reformat, 5-64

- Data Manager Specific Field

 - Attributes, 5-10

- dictionary menu, 5-2

- field specifications, 5-4

- file linkages, 7-42

- file/SQL table deletion, 5-68

- function keys, 6-36

- help screens, 5-79

- main menu, 2-7

- messages, 5-75

- module builder, 8-11

- module details, 8-15

- numeric constants, 5-31

- record layout details, 5-43

- record layout header, 5-46

- record layout menu, 5-41

- report header, 7-26

- report line header, 7-38

- report selection criteria, 7-34

- reports menu, 7-24

- scratch-pad fields, 5-35

- screen development menu, 6-24

- screen field details, 6-30

- screen header, 6-26

- select list details, 5-70

- select list header, 5-73

- sign on, 2-3

- storage items menu, 5-22

- utilities menu, 13-1

- validation items menu, 5-15

- validation ranges, 5-17

- validation tables, 5-20

- variables, 5-24

- work areas, 5-39

screen and field commit actions, 6-11

screen buffer, 6-12, 12-176

SCREEN command, 6-4, 6-7, 12-165

screen commit action, 6-11

screen, data, 6-2

screen development, 1-8

screen display highlighting, 6-49, 6-56

screen display, scrolling data, 12-167

screen field and item types, 6-48

screen field referencing

- by name, 3-13, 6-31, 12-1

- by number, 3-14

- fully qualified, 3-13

- in SQL logic blocks, 9-5

screen fields

- auto flow, F-5

- behaviour, F-5

- buffers, 6-4, 6-34, 12-24

- clearing, 6-33, 6-39

- converting types, 6-53

- data acceptance, 6-3

- defined with dictionary field, 6-53

- detail screen, 6-3

- functions, 6-3, 6-7, 6-34

- justification, F-5
- name qualifier, 6-33
- numbering, 6-27
- numbering on windows, 12-193
- processing sequence, 6-3
- referencing, 3-13, 6-31
- screen generation, 6-23
- screen header, 6-3
- screen image printing, 2-23, D-21
- screen information display, 6-3
- screen input and output fields, 6-52
- screen items, deletions, 6-59
- screen line number, 12-34
- screen literals, 6-49
- screen painter, 6-3
 - cursor control, 6-48
 - function keys, 6-47, 6-58
 - layout keys, 6-47
 - main keys, 6-47
 - more keys, 6-47
 - more keys function keys, 6-61
 - prompt lines, 6-46
 - saving screens, 6-62
 - sequence keys, 6-47
- screen printing function keys, 6-41
- screen printing, to slave printer, 6-42
- screen processing
 - field display logic, 6-12
 - input field logic, 6-14
 - levels, 6-9
 - logic, 6-10
 - SHOW command, 6-19
- screen refresh, 6-42
- screen scroll area, displaying data, 12-33
- screen system items, 6-49
- screen types, 1-8, 6-1, 6-48
- scroll area, 6-2, 6-28, 6-29, 12-34, 12-167
- scroll areas on screens, 6-28
- SCROLL command, 12-167
- scrolling direction, 12-34, 12-167
- search mode, 2-19
- secondary files, 8-9
- sections, stored in database, 9-4
- secured components in versions, B-2
- security code, 1-14
- SELECT command, 9-4, 9-5, 9-8, 12-89, 12-169
- SELECT command in reports, 7-12
- SELECT command limitations, 9-6
- selecting menu items, 2-8
- selection criteria, 7-37, 7-44
- selection criteria, for linkages, 7-7
- selection criteria for linkages, 7-37
- selection criteria number, 7-35
- selection criteria screen, 7-34
- select lists
 - data references, 5-72
 - definition of, 1-11
 - details screen, 5-70
 - entries, 5-70
 - generation, 5-72
 - header screen, 5-73
 - record buffer, 12-91
 - record layout, 5-72
- sequence numbers keys, 6-60
- serial file error messages, C-3
- serial files
 - creation, 5-59, 12-83
 - dynamic naming, 4-4, 5-55
 - operations, 12-85
 - reading, 12-82
 - record buffers, 12-85
 - temporary, 5-56
 - writing, 12-82
- SERIES command, 12-172
- severity codes, messages, 5-75
- S-file inconsistencies, 13-22
- S-file indexes, D-2
- S-files, 1-15, 13-20, D-2
 - “clean”, D-3
 - copying, D-3
 - deleted records, D-15

