HP 9000 Computer Systems

ALLBASE/NET User's Guide

HP Part No. 36217-90093
Printed in U.S.A. 1994

Fifth Edition
E0494

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Printing History

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

Edition	Date	Software Version
First Edition	November 1988	92593A.00.00 (Series 800)
First Edition	February 1989	98884-1.00 (Series 300)
Second Edition	September 1989	92593A.07.00 (Series 800)
		98884A Rev 7.0 (Series 300)
Third Edition	January 1991	92593A.E1.00 (Series 800)
		98884A.E1.00 (Series 300)
Fourth Edition	September 1992	36217-02A.F0.00 (Series 700/800)
		79725A.F0.00 (Series 300/400)
Fifth Edition	April 1994	36217-02A.G0.00 (Series 700/800)

ALLBASE/SQL Manuals

Title	Part Number
ALLBASE/ISQL Reference Manual	36217-90188
ALLBASE/NET User's Guide	36217-90093
ALLBASE/SQL Advanced Application Programming Guide	36217-90186
ALLBASE/SQL C Application Programming Guide	36217-90014
ALLBASE/SQL COBOL Application Programming Guide	36217-90058
ALLBASE/SQL Database Administration Guide	36217-90005
ALLBASE/SQL FORTRAN Application Programming Guide	36217-90013
ALLBASE/SQL Message Manual	36217-90009
ALLBASE/SQL Pascal Application Programming Guide	36217-90007
ALLBASE/SQL Performance and Monitoring Guidelines	36217-90185
ALLBASE/SQL Reference Manual	36217-90001
HP ALLBASE/QUERY User's Guide	92534-64001
HP PC API User's Guide for ALLBASE/SQL	36217-90187
Up and Running with ALLBASE/SQL	36389-90011

Additional Resources

You may find the following manuals helpful:

- *HP NewWave Office: Information Access SQL/UX*
- *HP NewWave Office: Using Information Access PC*
- *HP NewWave Access User's Guide*
- *Installing and Administering LAN/9000*
- *Installing and Administering Network Services*
- *Installing and Administering ARPA Services*
- *NetIPC Programmer's Guide*
- *Networking Reference Pages* (the section "LAN/X25 Reference Pages")
- *Networking Overview*

Preface

The *ALLBASE/NET User's Guide* describes ALLBASE/NET, a product that allows an application on one MPE/iX or HP-UX system to access an ALLBASE/SQL DBEnvironment on another MPE/iX system or HP-UX system. This manual is written for the system administrator who is using ALLBASE/NET on an HP-UX system. He or she uses this guide to understand how ALLBASE/NET works and to learn how to use the NETUtil utility to maintain necessary files. In addition, portions of this guide will be of interest to application programmers and to users of ALLBASE/SQL DBEnvironments, particularly chapters 1, 3, and 4.

In some HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of 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). This guide is divided as follows:

- Chapter 1, "Getting Started," provides an overview of ALLBASE/NET and its components.
- Chapter 2, "Setting Up ALLBASE/NET," contains step-by-step instructions for setting up ALLBASE/NET on the client and the server.
- Chapter 3, "Using ALLBASE/SQL Through ALLBASE/NET," describes how to use ALLBASE/NET with various applications. It also discusses the ALLBASE preprocessors.
- Chapter 4, "NETUtil Reference," describes the NETUtil commands and gives examples of how they are used.
- Appendix A is a list of ALLBASE/NET files.
- The Glossary contains definitions of terms used in this manual.

What's New in this Release

The following table highlights the new or changed functionality in this release, and shows you where each feature is documented.

New Features in ALLBASE/SQL Release G.0

Feature (Category)	Description	Documented in . . .
Stored procedures (Usability)	Provides additional stored procedure functionality for application programs. Allows declaration of a procedure cursor and fetching of multiple rows within a procedure to applications. New statement: ADVANCE. Changed syntax: CLOSE, CREATE PROCEDURE, DECLARE CURSOR, DESCRIBE, EXECUTE, EXECUTE PROCEDURE, FETCH, OPEN.	*ALLBASE/SQL Reference Manual*, "SQL Statements" and "Using Procedures" in "Constraints, Procedures and Rules;" *ALLBASE/SQL Advanced Application Programming Guide*, "Using Procedures in Application Programs."
Case insensitivity (Usability)	Adds an optional attribute to the character and varchar type column attributes of tables. Allows search and compare of these columns in a case insensitive manner. Four new SQLCore data types are added. Changed syntax: ALTER TABLE, CREATE TABLE.	*ALLBASE/SQL Reference Manual*, "Comparison Predicate" in "Search Conditions," CREATE TABLE in "SQL Statements."
Support for 1023 columns (Usability)	Increases the maximum number of columns per table or view to 1023. Increases maximum sort columns and parameters in a procedure to 1023.	*ALLBASE/SQL Reference Manual*, CREATE TABLE and CREATE VIEW in "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, "ALLBASE/SQL Limits" appendix.
ISQL HELP improvements (Usability)	Gives help for entire command instead of only the verb.	*ALLBASE/ISQL Reference Manual* , HELP in "ISQL Commands."
EXTRACT command (Usability)	Extracts modules from the database and stores them in a module file. Allows for creation of a module file at any time based on the current DBEnvironment without preprocessing. New command: EXTRACT. Changed syntax: INSTALL.	*ALLBASE/ISQL Reference Manual* , "Using Modules" in "Using ISQL for Database Tasks," EXTRACT, INSTALL in "ISQL Commands."

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
New SQLGEN GENERATE parameters (Usability)	Generates SQL statements necessary to recreate modified access plans for module sections. New syntax for GENERATE: DEFAULTSPACE, MODOPTINFO, PARTITION, PROCOPTINFO, SPACEAUTH.	*ALLBASE/SQL Database Administration Guide*, "SQLGEN Commands" appendix.
Row level locking (Usability)	Permits multiple transactions to read and update a table concurrently because locking is done at row level. Since the transaction will obtain more locks, the benefits must be weighed against the costs. (Previously documented in an addendum after F.0 release.)	*ALLBASE/SQL Reference Manual*, "Concurrency Control through Locks and Isolation Levels;" *ALLBASE/SQL Database Administration Guide*, "Effects of Page and Row Level Locking" in "Physical Design."
Increased number of users (Usability)	Removes the limitation of 240 users supported by pseudotables. (Maximum is system session limits: 2000 on HP-UX; 1700 on MPE/iX.)	*ALLBASE/SQL Database Administration Guide*, "ALLBASE/SQL Limits" appendix.
POSIX support (Usability)	Improves application portability across MPE/iX and HP-UX. Enhances the ALLBASE/SQL preprocessors to run under POSIX (Portable Operating System Interface) on MPE/iX.	*ALLBASE/SQL Advanced Application Programming Guide*, "POSIX Preprocessor Invocation" in "Using the Preprocessor."
Application thread support (Performance, Usability)	Provides the use of threads in an application. Allows ALLBASE/SQL to be used in an application threaded environment on MPE/iX. Application threads are light weight processes that share some resources and last for the duration of a transaction. Threaded applications reduce the overhead of context switching and improve the performance of OpenTP applications.	*ALLBASE/SQL Advanced Application Programming Guide*, "Using the Preprocessor."

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
High Availability	Provides a collection of features to keep systems available nonstop including: Partial STORE and RESTORE, Partial rollforward recovery, DBEFiles in different groups (MPE/iX), detaching and attaching database objects, CHECKPOINT host variable, changing log files, console messages logged to a file, generating fewer log records by using TRUNCATE TABLE to delete rows, and new system catalog information. See the following features for new and changed syntax.	*ALLBASE/SQL Reference Manual*, "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, "Maintaining a Nonstop Production System" in "Maintenance" chapter and "SQLUtil" appendix.
Partial rollforward recovery (High Availability)	Supports partial rollforward recovery through PARTIAL option on SETUPRECOVERY. Used to recover specific DBEFiles while allowing access to other DBEFiles.	*ALLBASE/SQL Database Administration Guide*, "Backup and Recovery" chapter and SETUPRECOVERY PARTIAL in "SQLUtil" appendix.
Partial STORE and RESTORE (High Availability)	Gives more flexibility in backup and recovery strategies by allowing partial store and restore of DBEFiles, DBEFileSets or combinations of both. See "New and changed SQLUtil commands for increased availability" later in this table.	*ALLBASE/SQL Database Administration Guide*, "Backup and Recovery" chapter and "SQLUtil" appendix.
DBEFile group change on MPE/iX (High Availability)	Manages DBEFiles so they can be placed in a particular group or on a particular volume (MPE/iX). Use either CREATE DBEFILE or MOVEFILE.	*ALLBASE/SQL Reference Manual*, CREATE DBEFile in "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, "Maintaining a Nonstop Production System" in "Maintenance" chapter and MOVEFILE in "SQLUtil" appendix.
Detaching and attaching database objects (High Availability)	Detaches or attaches a DBEFile or DBEFileSet from the DBEnvironment. This is useful for data that is accessed infrequently such as tables containing historical data only. New SQLUtil commands: DETACHFILE, ATTACHFILE.	*ALLBASE/SQL Database Administration Guide*, "Maintaining a Nonstop Production System" in "Maintenance" chapter and DETACHFILE, ATTACHFILE in "SQLUtil" appendix.

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
New and changed SQLUtil commands for increased availability (High Availability)	Adds support for high availability and System Management Intrinsics. Intended for non-stop, continuously available operations. New SQLUtil commands: ATTACHFILE, CHANGELOG, DETACHFILE, RESTORE PARTIAL, STORE PARTIAL, STOREINFO, STOREONLINE PARTIAL, WRAPDBE. Modified SQLUtil commands: MOVEFILE, RESTORE, RESTORELOG, SHOWDBE, SETUPRECOVERY, STORE, STORELOG, STOREONLINE.	*ALLBASE/SQL Database Administration Guide*, "SQLUtil" appendix.
List files on backup device (High Availability)	Lists physical names of files stored on backup device with new SQLUtil command: STOREINFO.	*ALLBASE/SQL Database Administration Guide*, "Backup and Recovery" chapter and STOREINFO in "SQLUtil" appendix.
Log file improvements (High Availability)	Allows changing log files, switching of console messages to a file, and gives advance warning for log full. Increased maximum size of a single DBE log file to 4 gigabytes. A DBEnvironment can have up to 34 log files configured. Changed syntax: CHECKPOINT. New SQLUtil command: CHANGELOG.	*ALLBASE/SQL Reference Manual*, CHECKPOINT in "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, "Maintaining a Nonstop Production System" in "Maintenance" chapter, CHANGELOG in "SQLUtil" appendix, and "ALLBASE/SQL Limits" appendix.
New SET SESSION and SET TRANSACTION statements (Standards, Performance)	Provides additional flexibility and improved performance. Allows setting and changing transaction and session attributes.	*ALLBASE/SQL Reference Manual*, SET SESSION and SET TRANSACTION in "SQL Statements."
FIPS flagger (Standards)	Meets Federal Information Processing Standard (FIPS) 127.1 flagger support. Flags non-standard statement or extension. Invoked with a flagger option in the preprocessor command line or the SET FLAGGER command in ISQL. Updatability rules are different when flagger is invoked. New syntax: DECLARE CURSOR, WHENEVER. Changes to C and COBOL host variable declaration.	*ALLBASE/SQL Reference Manual*, DECLARE CURSOR in "SQL Commands" and "Standards Flagging Support" appendix; *ALLBASE/SQL Advanced Application Programming Guide*, "Flagging Non-Standard SQL with the FIPS Flagger;" *ALLBASE/ISQL Reference Manual* , SET in "ISQL Commands."

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
Optimizer enhancement (Performance)	Uses a more efficient algorithm that significantly reduces the time to generate the access plan.	*ALLBASE/SQL Performance and Monitoring Guidelines*, "Optimization" in "Basic Concepts in ALLBASE/SQL Performance."
Access plan modification (Performance)	Allows modification of access plans for stored section to optimize performance. View the plan with SYSTEM.SETOPTINFO. New statement: SETOPT.	*ALLBASE/SQL Reference Manual*, SETOPT in "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, SYSTEM.SETOPINFO in "System Catalog."
Syntax added to disable access plan optimization (Performance, Usability)	Specifies that the optimization information in the module file is not to be used. Changed syntax: EXTRACT, INSTALL, VALIDATE.	*ALLBASE/SQL Reference Manual*, VALIDATE in "SQL Statements; *ALLBASE/ISQL Reference Manual* ," EXTRACT, INSTALL in "ISQL Commands."
Application Development Concurrency (Performance, Usability)	Provides enhancements to improve preprocessing performance when simultaneously accessed by multiple users. Page or row level locking on any system base table and processing without storing sections. See the related features in this table. New SQL parameter: SET DEFAULT DBEFileSet. SQL changed syntax: ALTER TABLE, GRANT, REVOKE, UPDATE STATISTICS. ISQL changed syntax: INSTALL. Changed SYSTEM and CATALOG view. New STOREDSECT tables. Special owners HPRDBSS and STOREDSECT. Changed syntax for Full Preprocessing Mode.	*ALLBASE/SQL Reference Manual*, "Names" and "SQL Statements;" *ALLBASE/SQL Advanced Application Programming Guide*, "Using the Preprocessor;" *ALLBASE/ISQL Reference Manual* , "ISQL Commands;" *ALLBASE/SQL Database Administration Guide*, "Database Creation and Security" and "System Catalog."
System Catalog tables (Performance)	Provides greater concurrency by allowing users to specify table, page, or row level locking of any system table owned by STOREDSECT through the ALTER TABLE statement.	*ALLBASE/SQL Reference Manual*, "Names;" *ALLBASE/SQL Database Administration Guide*, "System Catalog."
Preprocessors (Performance)	Allows optional specification of a DBEFileSet for storage of sections. Allows preprocessing without storing sections in DBEnvironment.	*ALLBASE/SQL Advanced Application Programming Guide*, "Using the Preprocessor."

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
I/O performance improvement (Performance)	Optimizes I/O for initial load, index build, serial scans, internal data restructuring, file activity, pseudo mapped files and temporary files. See the following features for new and changed syntax.	*ALLBASE/SQL Reference Manual*, "SQL Statements."
TRUNCATE TABLE statement (Performance)	Deletes all rows in a specified table leaving its structure intact. Indexes, views, default values, constraints, rules defined on the table, and all authorizations are retained. TRUNCATE TABLE is faster than the DELETE statement and generates fewer logs. New statement: TRUNCATE TABLE.	*ALLBASE/SQL Reference Manual*, TRUNCATE TABLE in "SQL Statements."
New scans (Performance)	Reads tables with a new parallel sequential scan. The tables are partitioned and files are read in a round robin fashion to allow OS prefetch to be more effective. Allows the I/O for a serial scan to spread across multiple disc drives.	*ALLBASE/SQL Performance and Monitoring Guidelines*, "Using Parallel Serial Scans" in "Guidelines on Query Design."
Load performance improvement (Performance)	Improves performance with new SET and SET SESSION attributes, a new binary search algorithm, and deferred allocation of HASH pages. New attributes for SET SESSION statement: FILL, PARALLEL FILL.	*ALLBASE/SQL Reference Manual*, SET SESSION in "SQL Statements."
ISQL enhanced to improve the performance of LOADs (Performance)	Uses new parameters of the ISQL SET command to set load buffer size and message reporting. Improves load performance. Choose a procedure, command file, or new ISQL command to set constraints deferred, lock table exclusively, and set row level DML atomicity. Changed syntax: SET (see the following feature).	*ALLBASE/ISQL Reference Manual* , SET in "ISQL Commands."

