

# **NETBASE**

**for MPE**

## **Reference Guide**



**610 Newport Center Drive, Suite 1400, Newport Beach, CA 92660**

This manual contains proprietary information which is protected by copyright. The information in this manual is subject to change without notice and does not represent a commitment on the part of Quest Software. The software described in this manual is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of this agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser's personal use without the written permission of Quest Software.

**© 1997 Quest Software. All Rights Reserved.  
Quest™ is a trademark of Quest Software.**

**Quest Software**

610 Newport Center Drive, Suite 1400  
Newport Beach, CA 92660  
USA

Tel. (714) 720-1434 / Fax. (714) 720-0426  
support@quests.com

**Quest Software International Ltd**

Oakmoore Court, Kingswood Road, Hampton Lovett  
Droitwich, Worcs. WR9 0QH  
UK

Tel. +44 (0) 1905 797935 / Fax. +44 (0) 1905 797936  
support@uk.quests.com

NetBase™ is a trademark of Quest Software. Other trademarks and registered trademarks used in this guide are property of their respective owners.

**NetBase Reference Guide**

**Part Number - FBPMR1**

**Software Version - 9.7**

**Guide Version - 9.7**

# T A B L E O F C O N T E N T S

## NetBase Reference Guide

<b>Introduction - Welcome to NetBase .....</b>	<b>i</b>
NetBase Functionality.....	iii
NetBase Performance.....	v
How Does NetBase Work? .....	vii
Other NetBase Services.....	xiv
<b>Chapter 1 - The NetBase Components .....</b>	<b>1-1</b>
NetBase Background Job .....	1-2
NetBase Shell .....	1-4
NetBase Directory.....	1-7
NetBase Control Program .....	1-8
<b>Chapter 2 - Using NBCTRL .....</b>	<b>2-1</b>
NBCTRL Command Summary .....	2-2
Running NBCTRL.....	2-3
Special Command Features.....	2-4
Accessing NetBase Subsystems.....	2-5
Using the NBDISABLE JCW .....	2-6
NBCTRL Commands.....	2-7
ABORT .....	2-7
CHGDIR.....	2-11
CONFIG .....	2-12
DEFER .....	2-13
DIR.....	2-14
DISABLE .....	2-15
ENABLE .....	2-17
ROLL FORWARD.....	2-19
SECURITY.....	2-20
SERVER.....	2-21
SET .....	2-22
SHOW .....	2-25
START .....	2-36
STATS .....	2-39

STOP.....	2-43
TRACE .....	2-46
UNDEFER.....	2-48
<b>Chapter 3 - Using CONFIG.....</b>	<b>3-1</b>
CONFIG Command Summary.....	3-2
Configuring NetBase.....	3-3
Special Command Features.....	3-5
NetBase Parameter Definitions .....	3-6
Operational Parameters.....	3-7
Node Level Parameters .....	3-25
CONFIG Commands .....	3-31
ADD.....	3-31
DELETE .....	3-33
LIST .....	3-34
MODIFY .....	3-38
PRINT .....	3-41
<b>Chapter 4 - Using NBDIR .....</b>	<b>4-1</b>
NBDIR Command Summary .....	4-2
Running NBDIR.....	4-4
Wildcards within NBDIR .....	4-5
Editing a NetBase Directory .....	4-6
Special Command Features.....	4-8
NBDIR Commands.....	4-9
ACCOUNT.....	4-9
BUILD.....	4-11
DEVICE .....	4-12
DIR .....	4-15
DISPLAY .....	4-16
DSET .....	4-17
FILE.....	4-19
GROUP .....	4-24
LIMIT .....	4-25
LISTD .....	4-26
LISTF.....	4-28
LISTL.....	4-32
LISTP.....	4-34
LISTS.....	4-36
LISTSQL.....	4-39
LISTSQLO .....	4-41
LISTX .....	4-43
LOGON.....	4-44
PROGRAM .....	4-46

REBUILD .....	4-50
RESET .....	4-51
SHADOW .....	4-53
SQLIN .....	4-56
SQLOUT .....	4-58
TABLE .....	4-61
UNLOAD .....	4-63
XL .....	4-64
<b>Chapter 5 - NetBase Shadowing .....</b>	<b>5-1</b>
Process Overview .....	5-2
Master System .....	5-2
Shadow System .....	5-4
Configuration Parameters .....	5-5
Defining a File for Shadowing - NBDIR Entries .....	5-8
Examples of Directory Entries for Shadowing .....	5-12
Third Party Shadowing (TPS) for TurboIMAGE databases .....	5-14
Setting Up Shadowing .....	5-16
Monitoring Shadowing .....	5-17
Shadow Files .....	5-19
Checking the State of Shadowed Files .....	5-20
Synchronization Loss .....	5-22
Queue State .....	5-22
Sync or Discard State .....	5-23
NBSYNC Syntax .....	5-28
Structural Changes for Shadowed Databases .....	5-30
1) NetBase ADAGER Batch Interface .....	5-30
2) NetBase DBGENERAL Interface .....	5-31
3) Manually Coordinating the Structural Changes .....	5-31
Flushing Shadow Updates .....	5-33
Backups Using Shadow Data .....	5-36
User Exits .....	5-38
Export User Exits .....	5-38
Post User Exits .....	5-42
Dataset Shadowing .....	5-51
Dataset Shadowing Considerations .....	5-52
Shadowing With TurboIMAGE Log Files .....	5-53
Resetting the Logging Process .....	5-56
Shadowing with Dynamic Rollback Recovery .....	5-57

<b>Chapter 6 - NetBase NFA</b> .....	<b>6-1</b>
NFA Overview .....	6-2
Buffering .....	6-3
Configuring Network File Access .....	6-4
Defining a File for NFA .....	6-5
Network File Access Options .....	6-5
NFA Examples .....	6-6
Using OMNIDEX and NetBase .....	6-8
Monitoring NFA.....	6-9
NBCTRL Commands.....	6-9
Examples.....	6-9
Remote Procedure Calls.....	6-11
NBRPC Procedure Calling Conventions.....	6-12
Remote User Procedure (procname) Calling Conventions .....	6-13
NFA Security.....	6-16
NBSECURE Commands.....	6-16
Running NBSECURE.....	6-17
Special NBSECURE Commands.....	6-17
ACCESS.....	6-18
ACCOUNT.....	6-19
ALLOW .....	6-20
GROUP .....	6-22
LISTSEC .....	6-23
NODE .....	6-24
REMOVE.....	6-25
<b>Chapter 7 - NetBase Spooling</b> .....	<b>7-1</b>
Spooling Overview .....	7-2
Disaster Tolerance.....	7-3
Remote Spooling .....	7-5
Operating Principles .....	7-5
Setting Up Remote Spooling .....	7-6
Using Pseudo Devices with Remote Spooling.....	7-8
A NetBase Spooling Implementation Example.....	7-8
NBSpool Program .....	7-10
NBSpool Features .....	7-10
<b>Chapter 8 - AutoRPM</b> .....	<b>8-1</b>
AutoRPM Overview .....	8-2
Defining a Program for AutoRPM .....	8-3
AutoRPM Example.....	8-3
Notes on Implementing AutoRPM.....	8-5
AutoRPM Options .....	8-6

<b>Chapter 9 - NetBase Statistics .....</b>	<b>9-1</b>
Collecting Statistics .....	9-2
Reporting Statistics.....	9-4
Running NBLGDUMP.....	9-4
Running NBRPT01.....	9-7
Running NBRPT02.....	9-10
Generating Scenarios.....	9-13
Running NBLGTRAN.....	9-14
Scenario File Format.....	9-16
<b>Chapter 10 - NetBase and Third Party Products.....</b>	<b>10-1</b>
NetBase ADAGER Interface.....	10-2
NetBase DBGGENERAL Interface.....	10-4
NetBase OMNIDEX Interface.....	10-5
How the Interface Works.....	10-5
Special Installation Steps for Shadowing.....	10-6
NetBase MPEX Interface.....	10-7
<b>Chapter 11 - System Operations and Maintenance .....</b>	<b>11-1</b>
System Startup.....	11-2
System Shutdown .....	11-3
Partial Backup .....	11-4
Full Backup .....	11-5
HP Patches .....	11-6
FOS Upgrades .....	11-7
NetBase Updates.....	11-8
Removing NetBase .....	11-9
<b>Chapter 12 - Disaster Recovery Procedures for NetBase Shadowing Environments.....</b>	<b>12-1</b>
Case One.....	12-2
Planned Failover.....	12-2
Unplanned Failover.....	12-4
Case Two.....	12-6
Case Three .....	12-7
Planned Failover.....	12-7
Unplanned Failover.....	12-8
Creating Alternate Directories with NBDIR.PUB.NETBASE .....	12-10
<b>Chapter 13 - NetBase Utilities .....</b>	<b>13-1</b>
NBVERS.....	13-2
Enabling/Disabling NetBase via NBVERS.....	13-3
NBLOG.....	13-5
NBTRACE .....	13-8
QUARK.....	13-10

Additional Utilities.....	13-11
QUARK+ .....	13-11
NBCOPY.....	13-15
<b>Appendix A - Installation/Update Procedures.....</b>	<b>A-1</b>
Overview.....	A-2
Before Installing NetBase.....	A-3
Updating From NetBase 9.6 Global Install.....	A-3
Installation Steps.....	A-4
Step 1. Restore NETACCT job and STREAM NETACCT.....	A-4
Step 2. Configure Network Nodes for NetBase.....	A-6
Step 3: Install the NetBase Shell.....	A-10
Step 4. Validate Installation and Modify System Startup Procedures.....	A-12
Implementation Notes.....	A-15
System Startup Procedures.....	A-15
Global Installation.....	A-15
Installation Appendix.....	A-17
Converting From an Account Installation to a Global Installation.....	A-17
Updating From Global Installation Prior to NetBase 9.6.....	A-18
<b>Appendix B - Troubleshooting NetBase.....</b>	<b>B-1</b>
Technical Support Tips.....	B-2
SHOW POST.....	B-2
NetBase Versions.....	B-4
NFA.....	B-6
NBSYNC, VERIFY.....	B-8
MPE Builds, Purges and Renames.....	B-9
NetBase Error Messages.....	B-10
File System Errors.....	B-11
Directory File (NBD) Related Errors.....	B-16
Network Configuration File (NBI) Related Errors.....	B-17
Network Control File (NBN) Related Errors.....	B-18
Shadowing Queue file (NBM) Related Errors.....	B-19
Statistics Logging File (NBVL) Related Errors.....	B-20
Spooling Files (NBSEC and NBSED) Related Errors.....	B-21
AutoRPM Facility Detected Errors.....	B-22
NetBase Control Program (NBCOP) Detected Errors.....	B-25
Spooling Processes Detected Errors.....	B-26
NBSERVER Detected Errors.....	B-27
NMSERVER Detected Errors.....	B-28
Shell Detected Errors.....	B-30

## INTRODUCTION

# Welcome to NetBase

NetBase is comprehensive networking software consisting of five major services. The NetBase services allow a network of Hewlett-Packard commercial computers to work together acting as one large system. The NetBase services are:

- **NetBase Shadowing**
- **Network File Access**
- **NetBase Spooling**
- **NetBase AutoRPM**
- **NetBase Statistics**

Two other NetBase services are available: **NetBase SQL Shadowing**, a NetBase module that allows the shadowing of Allbase databases; and **NetBase Client**, an environment-specific service that allows PC applications and UNIX systems to access data on Hewlett-Packard mainframes. These services are briefly described later in this chapter but are not referenced in this guide. For reference information on these services, see the *NetBase SQL Shadowing Reference Guide* and the *NetBase Client Reference Guide*.