Index-18

- size, D-15
- S-files, repairing, 13-24
- S-file utilities, 13-13
- SHELL internal routine, 6-42
- short descriptions, 13-4
- SHOW command, 6-7, 6-19, 12-175
 - data movement, 6-21
 - functions, 6-7, 6-21, 6-22
 - *REFRESH option, 6-21, 12-176
- *SHOWING, 4-14, 6-7, 6-9, 6-21, 6-34
- SHOW *REFRESH command, 12-176
- signing off, 2-24
- signing on, 2-2
- sign on screen, 2-3
- sign on screen bypass, 2-27
- sign-on screen function key, 6-41
- slave printers, D-25
 - printing screens, 6-42
 - reports, 7-28
- sort fields, 7-3
- sorting fields in reports, 7-31
- sorting in reports, 7-3
- sorting sequence, A-1
- sort levels in reports, 7-3, 7-9, 7-31
- sort sequence, 7-32
- special text, 6-53
- special text function keys, 6-53
- special text items, 6-49
- SQL
 - access class, 5-52
 - BEGIN WORK command, 9-7
 - column names, 5-53
 - commands, 9-4
 - COMMIT WORK command, 9-7, 12-92
 - component generation, 14-5
 - CONNECT command, 9-6
 - cursor commands, 9-7
 - cursors, declaring and opening, 12-89
 - database environment, 5-52
 - data types, 5-8
 - DBEFileset, 5-52
 - DECLARE CURSOR command, 9-6
 - host variable references, 3-20
 - INSERT command, 9-11
 - logic block generation, 9-21
 - logic blocks, 1-10, 1-12, 9-3
 - logic blocks in reports, 7-12
 - logic command (HP ALLBASE/4GL), 12-178
 - names, 3-3
 - RELEASE command, 9-6
 - SELECT command, 9-4, 9-5
 - select list operations, 12-92
 - SQLERRD communication area, 4-9
 - table definition window, 5-51
 - table names, 5-52
 - table operations, 12-92
 - table ownership, 5-52
 - table record buffer, 12-91
 - table reformat, 5-66
- SQL commands
 - BEGIN WORK, 9-7
 - CLOSE CURSOR, 9-7, 12-92
 - COMMIT WORK, 12-92
 - CONNECT, 9-6
 - DECLARE CURSOR, 9-6
 - DECLARE CURSOR for SELECT, 9-9
 - DELETE, 9-16
 - DELETE WHERE CURRENT, 9-18
 - FETCH, 9-9, 12-93
 - generation, 9-6
 - INSERT, 9-11
 - issued implicitly, 9-6
 - LOCK TABLE, 9-24
 - OPEN CURSOR, 9-7, 9-9
 - RELEASE, 9-6
 - ROLLBACK WORK, 12-184, 12-186
 - SAVEPOINT, 12-186
 - SELECT, 9-8
 - UPDATE, 9-12

- SQLEXPLAIN command, C-8
- start again function key, 6-39
- start-of-report function, 7-16
- start of report function, 7-9, 7-29
- startup command file, D-19
- stationery for reports, 7-28, 7-30
- step factor
 - IFLOOP command, 12-118
 - LINKLOOP command, 12-131
 - MATHLOOP command, 12-140
 - MOVELOOP command, 12-153
- storage items, 1-6
- storage items menu, 5-22
- storage type, 5-5
- stored database sections, 9-4, 14-5
- structure file, for data files, F-3
- subheading line groups, 7-4, 7-32
- subscript references, 3-9
- substring references, 3-7
- subtotal line groups, 7-32
- *SUITE, 4-12
- supported terminals, D-23
- suppressed data set error messages, 12-105
- suppressed file error messages, 12-74
- suppressed zero fields on reports, 7-53
- suppressing fields on reports, 7-56
- suppressing report lines, 7-40
- switches, 4-12, 6-8
 - *BYPASS, 4-13
 - *ENDFIELD, 4-13, 6-8, 12-52
 - *ENDLINE, 4-13
 - *ENTERED, 4-13, 6-7, 6-8
 - fixed function, 4-13
 - function keys, 6-37
 - in report generation, 7-8
 - in screen processing, 6-7
 - *MOREREC, 4-14, 12-82
 - *SHOWING, 4-14, 6-7, 6-9
 - user, 4-12
- synonyms for logic commands, 12-2

- synonyms, system items on screens, 6-51
- syntax for the CALL command, 12-18
- system architecture, 1-2
- system files, 1-15
- system files, repairing, 13-24
- system file utilities, 13-13, 13-20, 13-21
- system item prefixes, 6-51
- system items on screens, 6-2, 6-49, 6-50
- system keys function keys, 2-12

T

- tab key
 - used for name recall, 2-21
- Tab key, 2-9
- table
 - indexes, 5-44
- table locking mode, 9-23
- table names, external, 5-52
- table ownership, 5-52
- TABSET file, D-24
- tabstop settings, D-24
- tabulations on screens, 6-60
- templates
 - constants, 8-34
 - function key sets, 8-30
 - functions, 8-31
 - help screens, 8-40
 - items not supported, 8-40
 - libraries, 8-26
 - messages, 8-37
 - naming, 8-24
 - processes, 8-31
 - SQL logic blocks, 8-33
 - type, 8-24
 - variables, 8-34
- temporary files, 5-56, 12-83
- temporary serial files, creation, 5-59
- termdata files, D-23
- terminal
 - configuration, D-25