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
Modified SET options (Performance)	Provides better performance for LOADs and UNLOADs. Specify buffer size, status reporting for LOAD/UNLOAD or exclusive lock for data table. AUTOSAVE row limit increased to 2147483647. New and changed SET options: LOAD_BUFFER, LOAD_ECHO, AUTOLOCK, AUTOSAVE.	*ALLBASE/ISQL Reference Manual* , SET in "ISQL Commands;" *ALLBASE/SQL Performance and Monitoring Guidelines*, "Initial Table Loads" in "Guidelines on Logical and Physical Design."
SQLMON (Tools)	Monitors the activity of ALLBASE/SQL DBEnvironment. Provides information on file capacity, locking, I/O, logging, tables, and indexes. Summarizes activity for entire DBEnvironment or focuses on individual sessions, programs, or database components. Provides read-only information.	*ALLBASE/SQL Performance and Monitoring Guidelines*, chapters 6-9.
Audit (Tools)	Provides a series of features to set up an audit DBEnvironment which generates audit log records that you can analyze with the new SQLAudit utility for security or administration. Includes the ability to set up partitions. See *ALLBASE/SQL Database Administration Guide* for SQLAudit commands. Modified statements: ALTER TABLE, CREATE TABLE, START DBE NEW, START DBE NEWLOG. New statements: CREATE PARTITION, DROP PARTITION, DISABLE AUDIT LOGGING, ENABLE AUDIT LOGGING, LOG COMMENT.	*ALLBASE/SQL Reference Manual*, "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, "DBEnvironment Configuration and Security" chapter and "SQLAudit" appendix.
Wrapper DBEnvironments (Tools)	Creates a DBEnvironment to wrap around the log files orphaned after a hard crash of DBEnvironment. New SQLUtil command: WRAPDBE.	*ALLBASE/SQL Reference Manual*, "Wrapper DBEnvironments" in "Using ALLBASE/SQL;" *ALLBASE/SQL Database Administration Guide*, WRAPDBE in "SQLUtil."
HP PC API is now bundled with ALLBASE/SQL.	PC API is an application programming interface that allows tools written with either the GUPTA or the ODBC interface to access ALLBASE/SQL and IMAGE/SQL from a PC.	*HP PC API User's Guide for ALLBASE/SQL.*

New Features in ALLBASE/SQL Release G.0 (continued)

Feature (Category)	Description	Documented in . . .
Increased memory for MPE/iX (HP-UX shared memory allocation is unchanged) (Performance)	Increases memory up to 50,000 data buffer pages and 2,000 run time control block pages. Increases the limits significantly allowing allocation of enough data buffer pages to keep the entire DBEnvironment in memory if desired for performance.	*ALLBASE/SQL Reference Manual*, STARTDBE, STARTDBE NEW, and START DBE NEWLOG in "SQL Statements;" *ALLBASE/SQL Database Administration Guide*, "ALLBASE/SQL Limits" appendix.
ALLBASE/NET enhancements (Connectivity, Performance)	Improves performance of ALLBASE/NET, allows more client connections on server system, and reduces number of programs on MPE/iX.	*ALLBASE/NET User's Guide*, "Setting up ALLBASE/NET."
ALLBASE/NET commands and options for MPE/iX (Connectivity, Usability)	Adds option ARPA. Adds option NUMSERVERS to check status of listeners and number of network connections. Changed syntax: ANSTART, ANSTAT, ANSTOP. Changed NETUtil commands: ADD ALIAS, CHANGE ALIAS.	*ALLBASE/NET User's Guide*, "Setting up ALLBASE/NET" and "NETUtil Reference."
ALLBASE/NET and NetWare (Connectivity)	ALLBASE/NET listener for NetWare now works with the 3.11 version of Novell's NetWare for UNIX (HP NetWare/iX).	*ALLBASE/NET User's Guide*, "Setting up ALLBASE/NET."
Changed restrictions for executing NETUtil commands for MPE/iX (Connectivity, Usability)	Adds SM or AM (in the specified account) to MANAGER.SYS for adding, changing, or deleting users for MPE/iX.	*ALLBASE/NET User's Guide*, "Setting up ALLBASE/NET."
ARPA is only TCP/IP interface for data communication through ALLBASE/NET beginning with HP-UX 10.0 (Connectivity)	Remote database access applications that specify NS will not work if the client and/or server machine is an HP 9000 Series 700/800 running HP-UX 10.0 or greater. Server Node Name entry must be changed from NS node name to ARPA host name. For the NETUsers file, the "Client Node Name" must be changed from the NS node name to the ARPA host name. New NETUtil commands: MIGRATE USER, MIGRATE ALIAS.	*ALLBASE/NET User's Guide*, "Setting up ALLBASE/NET" and "NETUtil Reference."

Conventions

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

```
COMMAND
```

can be entered as any of the following:

```
command        Command        COMMAND
```

It cannot, however, be entered as:

```
comm           com_mand       comamnd
```

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

`COMMAND` *filename*

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

(*filename*):(*filename*)

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

```
Do you want to continue? >>  yes
```

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

`COMMAND` { `ON` / `OFF` }

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

`COMMAND` *filename* `[OPTION]`

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can select `OPTION` or *parameter* or neither. The elements cannot be repeated.

`COMMAND` *filename* [`OPTION` / *parameter*]

`[ ... ]`

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

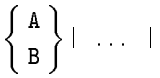
`[,`*parameter*`][...]`

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

`[`*parameter*`][,...]`

`| ... |`

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



`...`

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

`Δ`

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

`(`*parameter*`)Δ(`*parameter*`)`

The symbol indicates a key on the keyboard. For example, RETURN represents the carriage return key or Shift represents the shift key.

CTRL *character*

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

Contents

Figures

Tables

1

Getting Started

ALLBASE/NET is a database access product that provides access to remote data as though it were stored locally. It allows the user of an application on one HP-UX or MPE/iX system to query and update a database in an ALLBASE/SQL DBEnvironment on another HP-UX or MPE/iX system. The remote DBEnvironment must reside on either a HP 9000 Series 700/800, or HP 3000 Series 900. ALLBASE/SQL on the HP 9000 must be Version A.03.00, or later. ALLBASE/SQL on the HP 3000 must be Version A.12.0, or later.

The following applications can be used with ALLBASE/NET to access a remote ALLBASE/SQL DBEnvironment:

- ALLBASE/ISQL and ALLBASE Preprocessors

  HP 9000 Series 300/400, Version A.02.00 or later (client only)
  HP 9000 Series 700/800, Version A.03.00 or later
  HP 3000 Series 900, Version A.12.00 or later

- User-written application programs developed with the following releases of ALLBASE/SQL:

  HP 9000 Series 300/400, Version A.02.00 or later (client only)
  HP 9000 Series 700/800, Version A.03.00 or later
  HP 3000 Series 900, Version A.12.00 or later

- ALLBASE/Query

  Version B.01 or HP Visor, Version A.01.03 or later

- ALLBASE/4GL

  HP 9000 Series 300/400 (client only) and 700/800, Version B.01.00 or later
  HP 3000 Series 900, Version B.00.00 or later

- HP Information Access

  Version A.05.01 or later

- HP PC API, Version A.01.00 or later

Note To determine the version of ALLBASE/SQL and ALLBASE/NET you are using, type the `sqlver` command at the system prompt.

ALLBASE/NET gives users the same SQL functionality provided by the equivalent version of ALLBASE/SQL, with the following exceptions:

- DBEnvironments cannot be created remotely.
- New log files cannot be created remotely.
- Long column data types cannot be used remotely.
- The ALLBASE/SQL database administration utilities SQLUtil, SQLGEN, and SQLMigrate cannot access a remote DBEnvironment.

How ALLBASE/NET Works

ALLBASE/NET uses a client/server architecture. The application runs on the client node and the ALLBASE/SQL DBEnvironment resides on the server node. The client and server are connected by a LAN (Local Area Network) or a WAN (Wide Area Network).

A server node for one application can also be a client node for another application that accesses another DBEnvironment. In the illustration below, Node B is the server node for the application on Node A, and it is also a client node when accessing Node C.

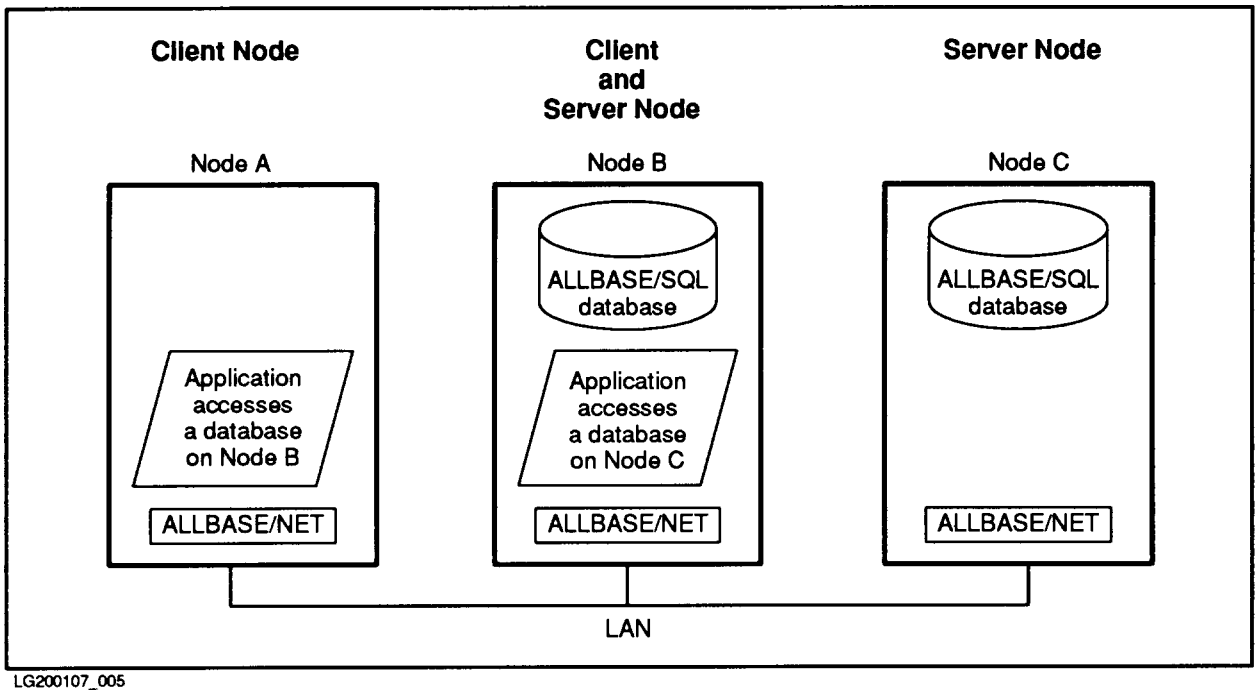


Figure 1-1. Node as Client and Server

ALLBASE/NET Configurations

The client and server systems can be either HP 9000 Series 700s/800s running HP-UX, or HP 3000 Series 900s running MPE/iX. An HP 9000 Series 300/400 can be used as a client. For full client/server functionality on both HP-UX and MPE/iX platforms, ALLBASE/SQL must be installed on both the client and server systems. This configuration allows the user to create and maintain DBEnvironments and develop programs on the client as well as to access DBEnvironments on the server.

Note The ALLBASE/SQL server version must be as new or newer than the ALLBASE/SQL client version.

Hardware and Software Requirements

ALLBASE/NET clients and servers must be connected by a LAN or a WAN (via X.25). The LAN or WAN provides all the necessary hardware and link software to allow communication between machines.

NS (Network Services) and **ARPA services** (Advanced Research Projects Agency) are the software interfaces that enable communication between machines on the same network using the TCP/IP protocol.

Note The HP-UX 10.0 release does not support NS.

Before you install ALLBASE/SQL on your HP 9000 Series 700/800, HP-UX Release 8.0 or later must be installed, along with one of the following:

- LAN/9000 Series 800
- X.25/9000 Link for the Series 800 Network Services (700/800)
- ARPA/Berkeley Services for the HP 9000 Series 700/800

Note If you are also connecting to an HP 3000 Series 900 system, refer to "Hardware and Software Requirements" in the "Getting Started" chapter of the *900 Series HP 3000 Computer Systems ALLBASE/NET User's Guide.*

The NETUtil Utility

ALLBASE/NET provides the **NETUtil** utility for creating and maintaining the AliasDB and NETUsers files that are discussed below. For complete descriptions of all the NETUtil commands, their syntax, and how they are used, refer to the "NETUtil Reference" chapter.

The AliasDB and NETUsers Files

For a user to connect to a remote DBEnvironment, two files must exist: the AliasDB file and the NETUsers file.

The **AliasDB file** resides on the client. The names of the server's DBEnvironments (the remote DBEs) are contained in this file. The AliasDB file has an **alias profile** for each alias name that corresponds to a DBEnvironment. The alias profile contains the following information:

- **Alias name** that corresponds to a DBEnvironment on the server
- Type of database management system being used on the server (ALLBASE/SQL)
- Absolute pathname of the DBEnvironment on the server
- Node name of the server where the DBEnvironment is located
- Machine type of the server
- Type of data communication interface used

The **NETUsers file** resides on the server. It contains a **user profile** for each user on the client that has access to a DBEnvironment on the server. The user profile contains the following information:

- Client node name
- Client login name
- Server login name for the client

You use the NETUtil utility on both the client and the server to add profiles to and maintain the AliasDB and NETUsers files. Additionally, users of DBEnvironments can use NETUtil to display the contents of the AliasDB file so that alias names of DBEnvironments can be checked or confirmed.