## In This Guide

In this guide, you will find the following chapters:

**Chapter 1** explains the NetBase components.

**Chapter 2** describes the **NBCTRL** commands.

**Chapter 3** describes the **CONFIG** commands.

**Chapter 4** describes the **NBDIR** commands.

**Chapter 5** explains NetBase Shadowing.

**Chapter 6** explains Network File Access.

**Chapter 7** explains NetBase Spooling.

**Chapter 8** explains NetBase AutoRPM.

**Chapter 9** explains NetBase Statistics.

**Chapter 10** discusses NetBase and third party products.

**Chapter 11** provides system operations and maintenance procedures.

**Chapter 12** provides disaster recovery procedures for NetBase shadowing environments.

**Chapter 13** describes the NetBase utilities.

An appendix on each of the following subjects is located at the back of this manual.

**Appendix A** provides instructions for global installation.

**Appendix B** provides troubleshooting information.

## NetBase Functionality

NetBase brings functionality and performance to HP systems that previously was not possible. Customers faced with expensive upgrades have found that adding a machine and using NetBase to distribute the computing load is less costly and provides better performance. Companies that own multiple systems may use NetBase to correct an imbalance in processor loading. In a situation in which one machine is over-utilized while others are under-utilized, NetBase provides load leveling techniques to balance the work load across all available machines. In addition, users can build a fault-tolerant environment without any changes to applications.

## Saving on Distribution Performance

Using NetBase, an application designed to run on a single computer can easily be distributed across multiple computers without modifying the application software. NetBase maintains a directory of all files and databases available to network users. When an application program accesses a file or database, NetBase checks its directory. If the file is found to reside on another computer, NetBase routes the requests to the appropriate machine.

The key to a successful networking solution is performance. Distributing an application across a network has a limited advantage if the cost outweighs the processing power of the additional computers. NetBase is designed with features that minimize overhead yet maximize throughput.

## Maintaining Multiple Copies

Maintaining multiple copies of files and databases can be useful for a variety of reasons. If 24-hour availability is important, NetBase can maintain concurrent copies of the data. During a backup, only the shadowed data is taken off-line while the master copy remains available for the application to use. Once a backup has finished, NetBase updates the shadowed copy with any changes that have occurred to the master in the interim.

Other reasons for maintaining multiple copies are heavy read activity and disaster recovery. If some files are heavily read from a number of computers in the network, it makes sense to maintain local copies to reduce networking traffic. NetBase uses the local copy for read type activity, and with NFA installed, automatically switches to the master copy when the application requires update access. NetBase ensures that all copies are kept in synchronous order. Disaster recovery is simplified if multiple copies of data exist on the network. If a machine goes down, you can redirect file access through the NetBase directory to get an application back online in a short period of time.

NetBase has been designed to achieve, simplify, and streamline all of your networking needs. NetBase allows you to obtain the most out of your hardware and software resources without having to change current applications.

## Transporting Spooling Output Efficiently

One desirable feature of networked computers is the ability to share printing resources. NetBase minimizes both networking and spooling overhead by efficiently transporting the output to the desired computer and spooling the output on the destination computer only. NetBase Spooling can be as fast as local spooling.

## NetBase Performance

NetBase is fast! Of all the features that NetBase offers, performance is perhaps the most valuable. Networking a transaction is more costly in terms of elapsed time and machine resources than performing the same transaction locally. The key to enhanced performance is that NetBase minimizes the number of network transactions.

### Minimizing Network Transactions

When NetBase was developed, great care was taken to ensure that the code was written as efficiently as possible. However, the main reason that NetBase is efficient is Quest's approach in implementing network file access. The major cost associated with accessing remote files results from the overhead incurred transporting the request and result between computers. Most of this additional overhead comes not from the actual transfer of data across the LAN, but is incurred in the preparation of the data to be transported and the handling of the data on the other side of the line. Sending 1000 bytes of data requires about the same overhead as sending 10 bytes.

To illustrate this point, consider the following: A typical DBGET on an unloaded Series 947 takes less than 1 millisecond. The difference in elapsed time is the overhead required to prepare and route the request across the LAN link and return the result to the requesting process. Keeping in mind that most of the overhead was incurred preparing the transaction and not transferring the data, a large performance benefit lies in getting the most out of each network transaction. For example, if the number of network transactions can be cut in half, the throughput will be almost doubled.

NetBase utilizes two techniques for reducing the number of network transactions. Anticipatory reading by NetBase yields the highest performance savings. NetBase also piggybacks calls together whenever possible to further reduce the number of network transactions. Some calls lend themselves to being serviced by both techniques.

## Improving Data Access

Application programs usually access data in a very predictable manner. Generally speaking, a DBFIND to an IMAGE dataset is followed by one or more mode 5 or 6 DBGETs. This predictability is true for a variety of other calls. NetBase makes use of these facts when servicing calls to remote files and databases.

When NetBase services a call to a remote file, the results of that and any anticipated subsequent calls to that file are returned to the requesting user's NetBase buffering area. Only the results of the original call are returned to the user. Subsequent calls to that file can now be serviced locally using the buffered data until data outside the buffer is requested or the user performs an activity that requires flushing the buffer.

The performance increase is dependent upon the application. Increases can range up to 10 or more times faster than RFA. With NetBase, printing to a remote spooled device is accomplished in a fraction of the time and consumes fewer valuable system resources.

## How Does NetBase Work?

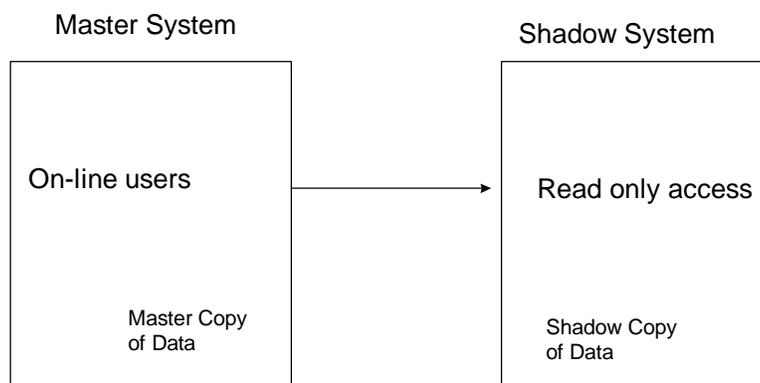
NetBase works by intercepting and looking at all calls and file requests to the file system. Requests to local files are passed directly to MPE to be executed locally, while requests to remote files are passed by NetBase to the appropriate computer where the file resides. To determine where a particular file exists, NetBase uses its own directory. The directory is the key to the simplicity of NetBase implementation.

Specifying the location of networked files is accomplished using the NetBase **NBDIR** utility program. To distribute a file or database on the network, you enter the file or database name and its location in the NetBase directory. A one-line command is all that is required to redirect access to a file or database. Directory changes take effect immediately, and affect all subsequent access to the file. Locally accessed files do not need to be entered into the directory.

The following discussions are of each of the NetBase services.

## NetBase Shadowing

Integrated into NetBase is the ability to maintain concurrent multiple copies of files and databases. Keeping multiple copies of data can be useful for a variety of reasons. A local copy of production data can dramatically reduce I/O contention against the master copy. The production system can remain online 24 hours a day if shadowed copies of files and databases are used for backup. Disaster recovery is greatly simplified when current up-to-date copies are always ready in the event of a disaster.



### A Different Approach

NetBase approaches shadowing differently from other shadowing products in several ways. First, since NetBase does not use IMAGE logging, all types of disk files are shadowed, not just databases. One product can now be used to maintain a complete copy of all application files. NetBase Shadowing also guarantees that updates occur to the shadow copy in the exact order that they occurred against the master.

Second, the process that NetBase uses to move data around the network reduces networking overhead to a bare minimum. Based upon the needs of the particular application environment, the NetBase shadow process can be configured to minimize the number of network transactions required to transport updates to the other computer.

Third, if NetBase NFA (Network File Access) is being used, shadowing is integrated with NFA and NetBase automatically decides when to use the shadow copy and when to use the master copy. NetBase operates on the principle of master and shadow copies of files and databases. Each shadow file has a master copy to which all updates for the file are performed. The updates then are performed on the shadow copies. By using this technique, NetBase can guarantee that all copies will be exact duplicates of each other.

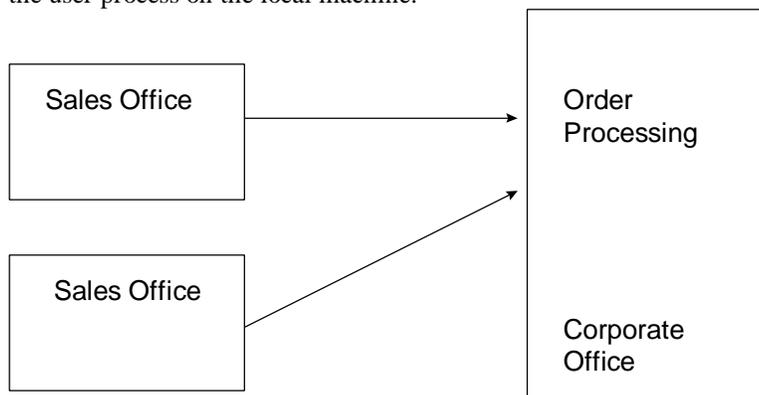
### Shadowing Functionality

NetBase Shadowing is designed to function during all modes of operation and failure. It functions when posting to shadow copies is interrupted during system backups and during hardware failures. If the shadow computer is unavailable, the NetBase export process automatically queues all updates on the master computer until the shadow machine becomes available. The posting process also queues changes to files or databases that are unavailable. NetBase Shadowing features provide you with a great deal of flexibility, greatly increasing the functionality of your system and eliminating unnecessary re-synchronizations.

## NetBase NFA

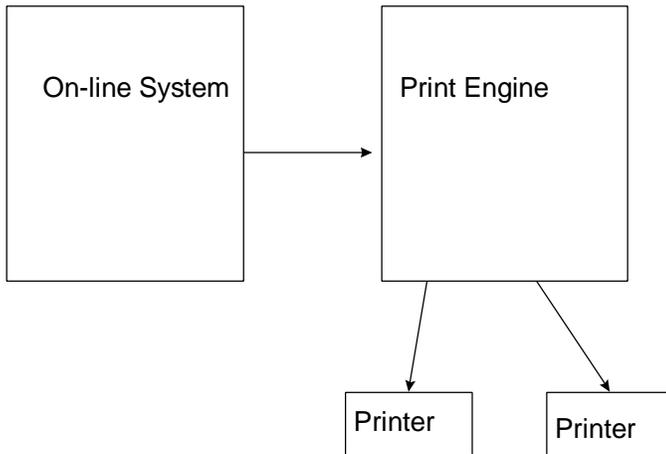
The NetBase product is built around Network File Access (NFA). NFA allows programs running on one computer to access files in the domain of another computer. With NetBase NFA, a file can be placed anywhere in the network without having to change programs, file equations or UDCs. Additionally, NetBase does not require any type of remote sessions or virtual terminals, and it does not count against the OS user license limit on the remote system.

The NetBase Shell intercepts all calls to the file and database systems. The NetBase Shell is a set of procedures with the same names and parameter lists as their MPE counterparts. Depending on the location of the file as specified in the directory, NetBase passes the calls either to MPE or to a server on the remote machine. The server process performs the call and the results are returned to the user process on the local machine.



## NetBase Spooling

Network spooling is a standard NetBase feature. NetBase Spooling provides several transparent methods for defining the printing environment and uses a minimal amount of overhead. NetBase Spooling allows users to move or copy selected spool files from one computer to another. Users are provided with flexible syntax for specifying spool files to move, making a simple task of directing an output file to the appropriate device. This process can also be used to automatically transfer a set of spool files as they are created.