Index-20

- configurations with external command, 12-56
- reports on terminal screen, 7-28
- reset, 6-42
- slave printer, D-25
- supported, D-23
- support files, D-23
- tabstop settings, D-24
- touchscreen, 2-6, D-24
- terminal setting with the CALL command, 12-12
- terminating HP ALLBASE/4GL, 6-40
- terminating screens, 2-18, 6-18
- terminology
 - HP TurboIMAGE/iX, 1-13
- testing applications, 14-6
- text literals, 6-50
- TF line groups, 7-9, 7-18
- TIE command, 6-9, 6-16, 6-17, 12-181
- *TIME, 4-12
- title referencing, 3-19, 5-38
- titles, 6-2, 12-4
- T line groups, 7-10, 7-18
- to do list, 8-19
- TOP command, 12-183
- top of page heading line group, 7-21
- total line groups, 7-4
- totalling facilities in reports, 7-9
- totalling matrix in reports, 7-9
- *TOTALS, 4-10, 7-8, 7-10, 7-56
- touchscreen operation, D-24
- touchscreen terminals, 2-6, 6-48
- trace log file, 14-8
- trace mode, 6-42, 14-7
- training mode, 6-42, 14-7
- TRANSACTION command, 9-7, 9-23, 12-184
- transaction blocks, 12-184
- transaction management, 9-22
- transaction rollback, 12-185, F-8
- type codes, messages, 5-75, 12-143

U

- undefined items, error messages, 14-5
- underlining numbers on reports, 7-40, 7-56
- unique indexes, 5-45
- unloading applications, 14-8
- unloading versions, B-4
- *UNLOCK, 12-71
- UPDATE command, 12-188
- UPDATE command (SQL), 9-12
- updating HP ALLBASE/SQL data
 - via a cursor, 9-12
 - without a cursor, 9-15
- uploading, 10-3
- uploading HP TurboIMAGE\XL
 - dictionary definitions, 10-4
- upload utility, 10-2
- *USER, 4-12
- user cancelled transactions, 12-185
- user input in reports, 7-16
- user name, developer, 2-3
- user password, 2-3
- user switches, 4-12
- utilities
 - catalog copying, 13-7
 - catalog deletions, 13-9
 - catalog display, 13-11
 - catalog printing, 13-4
 - remake, D-16
 - reorder, D-16
- utilities menu, 13-1
- utility programs, 13-13

V

- VALIDATE command, 12-189
- validation by secondary file, 8-9
- validation error message, 5-18, 5-20
- validation items, 1-6
- validation items menu, 5-15
- validation of data, 5-7
- validation range, 6-15, 7-2

- validation range checking, 5-19
- validation ranges screen, 5-17
- validation table, 7-2, 12-19
- validation table checking, 5-21
- validation tables screen, 5-20
- variable referencing, 3-19
- variables
 - for multiple HP ALLBASE/4GL systems, D-18
 - HP4APPNPATH, 2-26, D-13, D-19
 - HP4BG, D-14
 - HP4DATAPATH, 2-26, D-13, D-19
 - HP4DBMPATH, 2-26, 9-21, 14-5, D-13, D-19
 - HP4DBMSIZE, 9-21, D-14
 - HP4FSPATH, 2-26, B-4, D-13, D-19
 - HP4INV_PRT_CHAR, D-14
 - HP4SCREEN_PRINT, 6-41, D-14, D-22
 - HP4SPATH, 2-26, 13-20, D-13, D-16, D-19
 - HP4SQLPATH, 2-26, D-14, D-19
 - HP4TERM, 2-5, 2-26, D-14, D-19, D-23
 - HP4TIPATH, 2-26, 10-2, D-14
 - with the CALL command, 12-17
- variables screen, 5-24
- *VERSION, 4-12
- version data files, B-4

- versions, B-1
- vertically aligned prompts, 6-60
- VISIT command, 12-173, 12-192

W

- WHERE clause in SQL, 9-15
- WINDOW command, 12-193
- window field numbering, 12-193
- window function keys, 12-193
- window help screen, 12-194
- windows, 6-1, 6-2, 6-29, 6-49
- windows, logic commands, 12-3
- windows, on screens, 6-29
- work area
 - buffer, 5-40
 - buffer referencing, 3-16
 - clearing fields, 5-40
 - definition screen, 5-39
 - field references, 3-15
 - structure, 5-40
- *WRITE, 12-71, 12-98
- write protection for data files, 14-7
- writing serial files, 12-82

Z

- zero step factor, 12-119, 12-131, 12-140, 12-153
- ZIP command, 12-196