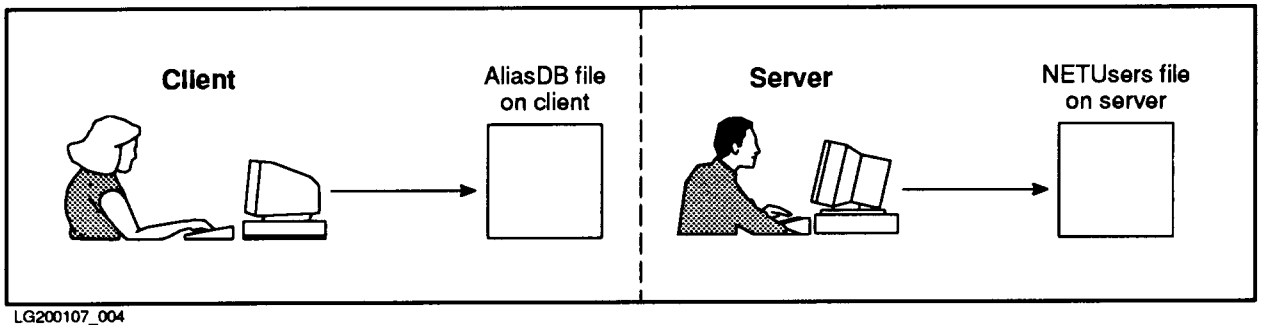


Figure 1-2. System Administrators Using NETUtil

Creating the AliasDB and NETUsers Files

After ALLBASE/SQL has been installed on both the client and the server, the AliasDB and NETUsers files must be created using NETUtil. Adding the first profile to the AliasDB or NETUsers file automatically creates the respective file. These profiles allow specified users on the client to access particular DBEnvironments on the server.

As system administrator on the client, you invoke NETUtil and use the ADD ALIAS command to add an alias profile to the AliasDB file. ADD ALIAS prompts you for details about the specific DBEnvironment to be accessed. Use ADD ALIAS to add more alias profiles to the AliasDB file as needed.

As system administrator on the server, you invoke NETUtil and use the ADD USER command to add a user profile to the NETUsers file. ADD USER prompts you for details about the specific user who will be accessing the remote DBEnvironment. Use ADD USER to add more user profiles to the NETUsers file as needed.

For step-by-step instructions to create the AliasDB and NETUsers files, refer to the "Setting Up ALLBASE/NET" chapter.

The Listener Daemon

The listener daemon on the server provides three functions:

- It listens for connection requests.
- It validates the server login name.
- It sets up a direct communication line between the client and server.

The listener daemon monitors a specified datacomm port for any incoming connection requests. When it receives a request, it validates the server login name for the user on the client and then generates a database process for that user. The listener daemon redirects communications so that the user and the database process communicate directly with each other over the network. It then resumes listening for more connection requests.

You manually start the listener daemon on the server after ALLBASE/SQL has been installed.

You can shut down the listener daemon by terminating the listener process. Once it is down, no further connection requests are accepted. However, any remote database activity that is already in progress terminates normally and is not affected by the shutdown.

Listener and network status and errors are recorded in a listener log file (see the next section, "ALLBASE/NET Files on HP-UX") that you can check in the event of problems with remote access.

For instructions on starting the listener daemon and how to handle error conditions, refer to "Starting the Listener Daemon on an HP-UX Server" in the "Setting Up ALLBASE/NET" chapter.

Cross-System Connectivity

With ALLBASE/NET, dissimilar HP systems can be connected. This is called cross-system connectivity.

The client and server can be any combination of HP 9000 Series 700s or 800s, or HP 3000 Series 900. However, the G.0 version of ALLBASE/SQL does not support the HP 9000 Series 300/400 as a server; it can be used as a client.

If you are developing applications that do dynamic queries, do not assume that you know the format of fetched data. The format array, as described in the applicable ALLBASE/SQL application programmer's guide, should always be used to parse dynamically fetched data buffers to avoid erroneous assumptions about data alignment on remote systems.

HP-UX and MPE/iX Syntax Differences

There are some differences in the syntax of DBEnvironment names and login names between HP-UX and MPE/iX systems. These are discussed in the "Setting Up ALLBASE/NET" and "NETUtil Reference" chapters.

Client and Server Login Names

ALLBASE/NET does not require that client and server login names must be identical. However, under certain circumstances, identical client and server login names are recommended. The following guidelines for handling these names are divided into four parts to address the needs of the specific users of ALLBASE/NET and ALLBASE/SQL.

ALLBASE/NET System Administrators

As system administrator on the server, you are responsible for assigning a valid server login name to the client who will be accessing the DBEnvironment. That client's login name is then mapped to the server login name in the NETUsers file.

When possible, it is recommended that the client and server login names be the same. This is a precautionary measure to avoid authorization and ownership problems that arise during preprocessing and execution of certain application programs (such as ALLBASE/Query). Such programs make the erroneous assumption that the **DBEUserID** is based on the client login name. This assumption is true in a local environment, but in a remote environment, the DBEUserID is defined by the server login name specified in the NETUsers file. If the server login name is different from the client login name, you may receive an error message about an invalid authorization or a nonexistent object.

In some cases, such as in HP-UX to MPE/iX connections, it is impossible to have identical client and server login names. In this case, you should make sure that the database administrator is aware of the difference in the client and server login names. (In many cases, you are also the database administrator. If so, see the recommendations below for database administrators.)

ALLBASE/SQL Database Administrators

As database administrator on the server, you need to be aware that some of your remote client users may have client login names that are mapped to different server login names. In such cases, you might want to take some precautionary measures to avoid problems with ALLBASE/SQL object ownership and authorizations.

One recommendation is to create an authorization group and add the server login name (DBEUserID) to the group. Be sure to grant all the desired authorizations to this group. Give the name of the group to the application users and developers on the remote client. Encourage them to use this group as the ownership qualifier when accessing and creating SQL objects that they own.

ALLBASE/SQL Application Programmers

If you are an application programmer on the client, you should design your programs so that the data they are accessing can be moved to a remote node without causing changes to the programs. Here are some programming recommendations:

Do not assume that the DBEUserID is based on the client login name. In a remote access configuration using ALLBASE/NET, the DBEUserID is based on the server login name. The server login name need not be the same as the client login name. Therefore, do not fully qualify an unqualified object's owner with the client's login name. For example, if an end user types the command `select * from tablename`, pass the `tablename` directly to ALLBASE/SQL without prefixing the table name with the client's login name. ALLBASE/SQL resolves unqualified object owners.

When preprocessing programs remotely, qualify the module owner to be the server login name (DBEUserID) or a group name to which the server login name belongs. This is done by using the -o option in the preprocessor command. For example:

```
psqlpas sampledb -i sample.sql -o serverlogin -d
```

ALLBASE/SQL Application Users

If you are a user of an application program, do not assume that the DBEUserID is based on the client login name. In a remote access configuration using ALLBASE/NET, the DBEUserID is based on the server login name. The server login name need not be the same as the client login name. Therefore, do not fully qualify an unqualified object's owner with the client's login name. Talk to your database administrator about possibly creating a group with your server login as a member. The group can be used to fully qualify ALLBASE/SQL objects.

Some existing programs, such as ALLBASE/Query, may incorrectly assume that the client and server login names are the same. If that is the case, remote access using ALLBASE/NET may have restricted functionality unless the server login name can be made the same as the client login name, or the application is rewritten using the recommendations made above for application programmers.

Native Language Support

ALLBASE/NET is localizable to a variety of languages, including languages using 8-bit and 16-bit character sets. In ALLBASE/NET, the default language is *n-computer*. A list of supported languages can be found in /usr/lib/nls/config.

The ALLBASE/NET utility, NETUtil, has native language support for the following:

- The alias name in the AliasDB file
- The DBEnvironment name in the AliasDB file (HP-UX DBEnvironments only)
- Messages and prompts
- Yes/no responses
- Time and date formats

With the exception of the alias name, DBEnvironment name, and yes/no responses, all NETUtil commands and responses must be typed using only ASCII characters. ASCII is a subset of all supported character sets.

The DBEnvironment name in the AliasDB file can contain 8-bit or 16-bit characters if the DBEnvironment it defines resides on an HP-UX system. This is independent of the operating system of the client system where the AliasDB file resides. For example, a NETUtil user on an MPE/iX system can enter a 16-bit character string for the DBEnvironment name if the DBEnvironment being accessed resides on an HP-UX server. DBEnvironment names on MPE/iX are restricted to ASCII characters because MPE/iX does not support names containing 8-bit and 16-bit characters, whereas HP-UX does.

NETUtil does not prohibit users from entering 16-bit characters in node names, login names, or DBEnvironment names. However, when the user attempts to connect to a remote DBEnvironment from an ALLBASE/SQL application, the connection fails if the operating system doesn't support 16-bit characters in node names.

To get localized messages, prompts, and yes/no responses for NETUtil, you must have the appropriate ALLBASE/SQL message catalog installed for the desired language. On HP-UX, set the LANG environment variable to the desired language string to produce the localized user interface.

ALLBASE/NET has no effect on ALLBASE/SQL native language support. The server process inherits the native language of its remote client process. The language of the DBEnvironment itself is determined at create time by the "LANG"= option of the START DBE NEW command, which must always be invoked local to the server machine. The user's native language and the language of the DBEnvironment should be the same. For additional information about ALLBASE/SQL native language support, refer to the *ALLBASE/SQL Reference Manual*.

Gathering Your Data

Before you set up ALLBASE/NET, you need to gather the information about the client and/or server that you will be entering in the AliasDB and NETUsers files.

- If you are the system administrator on both the client and server, all the information you need is readily available to you.
- If you are the system administrator on *only* the client or the server, contact the system administrator on the other server or client to obtain specific information. You also need to provide that system administrator with information specific to your client or server.

As system administrator on the client, you need the following information about the server to put in the AliasDB file:

- Type of database management system being used on the server.
- Absolute pathname of the DBEnvironment on the server.
- Node name of the server where the DBEnvironment is located.
- Machine type of the server.
- Type of data communication used.

As system administrator on the server, you need the following information about the client to put in the NETUsers file and for assigning a server login to the client:

- Client node name.
- Client login name.

To obtain the host name used by the ARPA interface, check the /etc/hosts file on HP-UX. If the hosts file does not exist, check the domain name service file, /etc/resolv.conf. This file contains the names of the systems that have hosts files.

When you have obtained all the needed information, set up ALLBASE/NET using the step-by-step instructions in the "Setting Up ALLBASE/NET" chapter.

2

Setting Up ALLBASE/NET

This chapter provides you with procedures for setting up ALLBASE/NET on both the client and server. It explains how to:

- Use NETUtil to migrate profiles *before* HP-UX 10.0 is installed on a server system.
- Invoke and use NETUtil.
- Set up ALLBASE/NET on the client.
- Set up ALLBASE/NET on the server.

Before you begin setting up ALLBASE/NET, make sure your networking software is up and running. To do this, type `ps -ef |grep hpda` at the system prompt. For more information on this and other listener commands, refer to the section "Setting Up ALLBASE/NET on the Server" in this chapter.

Using NETUtil to Migrate Profiles Before Installing HP-UX 10.0

If you are installing NETUtil for the first time on your system you can skip this section, and go to the section "Using NETUtil." Beginning with HP-UX 10.0, ARPA Services will be the only valid datacomm type for new and existing applications that run on the HP 9000 S700 or S800 client and/or server through ALLBASE/NET. Therefore, you *must* change the alias profiles in the AliasDB file on the client and the user profiles in the NETUsers file on the HP-UX server. You can use the migration commands MIGRATE ALIAS and MIGRATE USER to change profiles where the client and/or server is an HP9000 S700 or S800 and the datacomm type is NS. You must have superuser capability to use the migration commands.

The MIGRATE ALIAS command can be executed with either the automatic or manual option. The automatic option changes *all* the alias profiles. The manual option prompts you for changes to each alias profile. Listed below are the requirements for executing MIGRATE ALIAS for automatic migration on the HP 9000 S700 and S800. If the requirements are not met, you can use the manual migration option of the MIGRATE ALIAS command.

1. Automatic migration must occur before HP-UX 10.0 is installed on client and/or server machines. The implementation of the MIGRATE ALIAS command for the HP 9000 S700 and S800 will use NetIPC calls, and will require a connection to the server. If HP-UX 10.0 is installed on either client or server, the NetIPC calls won't exist.
2. Patch releases are available on HP-UX 9.0 and must be installed on the HP 9000 S700 or S800 client. The patch release numbers are phne_3502 (for S700) and phne_2288 (for S800).
3. Remote loopback must be enabled on the server machine for the conversion of NS node name to ARPA host name to work. The implementation of the MIGRATE ALIAS command on the HP 9000 S700 and S800 makes a connection to the remote loopback

service on the server. To setup remote loopback on an HP 9000 S700 or S800, the remote loopback daemon, /etc/rlbdaemon, should be started in the /etc/netlinkrc startup script. For more information on the rlbdaemon, refer to the *HP-UX Reference - Vol 3*. On an HP 3000, loopback should already be started. If not, issue the command `NETCONTROL START;NET=LOOP`. For more information see the *HP 3000/iX Network Planning and Configuration Guide*.

Note Any HP 9000 S300 client on ALLBASE/SQL pre-G.0 accessing an HP 9000 S700 or S800 server on ALLBASE/SQL G.0 and HP-UX 10.0 must use ARPA as the datacomm type. However, you must use the CHANGE ALIAS command to change the datacomm type from NS to ARPA, you cannot use the MIGRATE ALIAS command.

The migration commands create two files, a migration log file and a server list file.

Migration Log File

Every time you execute and save the changes from MIGRATE ALIAS, MIGRATE USER, MIGRATE ALIAS BACKWARD, OR MIGRATE USER BACKWARD, the migrate log file hpdamig.log is appended. If the file does not exist it is created.

After migration is successfully, you can delete the following temporary files:

/etc/daalfmbk

/etc/tmpalmbk

/etc/dausfmbk

/etc/tmpusmbk

/tmp/hpdamig.log

/tmp/tmpmig.log

/etc/dasvrlst

/etc/dasvrmbk

/etc/tmpsvmbk

Here is a sample log file for a client. The MIGRATE ALIAS BACKWARD command returns the AliasDB file to its state before the 3:35 PM migration.

```
TUE, APR 20, 1993,  3:35 PM: MIGRATE ALIAS
    Alias: testdb1  Migrated Server : snode.sdomain.sorg -> snode
    Cannot migrate alias (testdb2) for server (node1.dom.org) (DBERR 28029)
    Alias: testdb3  Migrated Server : hana.dom.org -> hana1.site1.co1.domain1
    Alias: testdb40  Migrated Server : merc.dom.org -> merc2
    Alias: testdb41  Migrated Server : mach1.dom1.org1 -> mach1.site1.co1.dom1

 Migration complete.
 4 alias(es)     migrated successfully.
 1 alias(es) NOT migrated successfully - see above for error information.

TUE, APR 20, 1993,  3:40 PM: MIGRATE ALIAS BACKWARD
```

Here is a sample log file for a server. The MIGRATE USER BACKWARD command returns the NETUsers file to its state before the 3:41 PM migration.

```
TUE, APR 20, 1993,  3:40 PM: MIGRATE USER
    Migrated Client : Umach.Udom.Uorg -> Umach.Usite.Uco.Udom
          Login Name : user1.somegrp.someacct
          Login Name : user2.somegrp.someacct

 Migration complete.
 2 user(s)     migrated successfully.
 0 user(s) NOT migrated successfully - see above for error information.

TUE, APR 20, 1993,  3:41 PM: MIGRATE USER
    Migrated Client : mach2.dom.org -> mach2.site.co.domain
          Login Name : user3.somegrp.someacct

 Migration complete.
 1 user(s)     migrated successfully.
 0 user(s) NOT migrated successfully - see above for error information.

TUE, APR 20, 1993,  3:45 PM: MIGRATE USER BACKWARD
```

Server List File

Each alias profile that was changed to show a different datacomm type requires that all corresponding user profiles on the server be changed also. If you use MIGRATE ALIAS to change an alias profile, a server list file, /etc/dasvrlst, is created and lists all affected corresponding servers for the client. At the top of the file, the NS value and ARPA value (if it can be determined by ALLBASE/NET) for the client node name are displayed.