### Transporting Multiple Output Lines

Creating a local spool file and copying it across the network to a remote spooler increases the amount of spooling overhead. NetBase Spooling transports multiple lines of output per network transaction, instead of one line. Blocking the output greatly reduces networking overhead, increasing the performance across the LAN.

### Bypassing the Local Spool File

Unlike other third party networking spooling solutions, NetBase does not create the spool file on the local computer. Since NetBase is positioned between the application and the file system, it can send the report directly to the destination computer. This removes the need for a local copy, eliminating unnecessary disk I/Os. The result is a simpler, more efficient method of sharing remote spooled devices. In fact, program output to a remote spool device using NetBase is less impeded than if it was outputting to the local MPE spooler.

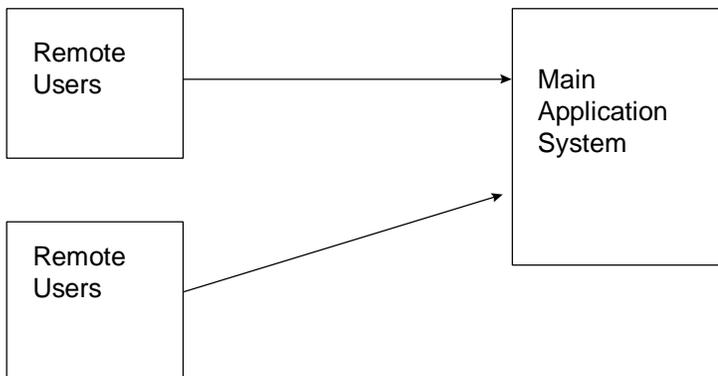
### Creating a Local Copy

Another NetBase feature is that any spooled device can be configured to create a local copy of the output automatically. This extra copy can be used as a backup copy, or it can be deleted by NetBase once the entire spool file has been transmitted. If a network failure should occur while a process is outputting, and the remote spooler is defined with the **COPY** option, NetBase will allow the process to continue using the local spool copy.

## AutoRPM

AutoRPM is the NetBase interface to Remote Process Management (RPM), a feature of Hewlett-Packard's Network Services. AutoRPM provides the ability to run an interactive program on a remote computer. NetBase AutoRPM is completely transparent to the user. The user is not aware that the program is actually running on another machine.

AutoRPM provides two major financial values. As an option for load balancing and reduced processing costs, NetBase AutoRPM saves you money. Additionally, in reference to software licensing for specific machines, NetBase AutoRPM may eliminate the need to purchase multiple copies of application software.



## Automatic Transport

NetBase AutoRPM also makes running remote programs an easy task for users. Users simply request to run a program. NetBase AutoRPM does the rest: opening a line to the appropriate remote computer; issuing a remote logon command; and issuing a remote run command for the program. NetBase AutoRPM transports the session and file equations to the remote machine so that they affect the actual running environment. As a result, NetBase AutoRPM does more automatically than users do when remotely running programs.

## NetBase Statistics

Using NetBase Statistics, application systems can be easily profiled, monitored and tuned. NetBase provides a method of capturing statistics on file access as well as measuring process overhead. NetBase also provides reports that clearly illustrate the captured information. The accumulation of data is a by-product of running applications through NetBase. The collection of statistics can be enabled or disabled at any time for any subset of data required.

Minimal overhead is incurred when statistics are enabled. Unlike other profiling products, NetBase only logs the statistics to disk when the file is closed or the process terminates. Using these techniques, NetBase is able to capture accurate information in such a way that the overhead does not significantly distort the element that is being measured.

Any network strategy can be modeled using statistics gathered from the current environment, standard NetBase reports, and the Scenario Generator. The statistical data provides information on file and program activity, the reports display the information, and the Scenario Generator is used to manipulate the statistics to reflect the desired configuration. See the following statistics example.

				-----READS-----								-----UPDATES-----			-----OTHER-----			-----OVERALL-----		
File	Node	File Name	Mode	OPENS	Count	Avg Wall	msec CPU	Count	Avg Wall	msec CPU	Count	Avg Wall	msec CPU	Count	Avg Wall	msec CPU				
SUN, JAN 10, 1988, 4:05 PM																				
NetBase - QUEST Software																				
Network Statistical Analysis - File Access by Program																				
Samplings from: THU, JUL 16, 1987, 3:58 AM																				
thru: THU, JUL 16, 1987, 6:31 AM																				
NBRPT01 [0.7.0]																				
Page: 1																				
-----READS-----																				
-----UPDATES-----																				
-----OTHER-----																				
-----OVERALL-----																				
MENUMPROG.PUB.AAA (TEST)																				
Local	TSMENU	PUB	CEI	67	1486	3	2	0	0	0	272	20	10	1758	5	4				
Local	MENUCONF	DATA	CEI	67	170	2	1	0	0	0	205	29	20	375	16	12				
Local	IDEAS	DATA	CEI	33	33	29	25	0	0	0	66	248	90	99	175	69				
Local	TSPROC	JOB	CEI	1	11	2	1	0	0	0	3	20	14	14	6	4				
Local	SPLYMENU	PUB	CEI	1	0	0	0	0	0	0	5	21	11	5	21	11				
Local	MENUACCT	DATA	CEI	1	1	16	4	0	0	0	2	140	18	3	98	13				
				170	1701	3	3	0	0	0	553	51	24	2254	15	8				
TSPOLL.PUB.CEI (TEST)																				
Local	TFILE	POLL	CEI	30	0	0	0	2315	6	6	60	31	16	2375	6	6				
Local	RFILE	POLL	CEI	30	0	0	0	2070	2	2	60	38	19	2130	3	2				
Local	ANSPOS	DATA	CEI	38	1749	6	4	98	146	77	164	329	35	2011	39	10				
Local	SLUALL	POLL	CEI	30	30	4	3	0	0	0	90	16	12	120	13	10				
				128	1779	6	4	4483	7	5	374	159	24	6636	15	6				
TSPROC.PUB.CEI (TEST)																				
Local	IDEAS	DATA	CEI	2	27649	27	6	7384	12	7	15979	1	1	51012	17	4				
CORP	RETAIL	DATA	CEI	2	11192	35	5	7402	200	55	17352	1	1	35946	53	13				
Local	TSPRH	DATA	CEI	2	2636	1	1	1317	4	4	7916	8	2	11869	6	2				
Local	SERIAL	DATA	CEI	2	0	0	0	0	0	0	2632	2	1	2632	2	1				
Local	ZEROS	DATA	CEI	2	0	0	0	659	10	3	1324	6	2	1983	7	2				
CORP	SLSMAN	DATA	CEI	2	1321	25	3	0	0	0	2	708	113	1323	26	3				
Local	FIXB	DATA	CEI	2	2	2	2	11	206	53	22	72	13	35	110	25				
Local	TRANSFL1	DATA	CEI	1	1	6	6	0	0	0	5	19	9	6	17	9				
Local	TSPRL	DATA	CEI	1	0	0	0	0	0	0	5	8	7	5	8	7				
Local	TSPRM	DATA	CEI	1	0	0	0	0	0	0	5	8	8	5	8	8				
Local	TSPRP	DATA	CEI	1	0	0	0	0	0	0	5	8	8	5	8	8				
Local	TSPRS	DATA	CEI	1	0	0	0	0	0	0	5	8	7	5	8	7				
				19	42801	27	5	16773	94	28	45252	3	1	104826	27	7				
Utilization by Mode (%)				Lcl	03															
				Count	64	36														
				Wall	33	67														
				CPU	34	66														
****Subtotal for Mode****				388	46575	26	5	21256	76	23	46354	5	2	114185	26	7				

## Other NetBase Services

Two other NetBase services are available: NetBase SQL Shadowing and NetBase Client. These services are briefly described below but are not referenced in this guide. For reference information, see the *NetBase SQL Shadowing Handbook* and the *NetBase Client Reference Guide*.

### NetBase SQL Shadowing

In addition to the NetBase Shadowing service designed for IMAGE databases and KSAM files, NetBase provides shadowing of ALLBASE DBEs. With NetBase SQL Shadowing, updates are applied to shadow copies in the same order as they were applied to the master copy of the DBE. If communication between the master and the shadow copies is interrupted, when the communication is resumed, NetBase SQL Shadowing determines where it was in the update process, and resumes where it left off.

You may shadow entire ALLBASE DBEs or you may shadow specific tables. Depending on which list is shorter, you may define the tables to be excluded from or included in the shadowing process. Using the same table specification logic, you may create partitioning, allowing different tables to be masters on different machines. By developing vertical partitioning, you reduce the possibility of the DBE losing synchronization.

#### Additional Processing

With NetBase SQL Shadowing, you are allowed additional processing before and after the network transaction is sent from the master to the shadow machines. User exit procedures may be invoked prior to sending the transaction to the remote nodes, prior to updating the DBE on the remote nodes, or after an update on a remote node has failed to be applied. NetBase SQL Shadowing reads the NetBase directory entries to determine which DBEs to shadow, which tables to include and if user exits are to be invoked.

User exits may be used to re-route or ignore data prior to sending the transaction across the network or after it has been received on a shadow node. A user exit may be used to log transactions without actually sending the data to remote nodes by specifying node zero, a “pseudo-node.” Another possibility for a user exit is to notify the console when a transaction was not successfully applied.

For more information on NetBase SQL shadowing, see the *NetBase SQL Shadowing Handbook*.

## NetBase Client

NetBase Client allows MS-DOS, Windows, and UNIX based applications to access MPE files and IMAGE databases on HP 3000 MPE/iX systems. To offer this access, NetBase Client uses libraries of APIs that have the same calling sequences as the MPE/iX intrinsics. Code that is written and compiled on an HP 3000 may, with minor modifications, be compiled and run on a PC or a UNIX system.

As a result, each intrinsic call from a PC or UNIX system is passed to the MPE/iX machine, executed, and the data returned. Because NetBase Client supports Microsoft Windows, advanced easy-to-use development environments such as Microsoft Visual Basic can be used for developing client/server type applications. Using NetBase Client also alleviates the need to learn a new programming paradigm such as SQL.

Performance is one of the major benefits when using NetBase. NetBase Client avoids the time-consuming and inefficient translations between a myriad of “standards” such as SQL, ODBC, and ALLBASE. NetBase Client is commonly 10 to 100 times faster than the HP ODBC interface to Image/SQL when accessing an IMAGE database.

For more information on NetBase Client, see the *NetBase Client Reference Guide*.

## CHAPTER 1

# The NetBase Components

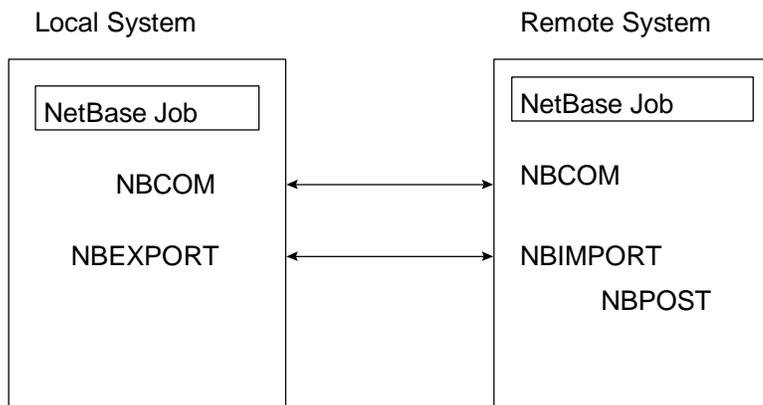
Four NetBase components exist that support the NetBase services. These components are: the NetBase background job to facilitate communication between systems; the NetBase shell to intercept file and database activity; the NetBase directory to define the files for NetBase services; and the NetBase control program to monitor and configure NetBase.

### In This Chapter

- **NetBase Background Job**
- **NetBase Shell**
- **NetBase Directory**
- **NetBase Control Program**

## NetBase Background Job

The NetBase background job is used to establish communication between all nodes configured within NetBase. The main purpose of the job is for transportation of data and communication between nodes. The NetBase job must be up and running at all times on the system after NetBase is installed.



Start the NetBase job as part of the normal system startup procedures. The **NETBASE.JOB.NETBASE** file is the actual background job that is streamed on the system. You can start the job using **NBCTRL** or by streaming the file.

### To start the NetBase job using NBCTRL:

1. Use **NBCTRL** to start the job by typing:

```
:NBCTRL.PUB.NETBASE;INFO="START NETBASE"
```

### To start the NetBase job by streaming the file:

1. Stream the file by typing:

```
:STREAM NETBASE.JOB.NETBASE
```

The NetBase background job runs a program called **NBCOP.NB.NETBASE** that starts all of the configured NetBase services. The NetBase background job reports all status and warning messages to the system console, the **stdlist** of the NetBase background job, and the **NBLOG.DATA** logfile. Each message in the **stdlist** is formatted with the Julian date and a timestamp (military time).

The **NBLOG.DATA** logfile contains all of the messages from the previous run of the NetBase job. To view the messages from the logfile, use the **NBLOG.PUB.NETBASE** utility program.

**To view logfile messages:**

1. View logfile messages by typing:

```
:RUN NBLOG.PUB.NETBASE
```

2. Enter the start date:

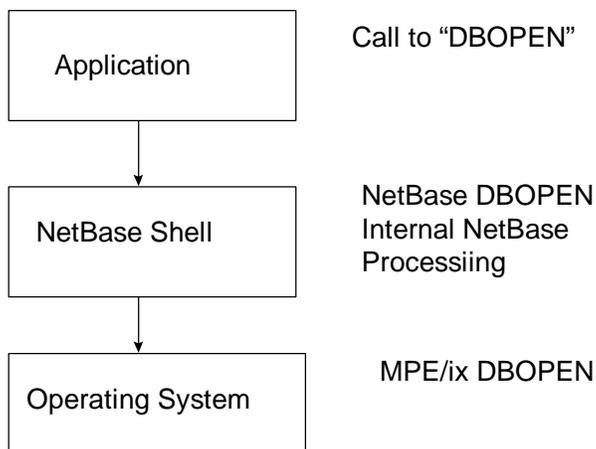
```
Enter Start Date <cr> for all: 12/13/96
```

```
12/20/96 15:14:22.9 NBN/Log File Validated
12/20/96 15:14:23.0 NBN/NetBase Control Process [0.9.7] Update 7
12/20/96 15:14:25.5 NBI007/NETBASE Shadow Importing for BUGS STARTED
12/20/96 15:14:26.6 NBN/NETBASE Local Access STARTED
12/20/96 15:14:26.6 NBN/NETBASE Remote Access STARTED
12/20/96 15:14:26.6 NBN/NETBASE Ready
12/20/96 15:14:28.8 NBN/COMMUNICATION READY TO NODE 7: BUGS
```

## NetBase Shell

The NetBase shell is a set of procedures that install into the system libraries. These procedures intercept TurboIMAGE and file system intrinsic calls. In fact, the NetBase shell procedures have the same name as the MPE/iX procedures. This allows NetBase to trap the activity to files or databases without having to make any changes to application programs. The NetBase shell traps programmatic activity to files or databases. It does not intercept MPE commands such as **BUILD**, **RENAME** or **COPY**. There is a special UDC that can be enabled for this. For more information on this UDC, turn to Appendix A “Installation/Update Procedures”.

For example, when an application program passes a **DBOPEN**, it is intercepted by the NetBase shell. The NetBase shell then does a lookup in the NetBase directory to check if this database is to be shadowed or accessed on another system. If no directory entry exists for this database in the NetBase directory, the NetBase shell passes the **DBOPEN** call to the real **DBOPEN** on the system for processing. If a directory entry exists for the database to access it on a remote system via NFA, the NetBase shell sets up the necessary processing to connect this user to the remote system.



The NetBase shell does a lookup in the NetBase directory for every file and database open that occurs on the system. If no directory entry exists, the request is passed to the MPE/iX operating system. If an entry exists, the NetBase shell does the necessary processing to either shadow the file or access it remotely, and then passes the request to MPE/iX.

The interception by the NetBase shell is completely independent of the NetBase background job. Once the shell is installed into the system libraries, the interception begins. Remember that the NetBase background job is used for transportation and communication.

When the NetBase shell is installed in the system libraries, new error codes may occur. These new error codes are greater than **1000**, and constitute some type of NetBase error. For example, if you were running an application program that passes a **DBOPEN** on a database that is to be accessed remotely via NFA but the local NetBase background job was not running, the **DBOPEN** would fail with **1001 - LOCAL NETBASE SERVICES ARE DOWN**. For a list of NetBase error codes and messages, turn to Appendix B “Troubleshooting NetBase”.

The following tables list the procedures that comprise the NetBase shell. The procedures include file system, IMAGE and other intrinsics.

### File System Intrinsics

AIFCLOSE	FGETKEYLOC	FRENAME
FOPEN	FLABELINFO	FSETMODE
FCHECK	FLOCK	FSPACE
FCLOSE	FPOINT	FUNLOCK
FCONTROL	FREAD	FUPDATE
FDELETE	FREADBYKEY	FWRITE
FERRMSG	FREADC	FWRITEDIR
FFILEINFO	FREADDIR	FWRITELABEL
FFINDBYKEY	FREADLABEL	HPERRMSG
FFINDN	FREADSEEK	HPFOPEN
FGETINFO	FRELATE	
FGETKEYINFO	FREMOVE	

### IMAGE Intrinsics

DBOPEN	DBERROR	DBPUT
DBBEGIN	DBEXPLAIN	DBUNLOCK
DBCLOSE	DBFIND	DBUPDATE
DBCONTROL	DBGET	DBXBEGIN
DBDELETE	DBINFO	DBXEND
DBEND	DBLOCK	
DBXUNDO	DBMEMO	

## Other Intrinsic

<b>ACTIVATE</b>	<b>GENMESSAGE</b>	<b>PRINTFILEINFO</b>
<b>COMMAND</b>	<b>HPCICOMMAND</b>	<b>SUSPEND</b>
<b>CREATE</b>	<b>IODONTWAIT</b>	<b>VPRINTFORM</b>
<b>CREATEPROCESS</b>	<b>IOWAIT</b>	

For more information about which system libraries the NetBase shell is installed into and how to install it, see the Installation Section in this manual.

## NetBase Directory

The NetBase directory is used to define the NetBase environment. Entries in the directory tell NetBase which files are on remote systems and which files are to be shadowed. The directory itself is a hash file similar to an IMAGE master set. In fact, NetBase makes use of the IMAGE hashing algorithm. This type of file is used because of its fast keyed access. A KSAM image of the file is also maintained to facilitate the **LIST** command within the **NBDIR** directory maintenance program. The directory files are created and maintained using **NBDIR**.

When entries are added to the NetBase directory, they take effect for all new processes on the system. Anyone on the system may look at the NetBase directory entries, however, to add entries to the directory requires either **NM** or **SM** capability. The default directory that is created when NetBase is installed is called **NBD.DATA.NETBASE**. The associated KSAM file is **NBD0.DATA.NETBASE**.

When making an entry in the NetBase directory, there are basically three ways to specify files:

- **filename.group.account**
- **@.group.account**
- **@.@.account**

This limited use of wildcards is for performance reasons. When the NetBase shell intercepts an open of a file (**FOPEN,DBOPEN**), it must do a lookup in the directory to see if a NetBase service is required for the file. To minimize the amount of time spent in the directory the use of wildcards is limited. So, you can specify a specific file, all the files within a group, or all of the files within the account. When NetBase searches the directory, it first looks for the specific file name. If it does not find the specific filename, it looks for **@.group.account**. If it does not qualify here, it then searches to for an **@.@.account** entry. If no match is found, the request is passed to MPE/iX. If a match is found, the NetBase shell will do any necessary processing and then pass the request to MPE/iX.

When using the **LIST** commands in **NBDIR**, full use of the wildcard is supported. This is because when doing a **LIST** command, the KSAM file associated with the directory is used for the listing.

For information about creating and maintaining the NetBase directory, turn to Chapter 4 “NBDIR”.

## NetBase Control Program

The NetBase control program, **NBCTRL.PUB.NETBASE**, is used to stop and start NetBase services and to also monitor NetBase activity on a system. **NBCTRL** is used to stop and start the NetBase background job. It is also used to stop and start individual NetBase services such as remote file access, the exporting of transactions for shadowing and remote spooling.

Only one person on the system can run **NBCTRL** at any given time. Many of the commands within **NBCTRL** also require certain capabilities. **NBCTRL** is also used to configure NetBase. Within the **CONFIG** option of **NBCTRL**, all nodes in the NetBase configuration will be defined along with the NetBase services that will be used.

**NBCTRL** can also be used to get a status of NetBase. It provides information on the number of shadow transactions that have been sent to the shadow system, the number of users using NFA.

For more information about **NBCTRL**, see the “NBCTRL” chapter.

## CHAPTER 2

# Using NBCTRL

NBCTRL is an interactive program for controlling and monitoring NetBase operations. The NBCTRL program is used for configuring and starting or stopping NetBase. It displays current NetBase users and their status. The NBCTRL program is the operator's interface for controlling the NetBase process.

### In This Chapter

- **NBCTRL Command Summary**
- **Running NBCTRL**
- **Special Command Features**
- **Accessing NetBase Subsystems**
- **Using the NBDISABLE JCW**
- **NBCTRL Commands**

## NBCTRL Command Summary

The following table lists the **NBCTRL** commands.

Command	Function
*> <b>ABORT</b>	Immediately terminates some or all NetBase features.
*> <b>CHGDIR</b>	Changes the current default NetBase directory.
*> <b>CONFIG</b>	Invokes the NetBase configuration subsystem.
*> <b>DEFER</b>	Defers posting of selected filesets.
*> <b>DIR</b>	Invokes the NetBase directory subsystem.
*> <b>DISABLE</b>	Turns off one of the NetBase features.
*> <b>ENABLE</b>	Turns on one of the NetBase features.
*> <b>EXIT</b>	Exits from <b>NBCTRL</b> .
*> <b>HELP</b>	Displays a list of <b>NBCTRL</b> commands.
*> <b>ROLL</b>	Starts updating of a shadow file from the posting queues.
*> <b>SECURITY</b>	Invokes the NetBase security subsystem.
*> <b>SERVER</b>	Allow or abort access to files.
*> <b>SET</b>	Changes the current operating limits of NetBase.
*> <b>SHOW</b>	Displays current NetBase activity and status.
*> <b>START</b>	Starts NetBase or starts a NetBase service.
*> <b>STATS</b>	Controls NetBase Statistics services.
*> <b>STATUS</b>	Displays current NetBase status.
*> <b>STOP</b>	Stops NetBase or stops a NetBase service.
*> <b>TRACE</b>	Traces intrinsic activity of specific server.
*> <b>UNDEFER</b>	Starts normal posting of deferred updates to shadowed files.

## Running NBCTRL

**NBCTRL** is run from MPE using the following command:

```
:RUN NBCTRL.PUB.NETBASE
```

Since **NBCTRL** communicates directly with NetBase's controlling process, only one person may execute it at a time. Any additional users attempting to run it will receive the message **NBCTRL is currently running elsewhere** and the program will terminate.