Here is a sample server list file.

```
Client Node Name (NS value) : cnode.cdom.corg
Client Node Name (ARPA value) : chost.site.co.domain

Server Node Name List (NS value, ARPA value) :
(snode.sdomain.sorg, snode)
(hana.dom.org, hana)
(merc.dom.org, merc2)
(mach1.dom1.org1, mach1)
```

Using NETUtil

NETUtil is the utility you use to create and maintain your AliasDB and NETUsers files. For descriptions of the NETUtil commands and examples of how they are used, refer the "NETUtil Reference" chapter.

Invoking NETUtil

To invoke NETUtil, type `netutil` at the shell prompt. The NETUtil prompt looks like this:

```
netutil=>
```

Special Control Characters

You can enter these special control characters while you are using any NETUtil command:

- A double forward slash (//), which stops the execution of a command while in prompting mode and returns the NETUtil prompt.
- The user's interrupt character (usually the ASCII DEL key or Control-C), which terminates an executing command and returns the NETUtil prompt.
- The user's EOF character (usually Control-D), which terminates an executing command and returns the shell prompt.

Setting Up ALLBASE/NET on the Client

Note

The following steps assume that you are setting up ALLBASE/NET on an HP-UX client and the DBEnvironment is located on an HP-UX server. However, if the DBEnvironment is located on an MPE/iX server, you must use MPE/iX naming conventions when you enter information in an alias profile. MPE/iX naming conventions and syntax are described for each NETUtil command in "NETUtil Reference" chapter.

Procedures for setting up ALLBASE/NET on an MPE/iX client are in "Setting Up ALLBASE/NET" chapter of the *900 Series HP 3000 Computer Systems ALLBASE/NET User's Guide*.

There are two tasks you perform on the client to set up ALLBASE/NET:

- Adding alias profiles to the AliasDB file.
- Checking the /etc/services file (for ARPA services).

The AliasDB file contains an alias profile for each alias name that corresponds to a DBEnvironment on the server. You create the file the first time you add an alias profile with NETUtil's ADD ALIAS command. Once the file is created, use ADD ALIAS to add more alias profiles to it.

NETUtil provides three commands for maintaining alias profiles in the AliasDB file: CHANGE ALIAS, DELETE ALIAS, and SHOW ALIAS. Instructions for using these commands follow later in this section. If you are using ARPA services, the /etc/services file

on the client requires an ARPA user entry. If this entry doesn't exist, you must update the file. The section "Checking the Services File on the Client" later in this chapter discusses updating the file. If loopback is being used, the ARPA name needs to be the local host.

Adding an Alias Profile

Follow these steps to add an alias profile to the AliasDB file:

1. Invoke NETUtil.
2. Type `ADD ALIAS` at the NETUtil prompt. ADD ALIAS prompts you first for the alias name of the DBEnvironment on the server and then prompts you for additional information.
3. Respond to each prompt as follows:

Prompt	Response
`Alias Name:`	Enter the alias name of the DBEnvironment on the server, to be used by an application on the client. The alias name must be unique on the client and does not have to be the same as the DBEnvironment name. It consists of 16-bit characters or other ASCII or non-ASCII 8-bit character, contains up to 128 characters, and must not begin with an asterisk (*). The alias name is case-sensitive.
`Database Server Type:`	Enter ALLBASE/SQL, the database management system server type on the server. (HP SQL is still supported as a database server type. However, if you enter HP SQL, NETUtil displays it as ALLBASE/SQL.) The database server type name is not case-sensitive and is stored and displayed in uppercase characters.
`DBEnvironment Name:`	Enter the *absolute* pathname of the DBEnvironment on the server. This name follows the file naming conventions of the operating system in which the DBEnvironment resides. An HP-UX DBEnvironment name cannot exceed 128 characters in length and is case-sensitive.
`Server Node Name:`	Enter the host name of the server. The server host name is not case-sensitive.
`Machine Type:`	Enter the machine type of the server: HP9000 S700, HP9000 S800, or HP3000 S900. A space is required between HP9000 or HP3000 and the series number. The machine type name is not case-sensitive.
`Datacomm Type:`	Enter the type of network service you are using, either ARPA or NS. If you are using HP-UX release 10.0 on either the client or server, ARPA is the only valid entry. The datacomm type name is not case-sensitive.

After you respond to the last prompt, the following message appears, followed by the NETUtil prompt:

```
Profile added.
```

For an example of how ADD ALIAS is used, refer to "Examples of Setting Up ALLBASE/NET" at the end of this chapter.

Warning **Use only NETUtil to modify the AliasDB file. Using other software, such as an editor, could damage the file.**

Changing an Alias Profile

You can change the contents of an alias profile with the CHANGE ALIAS command.

For the naming conventions and syntax to use to respond to the prompts, refer to "Adding an Alias Profile," above.

To change an alias profile:

1. Type `CHANGE ALIAS` at the NETUtil prompt.

   CHANGE ALIAS prompts you first for the current alias name of the profile you want to change and then prompts you for additional information.

2. Enter the current alias name.
3. Enter a new alias name. If you do not want to change the current alias name, press the Return key.

   The current value for each parameter in the entry is displayed in succession. You are prompted to enter a new value for each parameter. If you do not want to change a current value, press the Return key.

4. Confirm the changes by entering Y or N.

For examples of how CHANGE ALIAS is used, refer to the "NETUtil Reference" chapter.

Deleting an Alias Profile

To delete one or more aliases from the AliasDB file use the DELETE ALIAS command at the NETUtil prompt. If you enter an alias name, you delete the alias profile for that name. If you enter an asterisk, you delete all alias profiles for a specific server.

For the naming conventions and syntax to use to respond to the prompts, refer to "Adding an Alias Profile," above.

To delete one alias profile, do the following:

1. Enter the alias name of the alias profile you want to delete.
2. Confirm the deletion by entering Y or N.

To delete alias profiles for a specific server node, do the following:

1. Enter an asterisk (*) for the Alias Name.
2. Enter the database server type (ALLBASE/SQL).
3. Enter the server node name.
4. Confirm the deletions by entering Y or N.

For examples of how DELETE ALIAS is used, refer to "NETUtil Reference" chapter.

Displaying an Alias Profile

To display one or more alias profiles in the AliasDB file, use the SHOW ALIAS command at the NETUtil prompt.

For the naming conventions and syntax to use to respond to the prompts, refer to "Adding an Alias Profile," above.

To display one alias profile, enter the alias name of the alias profile you want to display.

To display all of the alias profiles, enter an asterisk for the alias name.

For an example of how SHOW ALIAS is used, refer to "Examples of Setting Up ALLBASE/NET" at the end of this chapter.

Checking the Services File on the Client

If you are using ARPA services, check for a user entry in the /etc/services file on the client. This file is used by ARPA for dynamic port assignment. The file entry for the ALLBASE/NET service has the official service name DAServer, the port number 987, and the protocol name tcp. The file should look like this:

```
ftp             21/tcp
login           513/tcp
   .
   .
   .
DAServer        987/tcp
```

If the last line above is not there, update the /etc/services file to include this line. For more information on /etc/services, refer to *Installing and Administering ARPA Services*.

Setting Up ALLBASE/NET on the Server

Note The following steps assume that you are setting up ALLBASE/NET on an HP-UX server and the user is located on an HP-UX client. However, if the user is located on an MPE/iX client, you must use MPE/iX naming conventions when you enter information in a user profile. MPE/iX naming conventions and syntax are described for each NETUtil command in the "NETUtil Reference" chapter.

Procedures for setting up ALLBASE/NET on an MPE/iX server are in "Setting Up ALLBASE/NET" chapter of the *900 Series HP 3000 Computer Systems ALLBASE/NET User's Guide*.

To set up ALLBASE/NET, perform the following steps:

- Start the listener.
- Assign a server login name for the client.
- Add user profiles to the NETUsers file.
- Check the /etc/service file (for ARPA services).

Starting the Listener Daemon on an HP-UX Server

As system administrator on the server, you need to start the listener daemon after ALLBASE/NET is installed. (You must have superuser capability to do this.) The listener daemon must be started the first time ALLBASE/NET is used after a system boot.

Note These steps are for starting the listener on an HP-UX server. To start the listener on an MPE/iX server, refer to "Setting Up ALLBASE/NET on the Server" chapter of the *900 Series HP 3000 Computer System ALLBASE/NET User's Guide*.

To start the listener daemon, type:

```
$ /usr/bin/hpdaX
```

where *X* is either NS or ARPA.

When the listener daemon is started, a log file named /tmp/hpda*X*.log is automatically created (where *X* is either NS or ARPA). This is a circular file that holds 1000 entries containing listener daemon status information.

Note NS is not included in the 10.0 version of HP-UX.

Error Handling

If an error occurs, determine the cause of the error and take the appropriate action. If you are using NS, the NetIPC error code is contained in the last three digits of the error number returned.

For descriptions of NetIPC error messages, refer to the *NetIPC Programmer's Guide*.

If you are using ARPA services, refer to the *ALLBASE/SQL Message Manual* for error message descriptions. The last three digits are specify the ARPA error message number.

After the error is corrected, terminate the listener daemon process, remove the log file (if desired), and restart the listener daemon using the following steps. You must have superuser capability to remove the log file.

1. Obtain the process identification number (pid) of the listener daemon by typing:

   ```
   $ ps -ef | grep hpdaX
   ```

2. Kill the process by typing:

   ```
   $ kill -9 pid
   ```

3. Remove the /tmp/hpda*X*.log file by typing:

   ```
   $ rm /tmp/hpdaX.log
   ```

   If this file is not removed, it will be appended to as new activities are logged.

4. Restart the listener daemon by typing:

   ```
   $ /usr/bin/hpdaX
   ```

5. Once the listener daemon is started, modify the /etc/netlinkrc file on the server to start the listener daemon files automatically when the system is booted. For example:

   ```
   $ vi /etc/netlinkrc
   .
   .
   .
   #start hpdaNS daemon, for HP-UX 9.0 or earlier
   #for ALLBASE/NET
   if [ -f /usr/bin/hpdaNS ]
   then
     /bin/rm -f /tmp/hpdaNS.log
     /usr/bin/hpdaNS
   fi
   #start hpdaARPA daemon
   #for ALLBASE/NET
   if [ -f /usr/bin/hpdaARPA]
   then
     /bin/rm -f /tmp/hpdaARPA.log
     /usr/bin/hpdaARPA
   fi
   ```

Examples of Listener Daemon Log Files

In the following NS log file example, the first three lines show that the listener daemon has started; the next three lines show an example of the information displayed when a connection has been made.

```
$ more /tmp/hpdaNS.log
Mon Jul 11 13:30:28 1993:  ipccreate: 0
Mon Jul 11 13:30:28 1993:  ipcname: 0
Mon Jul 11 13:30:28 1993:  ipccontrol: 0
Mon Jul 11 13:30:28 1993:  ipcrecvcn: 0
```

```
Tue Jul 12 12:27:12 1993:  ipccontrol: 0
Tue Jul 12 12:27:12 1993:  Client PID:19914  Server PID:11910
Client Nodename: cnode.cdomain.corg
```

The ARPA log file is similar, as the following example shows.

```
$ more /tmp/hpdaARPA.log
Mon Jul 11 13:30:28 1993:  socket: 0
Mon Jul 11 13:30:28 1993:  bind: 0
Mon Jul 11 13:30:28 1993:  listen: 0
Tue Jul 12 12:27:12 1993:  accept: 0
Tue Jul 12 12:27:12 1993:  Client PID:19914  Server PID:11910
Client Nodename: clientsystem
```

If there are no errors, a zero (0) is returned, as indicated in the examples above. A number other than zero or another message appears in the log file if an error condition has occurred.

Assigning a Server Login

Assign a server login for the user on the client who is accessing the DBEnvironment on the server. Use the information about the client login name and client node name that you obtained from the client's system administrator. For more information about server login names, refer to "Client and Server Login Names" under "Cross-System Connectivity" in the "Getting Started" chapter.

Add the server login name to the /etc/passwd file on the server. Put an asterisk (*) in the password field of the entry so that someone with the same name as the server login name cannot use the remote login (rlogin) command to access files on the server.

Note If you are setting up ALLBASE/NET on an HP-UX server but the user is on an MPE/iX client, refer to "Client and Server Login Names" under "Cross-System Connectivity" in the "Getting Started" chapter.

Authorities

The user for whom you are assigning the server login name will need the necessary authorities (such as CONNECT authority) for accessing the DBEnvironment on the server. These authorities must be granted by the database administrator responsible for the DBEnvironment.

Adding a User Profile

To add a user profile to the NETUsers file:

1. Invoke NETUtil.
2. Type `ADD USER` at the NETUtil prompt.

   ADD USER prompts you first for the node name of the client at which the user is located and then prompts you for additional information.

3. Respond to each prompt as follows:

Prompt	Response
`Client Node Name:`	Enter the name of the client node at which the user is located. The client node name is not case-sensitive.
`Client Login Name:`	Enter the login name of the user on the client. This name can be either a HP-UX or MPE/iX login. A wildcard (@) can be used as the login name. This is explained in the "NETUtil Reference" chapter. If the client is HP-UX, this name follows HP-UX naming conventions for a login name, cannot exceed eight characters in length, and is case-sensitive. If the client is MPE/iX, this name follows MPE/iX naming conventions for a logon name, a component cannot exceed eight characters in length, and is not case-sensitive. The session and logon group names are optional. The session name is not allowed. The group name is required.
`Server Login Name:`	Enter the server login name you assigned to the user accessing the DBEnvironment on the server. This name follows HP-UX naming conventions for a login name, cannot exceed eight characters in length, and is case-sensitive. The server login name must reside in the /etc/passwd file. When possible, this name should be the same as the client login name. Refer to "Client and Server Login Names" under "Cross-System Connectivity" in the "Getting Started" chapter.

After you respond to the last prompt, the following message appears followed by the NETUtil prompt:

```
Profile added.
```

For an example of how ADD USER is used, refer to "Examples of Setting Up ALLBASE/NET" at the end of this chapter.

Warning **Use only NETUtil to modify the NETUsers file. Using other software, such as an editor, could damage the file.**

Changing a User Profile

You can change the contents of a user profile with the CHANGE USER command.

For the naming conventions and syntax to use to respond to the prompts, refer to "Adding a User Profile," above.