When **NBCTRL** has been successfully started, it displays the following prompt:

```
NetBase Control Program [x.x] Update n (C) QUEST Software 1987
*>
```

Any **NBCTRL** command may be passed via the **INFO** parameter of the MPE **:RUN** command. If the **INFO** string contains a command, **NBCTRL** will start without printing the program banner, execute the command and terminate. This is useful for implementing NetBase UDCs (user defined commands), for example:

```
:RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"

:STREAMING NETBASE #J123
END OF PROGRAM
```

The installation tape contains a file with various NetBase UDCs. These can be enabled by typing:

```
:SETCATALOG NBUDC.PUB.NETBASE
```

Any user may run **NBCTRL**, however, certain commands require **SM**, **NM** or **OP** capability.

## Special Command Features

**NBCTRL** supports a help facility. When you type **HELP**, a list of valid commands display. **NBCTRL** also has some special features that provide MPE functionality within NetBase. Any MPE command that is executable in break may be issued from **NBCTRL**. To enter an MPE command, type the command prefixed by a colon (:).

The **REDO** command allows the last command entered to be modified and then executed. To modify and execute the last typed command, simply type **REDO** at the prompt, modify the line using standard **EDITOR** commands, and press return to execute the command.

## Accessing NetBase Subsystems

The configuration, directory, and security subsystems can be accessed through **NBCTRL**. Each of these subsystems has its own prompt. This helps the user determine what subsystem is running. See the following examples of the prompts:

- \*>** NetBase control system prompt
- C>** Configuration subsystem prompt
- D>** Directory subsystem prompt
- S>** Security subsystem prompt

## Using the NBDISABLE JCW

Many of the NetBase features that are enabled through **NBCTRL** can be disabled for a particular job or session. This is done with a job control word called **NBDISABLE**. The NetBase shell checks **NBDISABLE** the first time a user calls **FOPEN** or **DBOPEN** for a file or database. If any of the flags are set that disable a service, NetBase disables those services for that user only. **NBDISABLE** cannot be used to enable a service.

**NBDISABLE** can be set from MPE using the following command:

### Syntax

```
:SETJCW NBDISABLE=jcwvalue
where
jcwvalue =      1 - Disables buffering
                 4 - Disables userexits
                 8 - Disables critical
                16 - Disables Directory Wildcard search
                32 - Disables direct connect communication
                64 - Do not map NetBase equations locally
               128 - Disables NetBase spooling
               256 - Disables AutoRPM
               512 - Disables remote file building
              1024 - Disables server release on last close
              WARN - Enables NetBase Application Tracing
              FATAL - Disables all NetBase services
```

As an example, turn off all NetBase services by setting the JCW to **FATAL**:

```
:SETJCW NBDISABLE=FATAL
:RUN PROGRAM
```

The current value of **NBDISABLE** can be determined at any time by issuing the **:SHOWJCW** command in MPE.

The flags and their meanings can be displayed in **NBCTRL** using the following command:

```
*> HELP NBDISABLE
```



**Note.** NetBase services can also be disabled at the program level. Turn to the “PROGRAM” section in Chapter 4 for more information about this feature.

---

## NBCTRL Commands

The following sections describe the **NBCTRL** commands in detail.

### ABORT

This command immediately terminates NetBase services to a user or node. It also aborts a server, or aborts the NetBase control process. This command requires **NODE MANAGER (NM)** capability.



**Stop!** Aborting NetBase may cause synchronization problems with shadowing. *Do not abort NetBase while the shadow transport is active.*

---

The **ABORT** command immediately terminates NetBase services to a user or a node, or terminates the entire NetBase system. This command is similar to the **STOP** command. However, the **ABORT** command terminates services immediately instead of letting processes complete any open transactions before terminating.

When the **ABORT** command is issued, NetBase services to a user, server, node or NetBase job are terminated. Once NetBase services have been aborted, subsequent requests for those services will return a NetBase error notifying the requesting process that services have been stopped.

### Syntax

```
*> ABORT {SERVER      server-id}
          {NODE        node-id  }
          {CLIENT     client-id}
          {RPC         {node | @}}
          {SPOOLING
          {SPOOL IN   node-id  }
          {SPOOL OUT  node-id  }
          {SPOOL UNIX
          {SQL
          {EXPORT
          {IMPORT     node-id  }
          {NETBASE
```

## Parameters

Parameter	Description
<b>CLIENT</b>	Aborts the processes which allow NetBase Client to access data on an HP 3000. You can either enter a client number, or @ to abort all clients. Active as well as inactive clients are aborted.
<b>EXPORT</b>	Aborts the process responsible for exporting shadow records.
<b>IMPORT <i>node-id</i></b>	Aborts the process responsible for importing shadow records from the node specified.
<b>NETBASE</b>	Terminates all NetBase services and stops the background job.
<b>NODE <i>node-id</i></b>	Terminates the NetBase node name or number specified. The configured nodes are displayed in the <b>SHOW STATUS</b> command.
<b>SERVER <i>server-id</i></b>	Terminates the NetBase server number specified. The server number is displayed under <b>SRVR</b> in the <b>SHOW REMOTE</b> or <b>SHOW SERVER</b> command.
<b>SPOOLING</b>	Terminates all spooling activity to and from the local node.
<b>SPOOL IN <i>node-id</i></b>	Aborts the process that receives spooling output from the node specified. The configured nodes are displayed in the <b>SHOW STATUS</b> command.
<b>SPOOL OUT <i>node-id</i></b>	Aborts the process that transmits spooling output to the node specified. The configured nodes are displayed in the <b>SHOW STATUS</b> command.
<b>SPOOL UNIX</b>	Aborts the process that receives spooling output from UNIX nodes.
<b>SQL</b>	Aborts the SQL shadow transport.

## Examples

To abort a remote user, first obtain the server number:

```
*> SHOW REMOTE
```

NODE	JOBNUM	SRVR	PIN	JOB NAME	PROGRAM NAME
ACCT	#J102	1	47	GLPOST, JOB.ACCT	GLPOST.PUB.PROD
DIST	#S507	3	122	JILL.DIST	SOENTER.PUB.PROD
ACCT	#S202	4	132	JACK.PURCHASE	POENTER.PUB.PROD

```
*> ABORT SERVER 3
```

```
*> SHOW REMOTE
```

NODE	JOBNUM	SRVR	PIN	JOB NAME	PROGRAM NAME
ACCT	#J102	1	47	GLPOST, JOB.ACCT	GLPOST.PUB.PROD
ACCT	#S202	4	132	JACK.PURCHASE	POENTER.PUB.PROD

To abort all services to and from a node using the **SHOW STATUS** command to determine the ID numbers of the active nodes:

```
*> show status
```

```
STATUS FOR NODE 8: DBSERVER
```

NODE	STATUS	----- NFA -----		-- SPOOLING --		---- SHADOWING ---			
		ACCESS	LOCAL	REMOTE	EXPORT	IMPORT	EXPORT	IMPORT	POST
1 CLIENT2	UP	IN/OUT	0	0			DOWN	INACT	STOP
7 CLIENT1	UP	IN/OUT	0	0	ACTIVE	ACTIVE	DOWN	ACTIVE	STOP
9 PRSERVER	DOWN				INACT	INACT	DOWN	INACT	STOP

```
LOCAL:
```

```
REMOTE USERS: 0 LIMIT = 64
LOCAL USERS: 0 LIMIT = 500
SERVERS: 0 ALLOC = 0
CLIENTS: 0
ROLL FORWARD: 0
```

```
SERVICES:
```

```
NFA CACHING: ACTIVE
STATISTICS: INACTIVE
SQL SHADOW: ACTIVE
CLIENT ACC: ACTIVE
```

```
*> ABORT NODE 7
```

## Chapter 2

---

Issue the following command to shutdown the entire NetBase job, ignoring active services:

```
*> ABORT NETBASE
```



**Stop!** Aborting NetBase may cause synchronization problems with shadowing if aborted while the shadow transport is active.

---

Note that the **ABORT** command immediately terminates services to the process specified. This command will cause programs currently executing to terminate with a NetBase error upon their next NetBase file request. The **STOP** command will allow these processes to complete while refusing services to any new programs.

## CHGDIR

This command changes the active NetBase directory. *The new directory will effect new processes only.* Existing processes will continue to use the directory that was active when the process was started. This command requires **NODE MANAGER(NM)** or **SYSTEM SUPERVISOR(OP)** capability.

### Syntax

```
*> CHGDIR filename
```

### Parameters

Parameter	Description
<i>filename</i>	The fully-qualified name of a valid NetBase directory.

### Examples

To change the active NetBase directory to **NBDIR.TEST.NETBASE**:

```
*>CHGDIR NBDIR.TEST.NETBASE
```

## CONFIG

This command invokes the NetBase configuration subsystem. The user must be logged into the **NETBASE** account to access this subsystem. This command requires **ACCOUNT MANAGER(AM)** or **SYSTEM MANAGER(SM)** capability.

Changes made to the configuration parameters while NetBase is running may not take effect immediately. Turn to Chapter 3 “CONFIG” for a description of parameters and when changes to each parameter takes effect.

### Syntax

```
*> CONFIG
```

### Parameters

This command does not have parameters.

### Examples

To list the **DIR** configuration parameters:

```
:RUN NBCTRL.PUB.NETBASE

NetBase Control Program [x.x] Update n (C) QUEST Software 1987

*> CONFIG
C> L DIR

DIR - Default Directory Filename...NBD.DATA.NETBASE
DIR - Allow File Equates.....N
DIR - Allow Wildcards.....Y
DIR - Map Local Files with Dir.....N
```



**Note.** Enter the **CONFIG** subsystem also by running `:NBCTRL.PUB.NEBASE;INFO="CONFIG"`.

---

## DEFER

This command defers the posting of shadowed transactions from the specified node for the listed file sets, queuing those updates instead. This allows a specific shadow file or set of shadowed files to be taken off-line, without interrupting the posting for any remaining shadowed files. Reference the **UNDEFER** command for information on re-initiating the posting of the pending transactions. This command requires **SYSTEM MANAGER(SM)** or **SYSTEM SUPERVISOR(OP)** capability.

### Syntax

```
*> DEFER node={fileset[,fileset[,...]]}
           {!indirectfile}
```

### Parameters

Parameter	Description
<b>node</b>	The NetBase node number or name of the importing node.
<b>fileset</b>	A fully-qualified file set for which updates are to be deferred.
<b>!indirectfile</b>	The name of a file containing a list of file sets, each file set listed on a separate line, for which updates are to be deferred.

### Examples

To defer the posting of updates from node **FOX** for the shadow files **BANNER.DATA.NETBASE** and **PAYDB.DATA.PAYROLL**, queuing the updates instead:

```
:RUN NBCTRL

NetBase Control Program [x.x] Update n (C) QUEST Software 1987

*> DEFER FOX=BANNER.DATA.NETBASE,PAYDB.DATA.PAYROLL
Scanning Directory... 2 files deferred
*> SHOW POST FOX
```

File	Type	Updates	Time Stamp of Last Record Processed	S Y N	Q U E	D I S
BANNER.DATA.NETBASE	IMAGE	17978	03/02/92 12:04:32	-	P	-
PAYDB.DATA.PAYROLL	IMAGE	76280	03/02/92 12:11:51	-	P	-

## DIR

This command gives the user access to the NetBase directory subsystem. Turn to Chapter 4 “NBDIR” chapter for more information on the directory subsystem. This command requires **NODE MANAGER(NM)** or **SYSTEM MANAGER(SM)** capability.

### Syntax

```
*> DIR
```

### Parameters

This command does not have parameters.