To change a user profile, do the following:

1. Type `CHANGE USER` at the NETUtil prompt.

   CHANGE USER prompts you first for the node name of the client at which the user is located and then prompts you for additional information.
2. Enter the client node name of the profile you want to change.
3. Enter a new client node name. If you do not want to change the current name, press the Return key.
4. Enter the current client login name of the user or an asterisk (*).

   If you enter an asterisk, all client node names in the NETUsers file that match the current client node name are changed to the *new* client node name.

   If you enter the current client login name of the user, you are prompted for a new client login name.
5. Enter a new client login name.

   The current server login name is displayed, followed by a prompt for a new server login name.
6. Enter a new server login name. If you do not want to change the current name, press the Return key.
7. Confirm the changes by entering Y or N.

For examples of how CHANGE USER is used, refer to the "NETUtil Reference" chapter.

Deleting a User Profile

To delete one or more user profiles from the NETUsers file use the DELETE USER command at the NETUtil prompt. The current client node name and client login name uniquely identify the entry to be deleted. If you enter the client login name, you delete the user profile that has the same name. If you enter an asterisk, you delete all user profiles with the current client node name.

For the naming conventions and syntax to use to respond to the prompts, refer to "Adding a User Profile," above.

To delete a single user profile, do the following:

1. Enter the client node name for the user profile you want to delete.
2. Enter the client login name.
3. Confirm the deletion by entering Y or N.

To delete all user profiles with the same client node name, do the following:

1. Enter the client node name for the user profiles you want to delete.
2. Enter an asterisk (*) for the client login name.
3. Confirm the deletions by entering Y or N.

For examples of how DELETE USER is used, refer to "NETUtil Reference" chapter.

Displaying a User Profile

To display the contents of one or more user profiles in the NETUsers file type the SHOW USER command at the NETUtil prompt. You can display all user profiles, user profiles for a specific node, or user profiles with the same client login name.

For the naming conventions and syntax to use to respond to the prompts, refer to "Adding a User Profile," above.

To display one user profile, do the following:

1. Enter the client node name for the user profiles you want to display.
2. Enter the client login name for the user.

To display all of the user profiles in the NETUsers file, do the following:

1. Enter an asterisk (*) for the client node name.
2. Enter an asterisk (*) for the client login name.

To display all user profiles with the same client login name, do the following:

1. Enter an asterisk (*) for the client node name.
2. Enter the client login name.

To display all user profiles for a specific client node, do the following:

1. Enter the client node name.
2. Enter an asterisk (*) for the client login name.

For an example of how SHOW USER is used, refer to "Examples of Setting Up ALLBASE/NET" at the end of this chapter.

Examples of Setting Up ALLBASE/NET

Figure 2-1 shows a profile in the AliasDB and NETUsers files. Both the client and server are HP-UX systems.

Note that the NETUtil commands have varying user restrictions. Refer to the chapter "NETUtil Reference" for complete information.

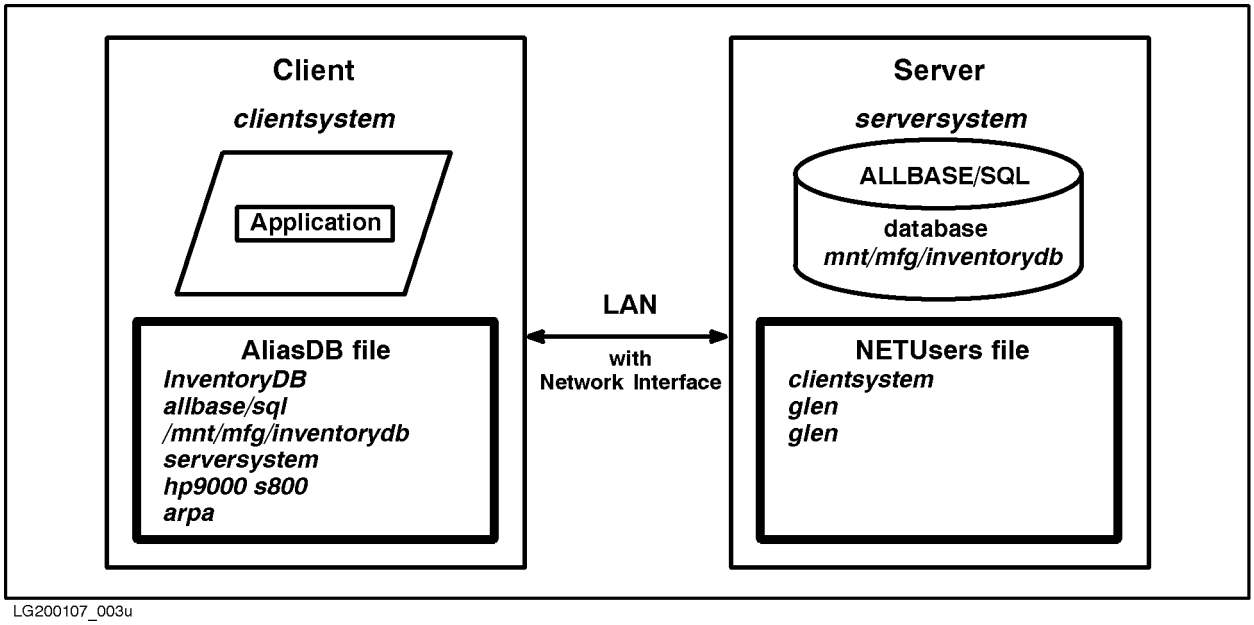


Figure 2-1. Sample Profiles in AliasDB and NETUsers Files

To create the sample AliasDB file on the client and add the information shown, you would do the following:

1. Invoke NETUtil on the client.
2. Type `ADD ALIAS` at the NETUtil prompt. Then reply to the prompts.

```
Alias Name: InventoryDB
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: /mnt/mfg/inventorydb
Server Node Name: serversystem
Machine Type: hp9000 s800
Datacomm Type: arpa

Profile added.
netutil=>
```

The ADD ALIAS command creates the AliasDB file and adds your alias profile to it.

3. Type SHOW ALIAS to check the alias profile in the AliasDB file:

```
Alias Name: InventoryDB

Alias Name: InventoryDB
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: /mnt/mfg/inventorydb
Server Node Name: serversystem
Machine Type: HP9000 S800
Datacomm Type: ARPA
```

Similarly, to create the above NETUsers file on the server and add the information shown in Figure 2-1 , you would do the following:

1. Invoke NETUtil on the server.
2. Type ADD USER at the NETUtil prompt. Then reply to the prompts with the necessary data for your user profile:

```
Client Node Name: clientsystem
Client Login Name: glen
Server Login Name: glen

Profile added.

netutil=>
```

   The ADD USER command creates the NETUsers file (if necessary) and adds a user profile to it.

3. Type SHOW USER to check the user profile in the NETUsers file:

```
Client Node Name: clientsystem
Client Login Name: glen

Client Node Name: clientsystem
Client Login Name: glen
Server Login Name: glen

netutil=>
```

4. Add the server login name to the /etc/passwd file on the server:

```
$ vi /etc/passwd

root:SuIeof1mj31A4:0:1:Root at horse:/:/bin/sh
bicca:25Fnph4JsBXng:5:1:Leigh Bicca 49u,77033:/mnt/admin/bicca:/bin/csh
.
.
.
glen:*:100:10:/mnt/mfg:/bin/csh
.
.
```

   Remember to put an asterisk (*) in the password field so that another user named glen cannot use the remote login (rlogin) to access files on the server.

3

Using ALLBASE/SQL Through ALLBASE/NET

After the AliasDB and NETUsers files have been created and contain valid profiles and the listener daemon is running, an application can transparently access a remote ALLBASE/SQL DBEnvironment using ALLBASE/NET.

You can use ALLBASE/NET with many applications. Some of them are discussed in this chapter:

- ALLBASE/ISQL
- ALLBASE/SQL preprocessors
- User-written application programs
- ALLBASE/Query (HP Visor)
- ALLBASE/4GL
- Information Access
- PC-based Application Software

Using ALLBASE/ISQL

To use ISQL to access a remote DBEnvironment, do the following:

1. Invoke ISQL on the client by typing:

```
$ isql
```

2. Initiate a DBE session by typing:

```
isql=> connect to 'AliasName';
```

AliasName references a particular DBEnvironment on the server. The alias name is specified in the AliasDB file on the client. The ISQL commands that are issued after this CONNECT command will act on the remote DBEnvironment. However, be aware that if you execute the STORE command on a remote DBEnvironment, the file is written to the client.

Note ALLBASE/NET does not allow the use of the START DBE NEWLOG or START DBE NEW commands remotely.

Using the ALLBASE/SQL Preprocessors

To use the ALLBASE/SQL preprocessors, do the following steps:

1. In the source code, define the alias name of the DBEnvironment to be accessed on the server.

```
begin
.
.
.
EXEC SQL CONNECT TO 'AliasName';
.
.
.
EXEC SQL RELEASE;
end
```

   You can specify an alias name dynamically by using a host variable. For example:

```
EXEC SQL BEGIN DECLARE SECTION;

AliasName: packed array [1..128] of char;

    .
    .
    .
EXEC SQL END DECLARE SECTION;
EXEC SQL CONNECT TO :AliasName;
```

2. Invoke the preprocessor on the client and use the *AliasName* for the DBEnvironment. In the following example, the preprocessor for Pascal is used on the Series 800.

```
$ psqlpas AliasName -i mysource.sql -p mysource.p -o serverlogin -m mymodule
```

   where:

   a. *AliasName* is the name of the alias profile described in the AliasDB file.
   b. mysource.sql is the name of the source file.
   c. mysource.p is the name of the modified source code file.
   d. owner is the server login name on the server that is specified in the NETUsers file. The owner has implicit RUN authority on the module created.
   e. mymodule is the name of the module to be stored.

A *module* is created and stored in the system catalog of the DBEnvironment on the server. The modified source code file and the include files created by the preprocesor remain on the client.

3. Compile and link the modified source file with the necessary libraries on the client.

   ```
   $ pc mysource.p -lsql -lportnls -o someprog.r
   ```

   where:

   a. mysource.p is the modified source code file.
   b. the -lsql option tells the compiler to link with the runtime library /usr/lib/libsql.a.
   c. the -lportnls option tells the compiler to link with the native language routines.
   d. the -o option causes the creation of the executable output file someprog.r.

   The executable program resides on the client. The program someprog.r is now ready to run from the client to access the DBEnvironment on the server.

For more information on preprocessors, refer to the ALLBASE/SQL application programming guide for the language you are using.

Running an Application Program

An application that was developed on a server can be run on the client.

Run the program by typing:

```
$ someprog.r
```

Authorizations

If an application program uses a CONNECT command to start a DBE session, the user must have both CONNECT authority and RUN authority to run the program. The runtime CONNECT and RUN authorities are based on the user's server login. Additionally, the module owner must have permission to access all objects acted upon by the program. This usually means one or more of the table or view authorities must be granted to the module owner. There may be other authorities needed as well.

The module owner is defined at preprocess time by the -o option of the preprocess command or, by default, the client's login. Refer to "Client and Server Login Names" under "Cross-System Connectivity" in the "Getting Started" chapter.

Using ALLBASE/Query

To access a remote DBEnvironment using ALLBASE/Query (formerly called HP Visor), enter the alias name on the ALLBASE/Query Choose DBEnvironment screen. If you have a DBEList, add the alias name to it.

Using ALLBASE/4GL

To access a remote DBEnvironment using ALLBASE/4GL, enter the alias name in the SQL Database Name field on the Application Definition Screen of the ADMINIST application. An SQL Owner Group must also be specified at the same time.

Using Information Access SQL/UX

Information Access SQL/UX, installed on the ALLBASE/NET client, allows PC users to select and retrieve information from an ALLBASE/SQL database on the server. It works with the PC-based applications Information Access PC and HP NewWave Access. With Information Access SQL/UX, users can view and work with ALLBASE/SQL tables in the following ways:

- Select only the data needed for viewing and manipulation.
- Sort and summarize table information.
- Save and/or print the results of table manipulation.
- Create and print reports from result tables with Information Access PC's Report Writer.
- Reformat result tables for use in other PC applications.

For more information about Information Access, refer to the manuals: *Information Access SQL/UX*, *Using Information Access PC*, and *HP NewWave Access User's Guide*.

Using PC-based Application Software

From your PC, you can use Microsoft Windows-based application software (4GLs, 3GLs, query or report writing tools) to access ALLBASE/SQL. Simply copy the HP PC API installation file, HPPCAPI.PUB.SYS, to your PC and follow the installation directions. For more information refer to the *HP PC API User's Guide for ALLBASE/SQL*.

Conditions for Connecting to a Remote DBEnvironment

When an application issues a command to connect to a DBEnvironment, a remote connection is attempted if one of the following conditions exist on the client:

- The /usr/lib/hpsqlproc file (database server process) does not reside on the client.
- There is no file in the client's working directory with the DBEnvironment name referred to in the CONNECT command.
- There is a file with that name on the client but it is not a DBECon file.
- The application is running on a diskless machine.

If the remote connection is not successful, an error message is returned, followed by a warning message about the local connection attempt (DBWARN 29007), as shown in figure 3-1.

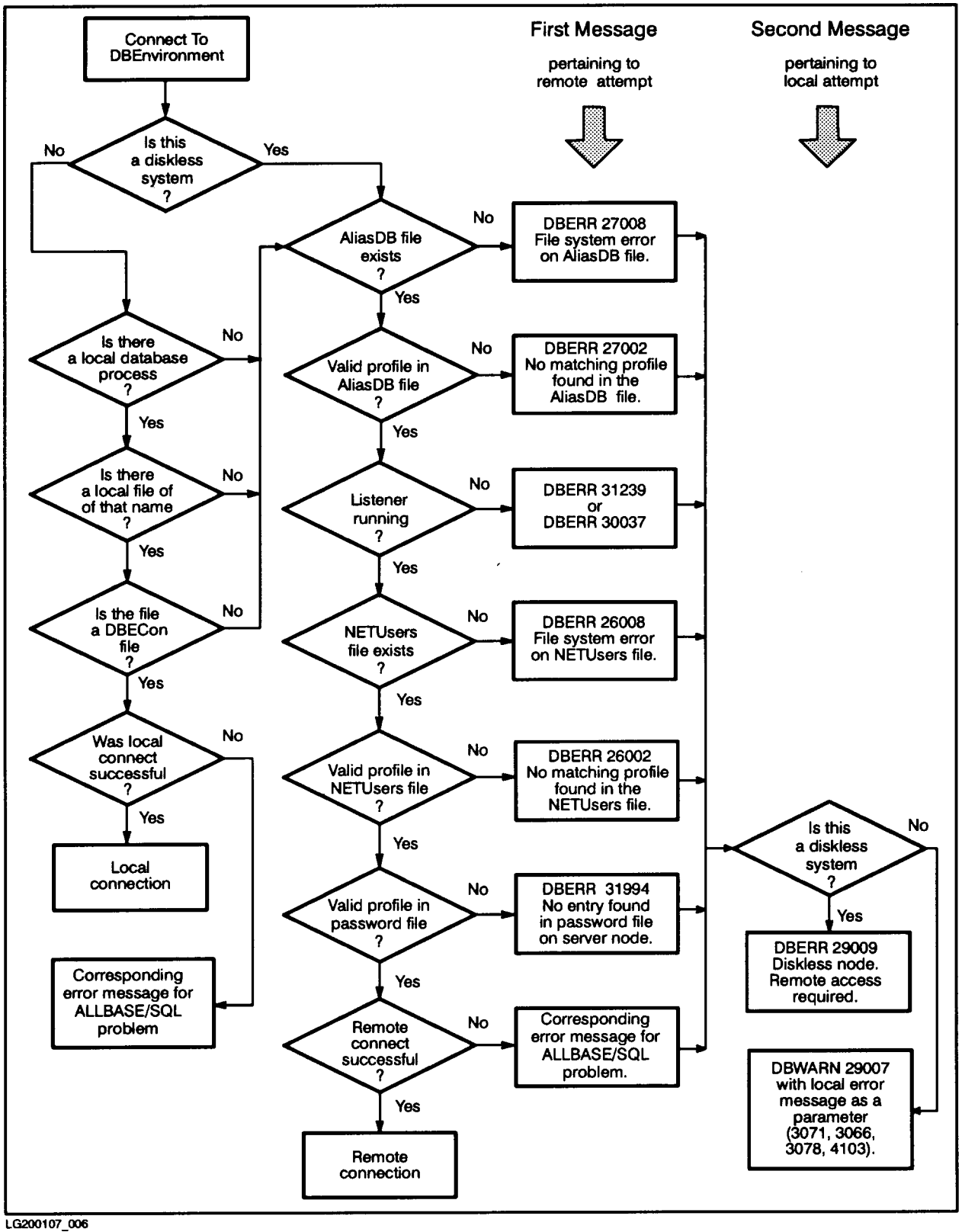


Figure 3-1. Connecting to a DBEnvironment

4

NETUtil Reference

This chapter describes the NETUtil commands and gives examples of how they are used.

To invoke NETUtil, type `netutil` at the shell prompt. The NETUtil prompt looks like this:

```
netutil=>
```

NETUtil lets you type these special control characters:

- A double forward slash (//), which terminates a command while in prompting mode and returns the NETUtil prompt.
- The user's interrupt character (usually the ASCII DEL key or Control-C), which terminates an executing command and returns to the NETUtil prompt.
- The user's EOF character (usually Control-D), which terminates an executing command and returns to the shell prompt.

The NETUtil commands have varying user restrictions. You must be superuser to use the following commands:

ADD ALIAS
ADD USER
CHANGE ALIAS
CHANGE USER
DELETE ALIAS
DELETE USER
MIGRATE ALIAS
MIGRATE ALIAS BACKWARD
MIGRATE USER
MIGRATE USER BACKWARD
SHOW USER

Command Summary

The table below lists the NETUtil commands along with their abbreviations and functions.

Table 4-1. NETUtil Commands

Command	Abbre-viation	Function
ADD ALIAS	AA	Adds an alias profile to the AliasDB file.
ADD USER	AU	Adds a user profile to the NETUsers file.
CHANGE ALIAS	CA	Changes an alias profile in the AliasDB file.
CHANGE USER	CU	Changes a user profile in the NETUsers file, or changes all user profiles having a given client node name with a new client node name in the NETUsers file.
DELETE ALIAS	DA	Deletes an alias profile from the AliasDB file, or deletes all alias profiles with a specified server node name from the AliasDB file.
DELETE USER	DU	Deletes a user profile from the NETUsers file, or deletes all user profiles with a specified client node name from the NETUsers file.
EXIT	E	Terminates execution of the NETUtil program.
HELP	H or ?	Displays and describes all NETUtil commands.
MIGRATE ALIAS	MA	Changes the datacomm type in an alias profile from NS to ARPA.
MIGRATE ALIAS BACKWARD	MAB	Changes the AliasDB file back to the state prior to the most recent saved migration attempt; changes ARPA datacomm types that were migrated back to NS datacomm types.
MIGRATE USER	MU	Changes the profiles for a given client node name with an NS value to a client node name with an ARPA value.
MIGRATE USER BACKWARD	MUB	Changes the profiles for a given client node name file back to the state prior to the most recent saved migration attempt; client node names that were migrated to ARPA values are changed to NS values.
QUIT	Q	Terminates execution of the NETUtil program.
SET ECHO	SE	Echoes user input to a standard output file.
SHOW ALIAS	SA	Displays one or more alias profiles in the AliasDB file.
SHOW USER	SU	Displays one or more user profiles in the NETUsers file.
SYSTEM	!	Escapes temporarily to the operating system for the execution of a single operating system command.

ADD ALIAS

Use the ADD ALIAS command to add an alias profile to the AliasDB file on the client. The first time you use this command to add an alias profile, the AliasDB file is created. To display the contents of one or more alias use the SHOW ALIAS command.

ADD ALIAS can be abbreviated to AA.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> ADD ALIAS
Alias Name: AliasName
Database Server Type: DBServerType
DBEnvironment Name on Server: DBEnvironmentName
Server Node Name: ServerNodeName
Machine Type: MachineType
Datacomm Type: DatacommType
```

Parameters

AliasName is the alias name of the DBEnvironment that resides on the server, to be used by an application on the client. This name must be unique on the client. The alias name does not need to be the same as the DBEnvironment name. It can consist of 16-bit characters or other ASCII or non-ASCII 8-bit characters and can contain up to 128 characters. The HP-UX alias name is case-sensitive; the MPE/iX alias name is not case-sensitive. An alias name is stored and displayed exactly as it was entered.

The first character of this field cannot be an asterisk (*).

DBServerType is ALLBASE/SQL, the database management system server type on the server. (HP SQL is still supported as a database server type. However, if you enter HP SQL, NETUtil displays it as ALLBASE/SQL.) The server type name is not case-sensitive and is stored and displayed in uppercase characters.

DBEnvironmentName is the *absolute* pathname of the DBEnvironment on the HP-UX server or fully qualified filename on the MPE/iX server. This name follows the file naming conventions of the operating system in which the DBEnvironment resides.

An HP-UX DBEnvironment name cannot exceed 128 characters in length, is case-sensitive, and is localizable (8-bit and 16-bit characters).

An MPE/iX DBEnvironment name cannot exceed 26 characters in length and is not case-sensitive.

The DBEnvironment name is stored and displayed exactly as it was entered.

ServerNodeName is the node name of the server you want to access. The server node name is not case-sensitive and is stored and displayed exactly as it was entered. The name format is based on the datacommm type.

MachineType is the machine type of the server. Enter HP9000 S700, HP9000 S800, or HP3000 S900. A space is required between HP9000 or HP3000 and the series number. The machine type name is not case-sensitive and is stored and displayed in uppercase characters.

DatacommType is the network service interface you are using. The datacomm type is either NS or ARPA. The data communication type name is not case-sensitive and is stored and displayed in uppercase characters. If you are using HP-UX release 10.0 on either the client or server, ARPA is the only valid entry.

Examples

In this example, the alias profile is for a DBEnvironment on an HP-UX server:

```
netutil=> add alias
Alias Name: purchdbe
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: /users/dbsupport/purchdbe
Server Node Name: system1
Machine Type: hp9000 s800
Datacomm Type: arpa

Profile added.

netutil=>
```

In this example, the alias profile is for a DBEnvironment on an MPE/iX server:

```
netutil=> add alias
Alias Name: InvenDB
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: PartsDB.DBgroup.Inven
Server Node Name: snode.sdomain.sorg
Machine Type: hp3000 s900
Datacomm Type: arpa

Profile added.

netutil=>
```

ADD USER

Use the ADD USER command to add a user profile to the NETUsers file on the server. The first time you use this command to add a user profile, the NETUsers file is created. To display the contents of one or more user profiles, use the SHOW USER command.

ADD USER can be abbreviated to AU.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> ADD USER
Client Node Name: ClientNodeName
Client Login Name: ClientLoginName
Server Login Name: ServerLoginName
```

Parameters

ClientNodeName is the node name of the client system. The client node name is not case-sensitive.

ClientLoginName is the login name of the user on the client.

The HP-UX client login name follows HP-UX naming conventions for a login name, cannot exceed eight characters in length, and is case-sensitive. The login can be wildcarded with the use of the "@" symbol.

The MPE/iX client login name follows MPE/iX naming conventions for a login name and is not case-sensitive. The session name is not allowed and the group name is required.

Any or all of the client user login name components may be wildcarded using the "@". When this is done, only the non-wildcarded components need to match the user's actual login. For example, a client login entry of @.acct7,group6 would allow any user in the acct7 account and group6 group to have access to the server login name.

If more than one client login name entry matches the actual user's login, then the highest prioritized match will be used in determining the server login name. Prioritization is as follows: A fully qualified entry has highest priority (user7.acct3,group2). A fully wildcarded entry has the lowest priority (@.@,@). When dealing with individual components, an actual match for a specific component takes priority over a wildcard match. When a conflict with this arises between different components, then the account is prioritized first, the user second, and finally the group. For example, @.acct3,@ would have a higher priority than user2.@,group1, so the corresponding server login for @.acct3,@ would be used.

The client login name is stored and displayed exactly as it was entered.

ServerLoginName is the login name used on the server for the user accessing the DBEnvironment on the server.

This name follows HP-UX naming conventions for a login name, cannot exceed eight characters in length, and is case-sensitive. It is stored and displayed exactly as it was entered. The server login name must reside in the /etc/passwd file.

The server login name allows the client node name and the client login name to be mapped to a single login on the server. Only one server login name is allowed for each client node name and client login name combination. When possible, the server login name should be the same as the client login name. This avoids problems in some applications that make assumptions about client and server logins. Refer to "Client and Server Login Names" under "Cross-System Connectivity" in the "Setting Up ALLBASE/NET" chapter.

Examples

In this example, the user profile is for a user on an HP-UX client.

```
netutil=> add user
Client Node Name: clientsystem
Client Login Name: glen
Server Login Name: glen

Profile added.

netutil=>
```

In this example, the user profile is for a user on an MPE/iX client.

```
netutil=> add user
Client Node Name: cnode.cdomain.corg
Client Login Name: glen.orders,purchasing
Server Login Name: glen.orders,purchasing

Profile added.

netutil=>
```

CHANGE ALIAS

Use the CHANGE ALIAS command to change an alias profile in the AliasDB file on the client.

CHANGE ALIAS prompts you for the current value of the alias name of the alias profile you want to change, and then for a new alias name. If you do not want to change the alias name, press the Return key instead of entering a new name.

The current value of each parameter following the alias name is displayed. You are prompted to enter a new value. To keep the current value, press the Return key.

CHANGE ALIAS can be abbreviated to CA.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> CHANGE ALIAS
Alias Name (current value): AliasName
Alias Name (new value) (opt): AliasName
Database Server Type (current value): DBServerType
DBEnvironment Name on Server (current value): DBEnvironmentName
DBEnvironment Name on Server (new value) (opt): DBEnvironmentName
Server Node Name (current value): ServerNodeName
Server Node Name (new value) (opt): ServerNodeName
Machine Type (current value): MachineType
Machine Type (new value) (opt): MachineType
Datacomm Type (current value): DatacommType
Datacomm Type (new value) (opt): DatacommType
```

```
Change Profile (y/n)? { y[es] }
                      { n[o]  }
```

Parameters

AliasName is the alias name of the DBEnvironment that resides on the server. It is used by the application on the client. This name must be unique on the client and does not need to be the same as the DBEnvironment name. The name can be composed of 16-bit characters or other ASCII or non-ASCII 8-bit characters and can contain up to 128 characters. The HP-UX alias name is case-sensitive; the MPE/iX alias name is not case-sensitive. The alias name is stored and displayed exactly as it was entered.

The first character of this field cannot be an asterisk (*).

DBServerType is ALLBASE/SQL, the database management system server type on the server. (HP SQL is still supported as a database server type. However, if you enter HP SQL, NETUtil displays it as ALLBASE/SQL.) The server type name is not case-sensitive and is stored and displayed in uppercase characters.

DBEnvironmentName is the *absolute* pathname of the DBEnvironment on the HP-UX server and the fully qualified filename of the DBEnvironment on the MPE/iX server. This name follows the file naming conventions of the operating system on which the DBEnvironment resides.

An HP-UX DBEnvironment name cannot exceed 128 characters in length, is case-sensitive, and is localizable (8-bit and 16-bit characters).

An MPE/iX DBEnvironment name cannot exceed 26 characters in length and is not case-sensitive.

The DBEnvironment name is stored and displayed exactly as it was entered.

ServerNodeName is the node name of the server you want to access. The server node name is not case-sensitive and is stored and displayed in uppercase characters.

MachineType is the machine type of the server. Enter HP9000 S700, HP9000 S800, or HP3000 S900. A space is required between HP9000 or HP3000 and the series number. The machine type name is not case-sensitive and is stored in uppercase characters.

DatacommType is the network service interface you are using. The datacomm type is either NS or ARPA. The data communication type name is not case-sensitive and is stored and displayed in uppercase characters. If you are using HP-UX release 10.0 on either the client or server, ARPA is the only valid entry.

Example

In this example, the NS server node name is changed from `system1` to `system2`.

```
netutil=>  change alias
Alias Name (current value): purchdbe2
Alias Name (new value) (opt): Return
Database Server Type (current value): ALLBASE/SQL:
DBEnvironment Name on Server (current value): /users/dbsupport/purchdbe
DBEnvironment Name on Server (current value) (opt): Return
Server Node Name (current value): system1
Server Node Name (new value) (opt): system2
Machine Type (current value): HP9000 S800
Machine Type (new value) (opt): Return
Datacomm Type (current value): ARPA
Datacomm Type (new value) (opt): Return

Change Profile (y/n)? y

Profile changed.

netutil=>
```

CHANGE USER

Use the CHANGE USER command to change a user profile in the NETUsers file on the server.

CHANGE USER can be abbreviated to CU.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> CHANGE USER
Client Node Name(current value): ClientNodeName
Client Node Name (new value) (opt): ClientNodeName
Client Login Name (current value): ClientLoginName
Client Login Name (new value) (opt): ClientLoginName
Server Login Name (current value): ServerLoginName
Server Login Name (new value) (opt): ServerLoginName

Change Profile (y/n)? { y[es] }
                      { n[o] }
```

Parameters

ClientNodeName is the client node name of the user profile you want to change. The client node name is not case-sensitive and is stored and displayed exactly as it was entered.

ClientLoginName is either the login name of the user on the client or an asterisk (*).

The HP-UX client login name follows HP-UX naming conventions for a login name, cannot exceed eight characters in length, and is case-sensitive.

The MPE/iX client login name follows MPE/iX naming conventions for a login name and is not case-sensitive. The session name is not allowed and the group name is required.