### Examples

```
:RUN NBCTRL
```

```
NetBase Control Program [x.x] Update n (C) QUEST Software 1987
```

```
*> DIR  
D> BUILD NBTEST.DATA,500  
D> EXIT
```



**Note.** Since **NBCTRL** communicates directly with the NetBase controlling process, only one person may execute it at a time. This should be considered when invoking the directory subsystem through **NBCTRL**. The directory subsystem can be accessed directly by running the program **NBDIR**. It is recommended that **NBDIR** be used whenever extended directory maintenance is required.

---

## DISABLE

The **DISABLE** command is used to turn off one or more of the NetBase functions. The function remains disabled until it is re-enabled using the **ENABLE** command. This command should be issued before NetBase is started. If issued while NetBase is running, it will only take effect on some of NetBase processes. This command requires **SYSTEM MANAGER(SM)** capability.



**Warning!** While the **DISABLE** command may be used to completely disable the shadowing module of NetBase, *it is not recommended*. The preferred method of turning off shadowing is to remove the entries in the directory which define the files to be shadowed, or **CHGDIR** to an empty directory or a directory which includes no shadowing definitions.

---

If the **DISABLE** command is used to globally disable shadowing, an **ENABLE SHADOWING** command will have to be followed by a restart of the NetBase background job and all user processes to completely ensure that shadowing has been turned back on. Failure to do this may result in sync loss.

### Syntax

```
*> DISABLE {AUTORPM }
           {BUFFERING }
           {CRITICAL }
           {PENDLOCK }
           {SHADOWING }
           {SPOOLING }
           {STATISTICS }
```

## Parameters

The parameters can be abbreviated to the first three characters.

Parameter	Description
<b>AUTORPM</b>	Disables automatic remote process management.
<b>BUFFERING</b>	Disables the buffering feature of NFA.
<b>CRITICAL</b>	Makes it possible for a shadowed file or database transaction to be interrupted, enabling the file update to happen while the shadow write may not.
<b>PENDLOCK</b>	Causes an unconditional lock to be performed as an individual network transaction, rather than combining it with the next call. Note that these changes are dynamic.
<b>SHADOWING</b>	Disables the shadowing of files and databases.
<b>SPOOLING</b>	Disables all spooling activity to and from this node.
<b>STATISTICS</b>	Disables the statistics collection module.

## Examples

To disable the buffering feature for all **LOCAL** users:

```
*-> DISABLE BUF
```

## ENABLE

Turns on various NetBase features that operate on and from the local node. This command is used to enable various NetBase services that have been disabled with the **DISABLE** command. It should be issued before NetBase is started. If issued while NetBase is running, it will only take effect on some NetBase processes. This command requires **SYSTEM MANAGER(SM)** capability.



**Note.** If the **DISABLE** command is used to globally disable shadowing, you must restart the NetBase background job and all user processes after issuing the **ENABLE SHADOWING** command. This is the only way to completely ensure that shadowing has been turned back on. Failure to do this will likely result in sync loss, as some user processes will perform shadowing and others, which were not restarted, will not.

### Syntax

```
*> ENABLE {AUTORPM }
          {BUFFERING }
          {CRITICAL }
          {PENDLOCK }
          {SHADOWING }
          {SPOOLING }
          {STATISTICS }
```

### Parameters

The parameters can be abbreviated to the first three characters.

Parameters	Description
<b>AUTORPM</b>	Enables automatic remote process management.
<b>BUFFERING</b>	Enables the buffering feature.
<b>CRITICAL</b>	Prevents a shadowed file or database transaction from being interrupted, preventing a file update to occur when the shadow write does not.

*Continued*

Parameters	Description
<b>PENDLOCK</b>	Couples an unconditional database lock with the next call to that database into a single network transaction. A DBLOCK followed by a DBUNLOCK of the same database without any other activity will not be sent to the remote machine.  Do not use if DBLOCKS are being used to synchronize external data structures, or if any remote system is an MPE/V system. Otherwise, this option further reduces networking overhead. Note that these changes are dynamic.
<b>SHADOWING</b>	Enables the shadowing of files and databases.
<b>SPOOLING</b>	Enables all spooling activity to and from this node.
<b>STATISTICS</b>	Enables the statistics collection module.

### Examples

To enable the shadowing of files and databases:

```
*-> ENABLE SHA
```

## ROLL FORWARD

This command causes the posting process to roll forward transactions from the posting queue files into their shadow files. This command requires **SYSTEM MANAGER(SM)** capability.

If the posting process encounters an error while opening a shadowed file, the transactions are preserved in a post queue file. **POST** reports the exceptional condition encountered during the file open to the system console, as well as the NetBase job **\$STDLIST**. Once the source of the problem has been determined and rectified, the **ROLL FORWARD** command is used to cause the posting process to update the specified file with the updates that have been queuing.

If a file is found to be in the **QUEUE** state, and the original error which caused this has been lost, it is generally safe to attempt a **ROLL FORWARD** command. If the original cause was a transient condition, i.e. a file being backed up when **POST** tried to open it, which has since cleared, this will most likely succeed. However, if the post encounters the same error again, for example, a bad database password in the **SHADOW** directory entry, the file will go back into the **QUEUE** state.

### Syntax

```
*> ROLL FORWARD filename
```

### Parameters

Parameter	Description
<i>filename</i>	The name of a file that is being shadowed that has post queues associated with it. These files show on the <b>SHOW POST</b> node display with the queuing flag set to <b>Q</b> .

### Examples

```
*> SHOW POST 1
```

File	Type	Updates	Time Stamp of Last Record Processed	S Q D Y U I N E S
FT.XDATA.QUEST	???	0	09/12/91 11:47:32	- Q -

```
*> ROLL FORWARD FT.XDATA.QUEST
ROLL FORWARD PROCESS 1 STARTED
```

## SECURITY

This command gives the user access to the NetBase security subsystem. Turn to the “NFA Security” section in Chapter 6 for more information on the security subsystem. This command requires **SYSTEM MANAGER(SM)** and **NODE MANAGER(NM)** capability.



**Note.** Since **NBCTRL** communicates directly with the NetBase controlling process, only one person may execute it at a time. This should be considered when invoking the security subsystem through **NBCTRL**. The security subsystem can be accessed directly by running the program **NBSECURE**. It is recommended that **NBSECURE** be used whenever extended security maintenance is required.

---

### Syntax

```
*> SECURITY
```

### Parameters

This command does not have parameters.

### Examples

To enter the security subsystem, issue a command and exit back to NetBase control:

```
*> SECURITY
S> ALLOW @. @:1;ACCESS=R,W;@. @. PAYABLES
S> EXIT
```

## SERVER

The **SERVER** command is used to allow or abort access to files. This command requires **NODE MANAGER (NM)** or **SYSTEM SUPERVISOR (OP)** capability.

```
*> HELP SERVER
```

Controls Server Processes

### Syntax

```
*> SERVER {ABORT=fileset[,fileset[,...]]}
          {ALLOW=fileset[,fileset[,...]]}
```

### Parameters

Parameter	Description
<b>ABORT</b>	Eliminates access to specified file set(s) for all remote users on the system. If the <b>SERVER ABORT</b> command is used to stop access to files, then the <b>SERVER ALLOW</b> command must be issued to allow access again.
<b>ALLOW</b>	Allows access to specified file set(s) for remote users on the system.

### Examples

To disallow access to all files in **DATA.AP**:

```
:NBCTRL.PUB.NETBASE
```

```
*> SERVER ABORT=@.DATA.AP
```

```
ABORT REQUEST HAS BEEN SENT TO ALL ACTIVE SERVERS
```

To allow access to the files in **DATA.AP**:

```
*> SERVER ALLOW=@.DATA.AP
```

```
ALLOW REQUEST HAS BEEN SENT TO ALL ACTIVE SERVERS
```

## SET

This command dynamically changes the limit or the priority queue for NetBase processes. The **SET** command allows current NetBase limits to be changed while NetBase is running. The number of local users, remote users and the number of servers available can be changed up to their maximum allowable values. This command will also re-schedule the priority of the shadowing export and posting processes. This command requires **NODE MANAGER(NM)** or **SYSTEM SUPERVISOR(OP)** capability.

Changing the **SERVER** allocation may cause server processes to be created or terminated. If the allocation is changed by a significant amount, there will be a delay while **NBCOP** creates or terminates these servers.

### Syntax

```
*> SET {LOCAL=limit          }
      {REMOTE=limit         }
      {SERVERS=allocation   }
      {EXPORT=priority      }
      {POST node=priority   }
```

### Parameters

Parameter	Description
<b>LOCAL=limit</b>	The number of local user processes able to access NetBase. This number cannot exceed <b>2048</b> .
<b>REMOTE=limit</b>	The number of remote processes able to access the local node. This number cannot exceed <b>2500</b> .
<b>SERVERS=allocation</b>	The number of servers allocated. This number cannot exceed the current limit for remote users.
<b>EXPORT=priority</b>	The priority queue ( <b>BS</b> , <b>CS</b> , <b>DS</b> or <b>ES</b> ) in which the shadow <b>EXPORT</b> process will run.
<b>POST node=priority</b>	The priority queue ( <b>BS</b> , <b>CS</b> , <b>DS</b> or <b>ES</b> ) in which the <b>POSTING</b> process for the specified node will run.

## Examples

If during a heavy period of user updating, it was discovered that the shadowing posting process for node 2 was using too much of the processor, you can issue the following command to lower the posting priority:

```
*> SET POST 2=DS
```

The **SHOW STATUS** command can be used to display current limits. The current limits are listed under **LOCAL** in the display. To allow 80 users on this system access to NetBase services, 100 users from other systems access to files on this system, and preallocate 70 servers for remote access:

```
*> STATUS
```

```
STATUS FOR NODE 1: SYSA
```

NODE	STATUS	----- NFA -----			-- SPOOLING --		---- SHADOWING ---			
		ACCESS	LOCAL	REMOTE	EXPORT	IMPORT	EXPORT	IMPORT	POST	
2	SYSB	UP	IN/OUT	0	0			DOWN	INACT	STOP
3	SYSC	UP	IN/OUT	0	0	ACTIVE	ACTIVE	DOWN	ACTIVE	STOP

```
LOCAL:
```

```
REMOTE USERS: 0    LIMIT = 64
LOCAL  USERS: 0    LIMIT = 500
SERVERS:      0    ALLOC = 0
CLIENTS:      0
ROLL FORWARD: 0
```

```
SERVICES:
```

```
NFA CACHING: ACTIVE
STATISTICS: INACTIVE
SQL SHADOW: ACTIVE
```

```
*> SET LOCAL=80
```

```
*> SET REMOTE=100
```

```
*> SET SERVERS=70
```

## Chapter 2

---

The **STATUS** command can be used to verify the new limits:

```
*> STATUS
```

```
STATUS FOR NODE 1: SYSA
```

NODE	STATUS	----- NFA -----			-- SPOOLING --		---- SHADOWING ---			
		ACCESS	LOCAL	REMOTE	EXPORT	IMPORT	EXPORT	IMPORT	POST	
2	SYSB	UP	IN/OUT	0	0			DOWN	INACT	STOP
3	SYSC	UP	IN/OUT	0	0	ACTIVE	ACTIVE	DOWN	ACTIVE	STOP

```
LOCAL:
```

```
REMOTE USERS:    0    LIMIT = 100
LOCAL  USERS:    0    LIMIT =  80
SERVERS:         70    ALLOC =   0
CLIENTS:         0
ROLL FORWARD:   0
```

```
SERVICES:
```

```
NFA CACHING: ACTIVE
STATISTICS: INACTIVE
SQL SHADOW: ACTIVE
```

## SHOW

The **SHOW** command is used to display the status of all the NetBase processes. The **STATUS** command is an abbreviation for the **SHOW** command with the **STATUS** option. The listing for any **SHOW** command can be terminated by entering <Ctrl-Y>.