Refer to the ADD USER command for details on wildcarding the client login name.

The client login name is stored and displayed exactly as it was entered.

If you enter an asterisk, all client node names in the NETUsers file that match the current client node name are changed to the new client node name.

ServerLoginName is the login name used on the server for the user accessing the DBEnvironment. You are prompted for the server login name only if you enter the client login name.

This name follows HP-UX naming conventions for a login name, cannot exceed eight characters in length, and is case-sensitive. It is stored and displayed exactly as it was entered. The new value must

exist in the /etc/passwd file on the server or you will receive an error message when the application attempts to connect to the remote DBEnvironment. If the server login name is different than the one in the /etc/passwd file, you need to update the /etc/passwd file.

Only one server login name is allowed for each client node name and client login name combination. When possible, the server login name and the client login name should be the same. This avoids problems in some applications that make assumptions about client and server logins. Refer to "Client and Server Login Names" under "Cross-System Connectivity" in the "Getting Started" chapter.

Examples

In this example, one user profile is changed. The client login name of `glen` is replaced with `charly`. The server login name is also changed.

```
netutil=> change user
Client Node Name (current value): clientsystem
Client Node Name (new value) (opt): Return
Client Login Name (current value): glen
Client Login Name (new value) (opt): charly
Server Login Name (current value): glen
Server Login Name (new value) (opt): charly

Change Profile (y/n)? y

Profile changed.

netutil=>
```

In this example, multiple user profiles are changed. The client node name is changed from `clientsystem` to `othersystem`. The current value for client node name is entered, then the new value is entered at the prompt. If an asterisk (*) is entered for the client login name, all user profiles in the NETUsers file that have a client node name of `clientsystem` are changed to `othersystem`.

```
netutil=> change user
Client Node Name (current value): clientsystem
Client Node Name (new value) (opt): othersystem
Client Login Name (current value): *

Profiles qualified: 2

Change all (y/n)? y

Profiles changed.
```

DELETE ALIAS

Use the DELETE ALIAS command to delete one or more alias profiles from the AliasDB file on the client. To prevent you from accidentally deleting an alias profile, you are prompted to confirm the deletion.

DELETE ALIAS can be abbreviated to DA.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> DELETE ALIAS
Alias Name: AliasName

Delete Profile (y/n)? { y[es] }
                      { n[o]  }
```

Parameters

AliasName is either the alias name of the DBEnvironment on the server to be used by the application on the client or an asterisk (*).

If you enter an alias name, the alias profile with the same name is deleted from the AliasDB file.

If you enter an asterisk, you are prompted for the database server type and then the server node name. In this case, all alias profiles that match the database server type and the server node name are deleted.

Examples

In this example, one alias profile is deleted from the AliasDB file.

```
netutil=> delete alias
Alias Name: purchdbe

Delete Profile (y/n)? y

Profile deleted.

netutil=>
```

DELETE ALIAS

In this example, all alias profiles for a specific node are deleted.

```
netutil=> delete alias
Alias Name: *
Database Server Type: ALLBASE/SQL
Server Node Name: system1

Profiles qualified: 2

Delete all (y/n)? y

Profiles deleted.

netutil=>
```

DELETE USER

Use the DELETE USER command to delete one or more user profiles from the NETUsers file on the server. The current client node name and client login name uniquely identify the user profile to be deleted. To prevent you from accidentally deleting a user profile, you are prompted to confirm the deletion.

DELETE USER can be abbreviated to DU.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> DELETE USER
Client Node Name: ClientNodeName
Client Login Name: ClientLoginName

Delete Profile (y/n)? { y[es] | n[o] }
```

Parameters

ClientNodeName	is the node name of the client system.
ClientLoginName	is either the login name of the user on the client or an asterisk (*). If you enter an asterisk for the client login name, all user profiles in the NETUsers file whose client node name matches the current client node name are deleted.

Examples

In this example, one user profile is deleted from the NETUsers file.

```
netutil=> delete user
Client Node Name: clientsystem
Client Login Name: joe

Delete Profile (y/n)? y

Profile deleted.

netutil=>
```

DELETE USER

In this example, all user profiles at the specified node are deleted from the NETUsers file.

```
netutil=> delete user
Client Node Name: clientsystem
Client Login Name: *

Profiles qualified: 2

Delete all (y/n)? y

Profiles deleted.

netutil=>
```

EXIT

Use the EXIT command to terminate execution of the NETUtil program.

EXIT can be abbreviated to E.

Anyone can issue this command.

NETUtil Syntax

```
netutil=> EXIT
```

Example

```
netutil=> exit
```

HELP

Use the HELP command to display and describe all NETUtil commands.

HELP can be abbreviated to H or ?.

Anyone can issue this command.

To leave the HELP command, type //.

NETUtil Syntax

```
netutil=> HELP
Command Name (opt): CommandName
```

Parameters

CommandName is the name or abbreviation of a NETUtil command. If you specify a command name or abbreviation, the correct syntax and a description of that command is displayed. If no name or abbreviation is given, a list of all NETUtil commands is displayed.

Example

```
netutil=> help
The following NETUtil commands are available:

    ADD ALIAS                ADD USER           CHANGE ALIAS
    CHANGE USER              DELETE ALIAS       DELETE USER
    EXIT                     HELP               MIGRATE ALIAS
    MIGRATE ALIAS BACKWARD   MIGRATE USER       MIGRATE USER BACKWARD
    QUIT                     SET ECHO           SHOW ALIAS
    SHOW USER                SYSTEM

For more information on any of these functions, enter the
command name at the prompt.  Pressing a carriage return will
display a brief description of each command.  Type // to leave
the netutil HELP command.

Command Name (opt): Return

Command Summary:
   ADD ALIAS     AA   Adds a profile into the AliasDB file.
   ADD USER      AU   Adds a profile into the NETUsers file.
   CHANGE ALIAS  CA   Changes a profile in the AliasDB file.
   CHANGE USER   CU   Changes a profile in the NETUsers file, or changes
                      all profiles having a given client node name with a
                      new node name in the NETUsers file.
   DELETE ALIAS  DA   Deletes a profile from the AliasDB file, or deletes
                      all profiles with a specified server node name or
                      APPC Session Type from the AliasDB file.
```

```
   DELETE USER   DU   Deletes a profile from the NETUsers file, or deletes
                      all profiles with a specified client node name from
                      the NETUsers file.
   EXIT          E    Terminates execution of the netutil program.
   HELP     H or ?    Displays and describes all netutil commands.
   MIGRATE ALIAS MA   Migrates datacomm definition from NS to ARPA in
                      AliasDB file.
   MIGRATE ALIAS
   BACKWARD      MAB  Changes the AliasDB file back to the state before
                      the most recent saved migration.
   MIGRATE USER       Migrates a specified client node name with an NS
                      node name to an ARPA host name.
   MIGRATE USER
   BACKWARD      MUB  Changes the NetUsers file back to the state before
                      the most recent saved migration.
   QUIT          Q    Terminates execution of the netutil program.
   SET ECHO      SE   Echoes user input to a standard output file.
   SHOW ALIAS    SA   Displays one or more profiles in the AliasDB file.
   SHOW USER     SU   Displays one or more profiles in the NETUsers file.
   SYSTEM         !   Executes a single operating system command.

Command Name (opt): add alias

netutil=> ADD ALIAS
  Alias Name: AliasName
  Database Server Type: DBServerType
  DBEnvironment Name on Server: DBEnvironmentName
  Server Node Name: ServerNodeName
  Machine Type: MachineType
  Datacomm Type: DatacommType

  Adds a profile into the AliasDB file.

  Abbreviation: AA

Command Name (opt): //

netutil=>
```

MIGRATE ALIAS

Use the MIGRATE ALIAS command to migrate alias profiles from NS to ARPA in the AliasDB file on the client. You can choose the automatic option to migrate all profiles or the manual option which prompts you for the server node name of each profile. To use the automatic option, *this command must be used before installing HP-UX 10.0* on the client and/or the server.

If you execute MIGRATE ALIAS VERBOSE, all diagnostic information written to the log file will also appear on the screen. See the description of the log file below.

MIGRATE ALIAS can be abbreviated to MA. MIGRATE ALIAS VERBOSE can be abbreviated to MAV.

You must have superuser capability to execute this command.

NETUtil Syntax

```
netutil=> MIGRATE ALIAS [VERBOSE]

Automatic Migration (y/n)? { y[es] }
                           { n[o]  }

Alias Name : testdb
Current Server Node Name : ServerNodeName
New     Server Node Name (NewServerNodeName) : ServerNodeName

Migrate profile (y/n)?  y
```

Parameters

ServerNodeName — is the name of the server you want to access. You should enter the ARPA host name here. If a value for `(ServerNodeName(ARPA value))` is displayed, you may press Return to select that value.

Description

- MIGRATE ALIAS only targets alias profiles where client and/or server is a HP9000 S700 or S800 and the datacomm type is NS. In addition, MIGRATE ALIAS will alert you if any alias profiles specify a machine type of HP 9000 S300 or S400. The use of a HP 9000 S300 and S400 as a server is not supported in the ALLBASE/SQL G.0 release.
- MIGRATE ALIAS prompts you to select a migration option, either automatic or manual:
  - automatic - All targeted alias profiles will be changed automatically. The datacomm type will be changed from NS to ARPA and the NS node name entry for *Server Node Name* will be changed to an ARPA host name. To use MIGRATE ALIAS with the automatic option, do the following:
    1. Do the migration before installing HP-UX 10.0 on the client and/or server, otherwise only manual migration can be used.
    2. Install patch release phne_2288 and phne_3502 on HP-UX.
    3. Make sure remote loopback is enabled on all servers.

- manual - For each targeted alias profile, the alias name and current server node name will be displayed. A prompt will appear for the new server node name, with the ARPA host name in parentheses (if it can be determined by NETUtil). You may enter an ARPA host name at the prompt, or if the ARPA host name appears, you may press the Return key instead. The datacomm type will automatically change from NS to ARPA.

- When all migration completes, totals for successful and unsuccessful migration attempts are displayed. Finally, you will be prompted to save the migration changes.
- When the migration commands are used, two files are created or appended to on the client: a migration log file, named /tmp/hpdamig.log, and a server list file, named /etc/dausf.mig. The migration log file contains the following information for each profile:

  - a timestamp
  - the command name MIGRATE ALIAS
  - the alias name
  - current server node name (NS value)
  - new server node name (ARPA value)

  The server list file contains a list of affected servers. For each server, use MIGRATE USER or CHANGE USER to migrate the client from an NS value to an ARPA value.

Examples

In this example, MIGRATE ALIAS is executed and the automatic option is chosen. A migration error occurs during migration.

```
netutil=>  MIGRATE ALIAS
Automatic Migration (y/n)?  y

Migration in progress ...

Cannot migrate alias (testdb2) for server (node1)  (DBERR 28029)

Migration complete.
3 alias(es)     migrated successfully.
1 alias(es) NOT migrated - see screen above or /tmp/hpdamig.log
for error information.

Save migration changes (y/n)?  y

Migration changes saved.

netutil=>
```

In this example, MIGRATE ALIAS VERBOSE is executed and the automatic option is chosen. A migration error occurs during migration.

```
netutil=>  MIGRATE ALIAS VERBOSE
Automatic Migration (y/n)?  y

Migration in progress ...
```

```
   Alias: testdb  Migrated Server : snode.sdomain.sorg -> snode
Cannot migrate alias (testdb2) for server (node1.dom.org) (DBERR 28029)
   Alias: testdb3  Migrated Server : hana.dom.org -> hana1.site1.co1.domain1
   Alias: testdb4  Migrated Server : merc.dom.org -> merc2

Migration complete.
3 alias(es)     migrated successfully.
1 alias(es) NOT migrated - see screen above or /tmp/hpdamig.log
for error information.
Save migration changes (y/n)?  y

Migration changes saved.

netutil=>
```

In this example, the manual option is chosen. Two profiles were found; the ARPA host name value for the new server node name could only be determined for the first profile.

```
netutil=>  MIGRATE ALIAS
Automatic Migration (y/n)?  n

Alias Name : testdb
Current Server Node Name : snode.sdomain.sorg
New     Server Node Name (snode) :  Return

Migrate profile (y/n)?  y

Profile migrated.

Alias Name : testdb2
Current Server Node Name : node1.dom.org
New     Server Node Name (snode1) :  newnode

Migrate profile (y/n)?  y

Profile migrated.

Migration complete.
2 alias(es)     migrated successfully.
0 alias(es) NOT migrated - see screen above or /tmp/hpdamig.log
for error information.

Save migration changes (y/n)?  y

Migration changes saved.

netutil=>
```

In this example, the manual option is chosen. The user exits before all migration is completed.

```
netutil=>  MIGRATE ALIAS
Automatic Migration (y/n)?  n

Alias Name : testdb
Current Server Node Name : snode.sdomain.sorg
New     Server Node Name (snode) : Return

Migrate profile (y/n)?  y

Profile migrated.

Alias Name : testdb2
Current Server Node Name : node1.dom.org
New     Server Node Name () :  //

Migration complete.
1 alias(es)     migrated successfully.
0 alias(es) NOT migrated - see screen above or /tmp/hpdamig.log
for error information.
Save migration changes (y/n)?  y

Migration changes saved.

netutil=>
```

MIGRATE ALIAS BACKWARD

Use the MIGRATE ALIAS BACKWARD command to change the AliasDB file back to the state prior to the most recent migration.

MIGRATE ALIAS BACKWARD can be abbreviated to MAB.

You must have superuser capability to execute this command.

NETUtil Syntax

```
netutil=> MIGRATE ALIAS BACKWARD

Migrate AliasDB file backward (y/n)? { y[es] }
                                     { n[o]  }
```

Description

- When you use this command the following information is written to the log file, /tmp/hpdamig.log, on the client machine:
  - a timestamp
  - the command name MIGRATE ALIAS BACKWARD

Example

```
netutil=> MIGRATE ALIAS BACKWARD

Migrate AliasDB file backward (y/n)?  y

AliasDB file migrated backward.
```

MIGRATE USER

Use the MIGRATE USER command to change all user profiles for a given client node name in the NETUsers file on the server. To change individual user profiles based on client login name, use the CHANGE USER command.

MIGRATE USER can be abbreviated to MU. MIGRATE USER VERBOSE can be abbreviated to MUV.

You must have superuser capability to execute this command.

NETUtil Syntax

```
netutil=> MIGRATE USER [VERBOSE]
Current Client Node Name : CurrentClientNodeName
New     Client Node Name (ClientNodeName(ARPA value)) : NewClientNodeName

Migrate All Profiles (y/n)? { y[es] }
                            { n[o]  }
```

Parameters

CurrentClientNodeName	is the client name for the user profile(s) you want to change.
NewClientNodeName	is the new client name for the user profile(s) you want to change. You should enter the ARPA host name here. The client node name is not case-sensitive and is stored and displayed exactly as it was entered. If a value for `(ClientNodeName(ARPA value))` is displayed, you may press Return to select that value.