### Syntax

```
*> SHOW {SERVER [n] }
        {LOCAL [n] }
        {REMOTE [n] }
        {CLIENT [n] }
        {EXPORT }
        {IMPORT [node] }
        {POST [import-node] }
        {SERVICES }
        {SQL }
        {STATUS }
        {JCWSTATUS }
        {VARSTATUS }
```

### Parameters

Parameter	Description
<b>SERVER [n]</b>	Displays information for all servers or the server specified. The information for a server includes a list of the files that the server is accessing and the access mode for each file. If the server number is not specified, a list of all servers is displayed.
<b>LOCAL [n]</b>	Displays the status of local user processes using <b>NFA</b> or <b>STATS</b> . The number is the NetBase identification number of a local user process. If not specified, a list of local user processes is displayed.
<b>REMOTE [n]</b>	Displays the status of one or all remote users accessing the local node. The number is the server number for the remote process. If not specified, a list of all remote user processes is displayed. The <b>SHOW REMOTE</b> display is similar to the <b>SHOW SERVER</b> display except that the files being accessed by the server are not listed.

*Continued*

Parameter	Description
<b>CLIENT</b> [ <i>n</i> ]	Displays the status of incoming client systems. The information for a client system includes a list of the files that the system is accessing and the access mode for each file. If the client number is not specified, a list of all client systems is displayed.
<b>EXPORT</b>	Displays the number of shadow transactions sent to each node, as well as the number queued for export. Also shows the current <b>BACKLOG</b> , which is the number of records written to <b>NBM</b> that have not yet been read by <b>EXPORT</b> . All statistics are reset when <b>EXPORT</b> is stopped.
<b>IMPORT</b>	Displays the number and status of shadow transactions coming in from each node. If the state displayed for a particular node is <b>transfer</b> , this command also shows what percentage of the transfer is complete.
<b>POST</b> [ <i>import-node</i> ]	Displays the posting status of the node specified. The <i>import-node</i> is the node name or number of one of the nodes in the configuration. If a node is not specified, the posting status of all nodes is displayed. If a node is specified, a detailed display for that node is shown listing all files that <b>POST</b> currently has open.
<b>SERVICES</b>	Displays the availability of each NetBase service.
<b>SQL</b>	Displays the status of each DBE shadow process.
<b>STATUS</b>	Displays the status of NetBase processes for the local node.
<b>JCWSTATUS</b>	Displays the status of NetBase processes by setting a series of MPE/V JCWs. It is replaced by <b>VARSTATUS</b> on MPE/iX systems.
<b>VARSTATUS</b>	Displays the status of NetBase processes by setting a series of MPE/XL variables. Variables set include <b>NBLOCAL</b> , <b>NBREMOTE</b> , and <b>NBCLIENT</b> . A normal <b>SHOW STATUS</b> display is done as well.

### SHOW STATUS Command

The information displayed for each command option is different. The **SHOW STATUS** command displays the status of NetBase processes, the number of local users, remote users and servers, and the status of NetBase services. The information that is pertinent to NetBase shadowing is the status of the shadowing processes, **EXPORT**, **IMPORT** and **POST**, and the **ROLL FORWARD** statistic. For spooling, it is the state of the spool **EXPORT** and **IMPORT** processes.

The shadowing processes may be in one of seven states. Following is a list of the valid states with an explanation of their meaning:

State	Description
<b>ACTIVE</b>	This indicates that the shadow process is configured and is running and that its co-process is also up and running on the other system.
<b>DOWN</b>	This indicates that the shadow process is configured but that the process has not been started or has been stopped.
<b>INACTIVE</b>	This indicates that <b>IMPORT</b> is configured, and that the process is up, but its corresponding <b>EXPORT</b> process is not up.
<b>QUEUE</b>	This indicates that this NetBase shadow <b>EXPORT</b> process is unable to communicate with its <b>IMPORT</b> counterpart and that the shadow transactions are being queued on this node.
<b>STOP</b>	This indicates that the shadow process is configured, and the process is running, but it has been stopped with a <b>STOP</b> command.
<b>WAIT</b>	This indicates that <b>EXPORT</b> or <b>IMPORT</b> is configured for <b>Transfer Mode Only</b> , and that the process is up waiting for the command to connect to its coprocess-process. The transport will connect when a <b>START IMPORT</b> or <b>START EXPORT</b> command is issued in <b>NBCTRL</b> .
<b>XFER</b>	This indicates that the shadow transport is in bulk transfer mode. This method of transport is used whenever there is a backlog of queued shadow transactions to be exported to a remote node or when the transport has been configured as <b>Transfer Mode Only</b> .
<i>(blank)</i>	Indicates that this process is not configured to run for this node.

### SHOW STATUS Example

To display the status of NetBase processes for the local node:

```
*> SHOW STATUS
```

```
STATUS FOR NODE 8: DAFFY
```

```

      NODE      STATUS      ----- NFA -----  -- SPOOLING --  ---- SHADOWING ----
      ACCESS LOCAL REMOTE  EXPORT  IMPORT  EXPORT  IMPORT  POST
1 MASTER      UP      IN/OUT      0      0
2 FOX         UP      IN/OUT      0      0  ACTIVE  ACTIVE  ACTIVE  ACTIVE  ACTIVE
9 BOB         DOWN
      INACT  INACT  DOWN  INACT  STOP

```

```

LOCAL:
  REMOTE USERS:    0    LIMIT =   64
  LOCAL  USERS:    0    LIMIT =  500
  SERVERS:         0    ALLOC =    0
  CLIENTS:         0    ALLOC =    0
  ROLL FORWARD:   0

SERVICES:
STATISTICS: INACTIVE
  CLIENT ACC: ACTIVE
    
```

In the above example, the status of all processes importing from or exporting to other nodes is displayed. The local user counts describe the number of local user processes that are using NFA to access remote systems or are being tracked locally by NetBase statistics. The remote user count is the number of remote users, running processes on remote systems, that are connected to a server process on the local system for access to local files. The shadow export process is active and is exporting to node 2. The shadow import and the posting processes are also active.

### SHOW POST Command

The **SHOW POST** command, when issued without a node number specified, displays a list of the posting processes and information about the status of each one. This command displays the following information:

Information	Description
<b>Node</b>	The node number of the computer exporting to this machine.
<b>Name</b>	The node name of the computer exporting to this machine.
<b>Status</b>	<p>The status of the posting of transactions from this node. One of three states will be displayed:</p> <p><b>DOWN ¾</b> Indicates that the posting process for this node is not running.</p> <p><b>IDLE ¾</b> Indicates that <b>POST</b> is up but there are no records to be posted.</p> <p><b>ACTIVE ¾</b> Indicates that records are being actively processed.</p>
<b>Backlog</b>	The current EOF of the <b>IMPORT</b> queue file that is being used by <b>POST</b> . If multiple queue files exist, an asterisk (*) will appear next to the value indicating that the EOF only reflects the current queue file that <b>POST</b> is reading from and does not indicate the total backlog across all queue files.
<b>Files</b>	The number of files and databases that <b>POST</b> has open.
<b>Closes</b>	The number of files that have been closed by request.

The **ROLL FORWARD** statistic is the number of roll forward processes that are executing. A roll forward process is started by the **ROLL FORWARD** command. It is used to roll transactions forward from a post queue into the actual file for which that post had been queuing.

### SHOW POST <import node> Command

The **SHOW POST <import-node>** command displays a list of all shadow files that the posting process has open, including the status of the posting to each file. The following information is displayed:

Information	Description
<b>File</b>	The name of the shadow file that is being updated by the posting process.
<b>Type</b>	The type of the shadow file. This field can have one of six values indicating which type of file the posting process has opened. The value may be <b>IMAGE</b> , <b>MPE</b> , <b>KSAM</b> , <b>MSG</b> or <b>LOG</b> indicating the type of file. If <b>RIO</b> is displayed in this field, the file is a relative <b>COBOL</b> file. If the value of this field is <b>???</b> , the posting process was not able to open the file and consequently does not know what the file type is.
<b>Updates</b>	The number of updates applied to this file since <b>POST</b> was last started.
<b>Time Stamp...</b>	The time and date stamp of the record that was last processed. This time and date reflects the time the transaction was created on the master system, not the time that it was posted to the shadow copy of the file. If this time is significantly in the past, it indicates one of two conditions. Either there is currently a backlog of transactions, in which case the difference between this time stamp and the clock of the master system will indicate a rough measure of the shadowing time lag, or (more likely) there have been no transactions made to this file since the time displayed.
<b>SYN</b>	This flag indicates that the shadow file is out of synchronization.
<b>QUE</b>	This flag indicates that updates to this file are being queued. If this field displays <b>Q</b> , it indicates that a post queue has been created for this file. If this field displays <b>P</b> , it indicates that the posting for this file has been deferred with the <b>DEFER</b> command.
<b>DIS</b>	This flag indicates that updates are being discarded due to a serious error on the file or a synchronization problem.

### SHOW POST <import node> Example

To display the status of all files being updated by the posting process for node 8:

```
*> SHOW POST 8
```

File	Type	Updates	Time Stamp of Last Record Processed	S Q D Y U I N E S
GL.DATA.ACCT	IMAGE	3721962	04/19/91 13:47:32	- - -
GLCUSTL.DATA.ACCT	MPE	95	04/19/91 11:53:17	S - D
FINDB.DATA.ACCT	???	0	04/19/91 13:45:38	- Q -

In the example above, the file **GL.DATA.ACCT** is an IMAGE database. It has had **3,721,962** updates since the posting process was last started. File **GLCUSTL.DATA.ACCT** is an MPE file and has had **95** updates applied to it. The **SYN** flag indicates that the file is out of synchronization and the **DIS** flag indicates that updates are being discarded. The file **FINDB.DATA.ACCT** was unable to be opened by the posting process. The error that **POST** received when attempting to open the file is displayed to the console and to the NetBase job **\$STDLIST**. Once the problem is rectified, the **ROLL FORWARD** command can be used to post the queued updates to the file.

### Other SHOW Examples

Information about local users and the remote computers that they are currently accessing can be displayed using the **SHOW LOCAL** command:

```
*> SHOW LOCAL
```

USR	JOBNUM	LCL	PIN	JOB NAME	PROGRAM NAME
2	#J173	01	201	MRP, JOB. PLANT DAFFY FOX	MRP. PUB. PROD SERVER 12 SERVER 3
3	#S223	01	97	JOHN. PLANT FOX	ORDINQ. PUB. PROD SERVER 14

For information about remote users, enter the **SHOW REMOTE** command:

```
*> SHOW REMOTE
```

NODE	JOBNUM	SRVR	SPIN	JOB NAME	PROGRAM NAME
DAFFY	#J102	1	105	GLPOST, JOB.ACCT	GLPOST.PUB.PROD
FOX	#S507	3	79	JILL.DIST	SOENTER.PUB.PROD
DAFFY	#S202	4	121	JACK.PURCHASE	POENTER.PUB.PROD



**Note.** The **SPIN** that is displayed here is the **PIN** of the server process allocated to the remote process.

For information about one of the remote users, enter **SHOW REMOTE** with the server number:

```
*> SHOW REMOTE 3
```

NODE	JOBNUM	SRVR	UPIN	JOB NAME	PROGRAM NAME
DAFFY	#S507	3	122	JILL.DIST	SOENTER.PUB.PROD

To display the status of incoming client systems, enter the **SHOW CLIENT** command.

```
*> SHOW CLIENT
```

Client	SPin	Node	User Name	Program Name
1	142	SILVER.QUESTS.COM	Joe	Query
1		ORDERS.NBUXDEMO.NETBASE		ACC=IN, LOCK

For UNIX systems (see above), the client, server process on the HP 3000, and node name are displayed along with the file name. For PCs (see below), the network address of the incoming client system is shown instead of the node name.

```
*> SHOW CLIENT
```

Client	SPin	Node	User Name	Program Name
1	167	192.77.210.69	John	
1		BANNER.DATA.NETBASE		ACC=IN, LOCK

## Chapter 2

---

To display the number of shadow transactions sent to each node, as well as the number queued for export, enter the **SHOW EXPORT** command:

```
*> SHOW EXPORT
```

```
No Backlog at this time. Priority is CS, shifted 1 times.
```

#	NODE	STATE	TRANSACTIONS SENT	TRANSACTIONS QUEUED
1	MASTER	ACTIVE	644	0
2	FOX	ACTIVE	31386	0
8	DAFFY	ACTIVE	416940	0
9	BOB	QUEUE	0	2

To display the number and status of shadow transactions coming in from each node, enter the **SHOW IMPORT** command:

```
*> SHOW IMPORT
```

#	Node	State	Transactions Received	Transfer % Done
1	MASTER	ACTIVE	0	
2	FOX	ACTIVE	0	
8	DAFFY	ACTIVE	23	
9	BOB	INACT		



**Note.** If the state displayed for a particular node is **transfer**, see the last column for the percentage of the transfer that is complete.