Description

- If you execute MIGRATE USER VERBOSE, all diagnostic information written to the log file will also appear on screen. See the description of the log file below.
- When all migration completes, totals for successful and unsuccessful migration attempts are displayed. Finally, you will be prompted to save the migration changes.
- For each changed profile, a migration log file, /tmp/hpdamig.log, is appended to on the server and contains the following information:
  - a timestamp
  - the command name MIGRATE USER
  - the user name
  - current client node name (NS value)
  - new client node name (ARPA value)
  - login client information

Example

In this example, the MIGRATE USER command is issued twice; first with the VERBOSE option, second without. The ARPA host name value for the new client node name could only be determined for the first profile.

```
netutil=> MIGRATE USER VERBOSE
Current Client Node Name: cnode.cdom.corg
New     Client Node Name (cnode.cdom1.cdom2.edu) : Return

Migrate All Profiles (y/n)?  y


Migration in progess ...

   Migrated Client : rosanne.dbms.hp -> rosanne2
        Login Name : tester
        Login Name : user1

Migration complete.
2 user(s)     migrated successfully.
0 user(s) NOT migrated - see screen above or /tmp/hpdamig.log
for error information.

Save migration changes (y/n) ?  y

Migration changes saved.

netutil=> MIGRATE USER
Current Client Node Name :  mach1.cdom.corg
New     Client Node Name () :  hpmach1

Migrate All Profiles (y/n)?  y

Migration in progess ...


Migration complete.
1 user(s)     migrated successfully.
0 user(s) NOT migrated - see screen above or /tmp/hpdamig.log
for error information.

Save migration changes (y/n)?  y

Migration changes saved.

netutil=>
```

MIGRATE USER BACKWARD

Use the MIGRATE USER BACKWARD command to change the NETUsers file back to how it was prior to the most recent migration.

MIGRATE USER BACKWARD can be abbreviated to MUB. You must have superuser capability to execute this command.

NETUtil Syntax

```
netutil=> MIGRATE USER BACKWARD

Migrate NETUsers file backward (y/n)? { y[es] | n[o] }
```

Description

- When you use this command the following information is written to the log file, /tmp/hpdamig.log, on the client machine:
  - a timestamp
  - the command name MIGRATE USER BACKWARD

Example

```
netutil=> MIGRATE USER BACKWARD

Migrate NETUsers file backward (y/n)?  y

NETUsers file migrated backward.
```

QUIT

Use the QUIT command to terminate execution of the NETUtil program.

QUIT can be abbreviated to Q.

Anyone can issue this command.

NETUtil Syntax

```
netutil=> QUIT
```

Example

```
netutil=> quit
```

SET ECHO

Use the SET ECHO command to echo user input to a standard output file. To turn ECHO on, type SET ECHO; to turn it off, type the command again. The SET ECHO command is useful for checking the redirected output files on HP-UX.

SET ECHO can be abbreviated to SE.

Anyone can issue this command.

NETUtil Syntax

```
netutil=> SET ECHO
```

Examples

Here is a command file that includes the SET ECHO command:

```
$ list cmdfile

 set echo
 show user
 clientsystem
 *
 quit
```

In this example, NETUtil is invoked and the command file shown above is executed. Output is always directed to a standard output file unless it is redirected to another output file.

```
$ netutil<cmdfile>out
```

The banner is displayed followed by the output from the command file. Here are the results of using this example command file.

```
netutil=> show user
Client Node Name: clientsystem
Client Login Name: *

Client Node Name : clientsystem
Client Login Name: ann
Server Login Name: ann

Client Node Name:  clientsystem
Client Login Name: charly
Server Login Name: charly

netutil=> quit
```

SHOW ALIAS

Use the SHOW ALIAS command to display the contents of one or more alias profiles in the AliasDB file on the client.

SHOW ALIAS can be abbreviated to SA.

Anyone can issue this command.

NETUtil Syntax

```
netutil=> SHOW ALIAS
Alias Name: AliasName
```

Parameters

AliasName is either the alias name of the DBEnvironment that resides on the server, or an asterisk (*).

If you enter an alias name, the contents of that alias profile is displayed. If you enter an asterisk, the contents of all alias profiles in the AliasDB file are displayed.

Examples

In this example, one alias profile in the AliasDB file is displayed when the alias name purchdbe3 is entered for *AliasName*.

```
netutil=> show alias
Alias Name: purchdbe3

Alias Name: purchdbe3
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: /users/dbsupport/purchdbe3
Server Node Name: system1
Machine Type: hp9000 s800
Datacomm Type: ARPA

netutil=>
```

When an asterisk is entered for the alias name, all alias profiles in the AliasDB file are shown, including an alias profile for a DBEnvironment on an MPE/iX server:

```
netutil=> show alias
Alias Name: *

Alias Name: purchdbe2
Database Server Type:  ALLBASE/SQL
DBEnvironment Name on Server: /users/dbsupport/purchdbe2
Server Node Name: system2
Machine Type: hp9000 s800
Datacomm Type: ARPA

Alias Name: purchdbe3
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: /users/dbsupport/purchdbe3
Server Node Name: system1
Machine Type: hp9000 s800
Datacomm Type: ARPA

Alias Name: InvenDB2
Database Server Type: ALLBASE/SQL
DBEnvironment Name on Server: PartsDB.DBgroup2.Inven2
Server Node Name: snode1.sdomain1.sorg1
Machine Type: hp3000 s900
Datacomm Type: NS
   .
   .
   .
netutil=>
```

SHOW USER

Use the SHOW USER command to display the contents of one or more user profiles in the NETUsers file on the server.

SHOW USER can be abbreviated to SU.

You must have superuser capability to issue this command.

NETUtil Syntax

```
netutil=> SHOW USER
Client Node Name: ClientNodeName
Client Login Name: ClientLoginName
```

Parameters

ClientNodeName is either the node name of a client system or an asterisk (*).

If you enter an asterisk (*) for the client node name, all user profiles in the NETUsers file with the specified client login name are displayed. If you enter an asterisk at both prompts, the contents of the entire NETUsers file are displayed.

ClientLoginName is either the login name of the user on the client node or an asterisk (*).

If you enter an asterisk for the client login name, all user profiles in the NETUsers file with the specified client node name are displayed. If you enter an asterisk at both prompts, the contents of the entire NETUsers file are displayed.

Examples

In this example, all user profiles with the client login name ann are shown.

```
netutil=> show user
Client Node Name: *
Client Login Name: ann

Client Node Name: clientsystem
Client Login Name: ann
Server Login Name: ann

Client Node Name: othersystem
Client Login Name: ann
Server Login Name: ann
```

In this example, all the user names are shown for a particular node.

```
netutil=> show user
Client Node Name: clientsystem
Client Login Name: *

Client Node Name: clientsystem
Client Login Name: ann
Server Login Name: ann

Client Node Name: clientsystem
Client Login name: charly
Server Login Name: charly
```

In this example, all user profiles in the NETUsers file are displayed.

```
netutil=> show user
Client Node Name: *
Client Login Name: *

Client Node Name: clientsystem
Client Login Name: ann
Server Login Name: ann

Client Node Name: othersystem
Client Login Name: ann
Server Login Name: ann

Client Node Name : clientsystem
Client Login Name: charly
Server Login Name: charly
   .
   .
   .
netutil=>
```

SYSTEM

Use the SYSTEM command to escape temporarily to the operating system to execute a single operating system command.

SYSTEM can be abbreviated with an exclamation point (!).

Anyone can issue this command.

NETUtil Syntax

```
netutil=> SYSTEM
Command: SystemCommand
```

Parameters

SystemCommand is the HP-UX command to be executed.

Example

```
netutil=> system
Command: ll

total 5586
-rw-rw-r--   1 guest      guest       10626 Jul 12 11:06 +invfile
-rw-------   1 hpdb       guest      204800 Apr 20 15:20 OrderDF1
-rw-------   1 hpdb       guest      204800 Apr 20 15:20 OrderXF1
  .
  .
  .
netutil=>
```

A

ALLBASE/NET File Names

The following ALLBASE/NET files are supplied as a part of ALLBASE/SQL:

Mode	Owner	Group	File Name	Description
555	hpdb	bin	/usr/bin/netutil	ALLBASE/NET utility program file
4544	root	bin	/usr/bin/hpdaNS	Listener daemon for NS *or*
4544	root	bin	/usr/bin/hpdaARPA	Listener daemon for ARPA

Note The NS listener daemon is not included in the 10.0 version of HP-UX.

The following files are created by the ALLBASE/NET utility program file:

Mode	Owner	Group	File Name	Description
644	root	adm	/etc/daalf	AliasDB file on the client
644	root	adm	/etc/daalf.bak	Copy of AliasDB file on the client
600	root	adm	/etc/dausf	NETUsers file on the server
600	root	adm	/etc/dausf.bak	Copy of NETUsers file on the server

The files /etc/daalf.bak and /etc.dausf.bak are copies of the AliasDB and NETUsers files that are created when you exit NETUtil. Should any errors or failures occur to the AliasDB (/etc/daalf) and NETUsers files (/etc/dausf), you can rename these copies to /etc/daalf and /etc/dausf, respectively. These duplicate files eliminate the need to make backup copies of the AliasDB and NETUsers files before modifying them.

The following files are created by ALLBASE/NET on the server:

Mode	Owner	Group	File Name	Description
775	root	root	/tmp/hpdaNS.log	Listener log file for NS *or*
775	root	root	/tmp/hpdaARPA.log	Listener log file for ARPA
775	root	root	/tmp/hpsqlerr	hpsqlproc error log file

The following files are created by the migration commands:

File Name	Description
/tmp/hpdamig.log	Migration log file
/etc/dasvrlst	Server list file

Glossary

alias name
The unique name on the client that represents a specific DBEnvironment on the server. The alias name is part of each alias profile in the AliasDB file.

alias profile
A set of information that is associated with the alias name for a DBEnvironment. Alias profiles are contained in the AliasDB file on the client. They are created and maintained by the system administrator using NETUtil.

AliasDB file
The AliasDB file contains sets of information, called alias profiles, that are associated with each alias name for a DBEnvironment. The AliasDB file resides on the client. It is created when the first alias profile is entered.

ALLBASE/NET
ALLBASE/NET is a software product that allows an application on one HP-UX or MPE/iX client to access a remote ALLBASE/SQL DBEnvironment on another HP-UX or MPE/iX server. Applications such as ISQL, ALLBASE preprocessors, ALLBASE/Query, ALLBASE/4GL, and Information Access SQL/UX as well as user-written applications, can use ALLBASE/NET to access a remote DBEnvironment.

ALLBASE/SQL
ALLBASE/SQL is an SQL database management system that resides on the HP 9000 Series 700/800 and HP 3000 Series 900.

ARPA services
ARPA (Advanced Research Projects Agency) is the network software interface that enables communication between machines on the same network via TCP/IP.

authority
An authority is a permission granted to a user to access specific objects for specific purposes within an ALLBASE/SQL DBEnvironment. *See also* object.

client
A system that secures the services of another system (a server). With ALLBASE/NET, an application on the client can access a remote DBEnvironment on the server. In some cases, a computer can act as both a client and a server. *See also* server.

CONNECT authority
CONNECT authority allows a user to access a DBEnvironment. A user with CONNECT authority has all table and view authorities that may have been granted to the special user PUBLIC.

database
A structured arrangement of data elements designed for the easy selection of information. In ALLBASE/SQL, a database is a collection of tables, views, and indexes having the same ownership in a DBEnvironment.

DBEnvironment
A collection of files containing one or more ALLBASE/SQL databases, plus other entities that contain information for all databases in the DBEnvironment.

DBEUserID
A login name that is one type of owner of database objects.

HP NewWave Access
The PC software, running in the HP NewWave environment, that works with Information Access SQL/UX on the HP 9000 Series 800 client to connect the PC to local or remote databases.

host name
The "ARPA host name" is used by ARPA Services/BSD IPC. On MPE/iX ARPA resolves host names by attempting to contact domain servers listed in the RESLVCNF.NET.SYS or HOSTS.NET.SYS file. On HP-UX ARPA resolves host names by attempting to contact domain servers listed in the /etc/resolv.conf or /etc/hosts file.

Information Access PC
An application that runs on the HP Vectra or on supported IBM and Compaq PCs. It gives PC users access to ALLBASE/SQL tables by interacting with Information Access SQL/UX on an HP 9000 Series 800 client.

Information Access SQL/UX
An application running on an HP 9000 Series 800 client that gives users running Information Access PC access to ALLBASE/SQL tables.

ISQL (Interactive Structured Query Language)
The interactive interface to ALLBASE/SQL is the tool you use for ad hoc queries as well as for loading and unloading data and other database administration tasks.

module
A group of sections stored in the DBEnvironment when an embedded SQL program is preprocessed or when the ISQL command PREPARE is issued. The module is activated when the program is run or when the EXECUTE command is issued in ISQL.

NS
NS is the network software interface that enables communication between machines on the same network via the TCP/IP protocol.

native language
The language of the DBEnvironment. In ALLBASE/NET and ALLBASE/SQL, the default language is n-computer. PORTNLS is used.

NETUtil
NETUtil is the command-driven utility provided by ALLBASE/NET for creating and maintaining the AliasDB and NETUsers files.

NETUsers file
The NETUsers file contains sets of information, called user profiles, that are associated with each ALLBASE/NET user. The NETUsers file resides on the server. It is created when the first user profile is entered.

NetWare
A networking product originally from Novell that is used in the PC environment. NetWare/iX is Hewlett-Packard Company's port of Novell Inc.'s NetWare for UNIX.

node name
The "NS node name" is used by NS/Net/IPC and defined through NMMGR on MPE/iX or through SAM on HP-UX. NS uses probe, probe-proxy and the network-directory to resolve NS name-to-IP-address. Note that the NETUtil prompts "Server Node Name" and "Client Node Name" do not imply "NS node name."

object
A structure created and stored in an ALLBASE/SQL DBEnvironment. The most common objects are tables, views, indexes, and groups.

owner
A DBEUserID, a group name, or a class name. Ownership applies to database objects such as tables, views, indexes, and authorization groups. The owner can drop the object or transfer it to another owner.

preprocessor
A component of ALLBASE/SQL that converts an embedded SQL program into a source file for input to a compiler in one of four programming languages: C, COBOL, FORTRAN, and Pascal.

server
A system that provides a specific service or services to other systems (the clients). With ALLBASE/NET, the server contains the remote ALLBASE/SQL DBEnvironment that is accessed by the application on the client. In some cases, a computer can act as both a client and a server. *See also* client.

Update Tool
A tool used to install ALLBASE/NET and ALLBASE/SQL on your system. This tool lets you select the products and filesets that you want to install. Once selected, the product files are copied into their target directories.

user profile
A set of information that identifies each ALLBASE/NET user. User profiles are contained in the NETUsers file on the server. They are created and maintained by the system administrator using NETUtil.

Index

H

I

L

M

N

O

P