---

To list the status of all the posting processes on the local node, enter the **SHOW POST** command:

```
*> SHOW POST
```

Node	Name	Status	Backlog	Files	Closes
2	FOX	ACTIVE	227*	18	
8	DAFFY	IDLE		3	

\* The backlog does not include additional queue files

In the previous example, the posting process for node 2 is actively posting to eighteen shadowed files. The asterisk indicates that the backlog is greater than the amount shown by one or more import queue files. This usually is a result of the posting process for this node having been shutdown for some period of time, such as for a system backup. The posting process for node 8 is also running, but there is no activity from the master node. That posting process has three files open and there is no current backlog of shadow data.

To determine the current availability of installed NetBase services, use **SHOW SERVICES**:

```
*> SHOW SERVICES
```

```
NETBASE SERVICES STATUS:
```

```
AUTO RPM:      ENABLED
BUFFERING:     ENABLED
CRIT UPDATE:   ENABLED
PEND LOCKS:    ENABLED
SPOOLING:      ENABLED
SHADOWING:     ENABLED
STATISTICS:    DISABLED
```

For information regarding SQL databases defined for shadowing, use the **SHOW SQL** command:

```
*> SHOW SQL
```

```
Exporting Status:
```

#	Node	DBE Name	State	PIN
3	FOX	DBE2.ROBIN.NETBASE	ACTIVE	73

```
Importing Status:
```

#	Node	DBE Name	State	PIN
3	FOX	DBE2.SHADOW.NETBASE	ACTIVE	119

To display the status of NetBase processes in descriptive MPE/iX variables, use the **SHOW VARSTATUS** command. In this way, batch job streams can be built to automate a number of operational tasks related to monitoring the state of the NetBase processes. For example:

```
*> SHOW VARSTATUS
*> EXIT
:SHOWVAR
NBCTRL = 0
NODE1NFA = IN/OUT
NODE1STATE = UP
NODE1EXPORT = DOWN
NODE1IMPORT = INACT
NODE1POST = STOP
NODE7NFA = IN/OUT
NODE7STATE = UP
NODE7SPOOLEXP = ACTIVE
NODE7SPOOLIMP = ACTIVE
NODE7EXPORT = DOWN
NODE7IMPORT = ACTIVE
NODE7POST = STOP
NODE9STATE = DOWN
NODE9SPOOLEXP = INACT
NODE9SPOOLIMP = INACT
NODE9EXPORT = DOWN
NODE9IMPORT = INACT
NODE9POST = STOP
NBSTATUS = 1
NBLOCAL = 0
NBREMOTE = 0
NBCLIENT = 0
```

You can also use **SHOW VARSTATUS** to check to see if a NetBase job is running. See the following description of **SHOW VARSTATUS**:

```
:showvar
NBCTRL - 0 - indicates NBCTRL ran successfully
         1 - indicates NBCTRL crashed
         2 - indicates Netbase is down.

NODenEXPORT - (n) is the node to which EXPORT is being reported
  ACTIVE - Export is active to this node
  QUEUE  - Export is queued for this node
  STOP   - Export has been stopped for this node

NODenIMPORT - (n) is the node for which IMPORT is being reported
  ACTIVE - Import is active for this node
  QUEUE  - Import is queued for this node
  STOP   - Import has been stopped for this node

NODenPOST - (n) is the node for which POST is being reported
  ACTIVE - POST is active for this node
  QUEUE  - POST is queued for this node
  STOP   - POST has been stopped for this node.

NODenSTATE - (n) is the node for which STATE is being reported
  UP      - The local NetBase job can communicate
            with this node
  DOWN    - The local NetBase job can't communicate
            with this node

NBSTATUS - 1 - Indicates that SHOW VARSTATUS worked
           0 - Indicates that "" didn't work.

NBLOCAL - N (where N indicates the number of remote users coming in to
use local files for NFA.)

NBREMOTE - N (where N indicates the number of local users going remote to
use files for NFA.)

NBCLIENT - N (where N indicates the number of NetBase client users
accessing databases on this system.)
```

## START

The **START** command is used to start up NetBase services. These services can be started to specific nodes or to all nodes currently up on the network. **START** also has the ability to initiate the background job that runs NetBase controlling programs. Typically NetBase is configured so that all of the services are available when NetBase is started. If this is not the case, or if any of the processes have been stopped for backup or other purposes, this command may be used to start these services as needed. This command requires **NODE MANAGER(NM)** or **SYSTEM SUPERVISOR(OP)** capability.

If the **STOP** command is used to stop a service for a particular node, for example **STOP POST 2**, that service will remain stopped for that node until an explicit **START** command is issued. This will be the case even if the NetBase background job is stopped and restarted.

Since NetBase requires the network link services, the link must be started (via the **:NETCONTROL** command) before attempting to start NetBase. Both the network link and the loopback network interfaces must be started.

### Syntax

```
*> START {LOCAL      [node]}
         {REMOTE    [node]}
         {NODE      [node]}
         {CLIENT    }
         {RPC       [node]}
         {ALL       [node]}
         {SPOOLING  }
         {SPOOL IN  [node]}
         {SPOOL OUT [node]}
         {SPOOL UNIX}
         {SQL       }
         {SQL IN   node }
         {SQL OUT  node }
         {EXPORT   [node]}
         {IMPORT   [node]}
         {POST     [node]}
         {ILOG     [file]}
         {NETBASE  }
```

## Parameters

Parameter	Description
<b>CLIENT</b>	Starts the processes necessary for NetBase Client to access data on an HP 3000.
<b>EXPORT [node]</b>	Starts the exporting of shadowed transactions. When <b>EXPORT</b> is initially started, exporting to all nodes is enabled regardless of whether a node is specified. If exporting to an individual node had been stopped, it can be restarted by specifying the remote node with the <b>START</b> command.
<b>ILOG [file]</b>	Starts IMAGE log file shadowing for the file specified. If a file is not specified, IMAGE log file shadowing is started for all shadowed IMAGE log files.  IMAGE log file shadowing will either shadow the IMAGE log file or use the information in the log file to send shadow updates to the shadow copy, depending on how the log file is defined in the directory.
<b>IMPORT [node]</b>	Starts the importing of shadowed transactions from the node specified. If the remote node is not specified, importing from all remote nodes is enabled.
<b>LOCAL [node]</b>	Starts access to the specified node by local users. If the node is not specified, access to all remote nodes is enabled.
<b>NETBASE</b>	Starts NetBase and all of its services that are configured to start when NetBase is started.
<b>NODE [node]</b>	Starts access to the remote node by local users. It also allows users on the remote node access to this node. Note that when you add a new node with this command, NetBase will begin communicating to it immediately. You do not need to stop and restart all NetBase jobs.
<b>POST [node]</b>	Starts the posting of shadowed transactions from the node specified. If a remote node is not specified, this command starts the posting of transactions from all nodes.
<b>REMOTE [node]</b>	Starts access of the local node by remote users on the node specified. If the node is not specified, access by all remote nodes is enabled.
<b>SPOOLING</b>	Starts all spooling into and out of the local node.

*Continued*

Parameter	Description
<b>SPOOL IN</b> [ <i>node</i> ]	Starts the process that receives spooling output from the node specified. If the node is not specified, the reception of spooling output from all nodes is enabled.
<b>SPOOL OUT</b> [ <i>node</i> ]	Starts the process that transmits spooling output to the node specified. If the node is not specified, the transmission of spooling output to all nodes is enabled.
<b>SPOOL UNIX</b>	Starts the process that receives spooling output from UNIX nodes.
<b>SQL</b>	Starts the import and export processes for SQL shadowing.
<b>SQL IN</b> <i>node</i>	Starts the importing of SQL shadow transactions from the specified node.
<b>SQL OUT</b> <i>node</i>	Starts the exporting of SQL shadow transactions to the specified node.

## Examples

To launch the NetBase main control process:

```
*> START NETBASE
STREAMING NETBASE #Jn
```

The job that is streamed from this command is copied from the file **NETBASE.JOB.NETBASE.NBCTRL** substitutes in the correct account, group and user passwords from the NetBase configuration file. If **LOCAL** or **REMOTE** services have been stopped to a particular node for any reason, or if the service has not yet been enabled due to configuration parameters, **START** can be used to enable services. For example, to start local services to a remote node:

```
*> START LOCAL 1
```

To start services to remote users of a node:

```
*> START REMOTE 1
```

To start exporting to all nodes:

```
*> START EXPORT
```

or just a specific node:

```
*> START EXPORT 2
```



**Note.** If the export process is down, the node parameter is ignored.

---

## STATS

The **STATS** command enables and disables NetBase statistical logging for the local system. The current log file can also be displayed and changed using this command. Statistics can be enabled for either remote or local files and data bases. This command requires **SYSTEM MANAGER(SM)** or **NODE MANAGER(NM)** capability.

If NetBase is unable to gain exclusive access to the current statistics log file, statistics will be suspended. When this occurs, the **STATS RESUME** command is used to reactivate statistics once the problem has been resolved. The **STATS STOP** command disables statistics capturing for all user processes started after the command is issued. If specified with parameters, only the specified services will no longer be tracked. Note that this command does not close the current log file or stop statistics for existing users.



**Note.** Statistics logging is always inactive when NetBase is started. Response time statistics are currently unavailable on MPE/iX systems.

---

### Syntax

```
*> STATS {START [=option] [,option]...}
          {STOP  [=option] [,option]...}
          {SWITCH}
          {RESUME}
          {SHOW }
```

where

```
option = ALL
        FILES
        IMAGE
        LOCAL
        REMOTE
```

## Parameters

Parameter	Description
<b>START</b>	Enables statistical tracking of user processes. If entered without an option, this command starts the process that gathers statistics. The <b>STATS SHOW</b> command displays the activities that are being tracked. If entered with one or more options, this command starts statistical tracking of the activity or activities specified.
<b>STOP</b>	Disables statistical tracking of user processes. If entered without an option, this command stops statistical tracking for all configured activities. If entered with one or more options, this command stops statistical tracking of the activity or activities specified. Processes which are collecting statistics at the time this command is issued will continue to do so until they terminate.
<b>option</b>	Specifies the activity to be tracked for statistics. Option is one or more of the following: <ul style="list-style-type: none"> <li><b>ALL</b>            The complete set of activities that may be tracked for statistics. This includes local and remote file access, local and remote database access, and process information.</li> <li><b>FILES</b>           The statistical tracking of processes accessing MPE and KSAM files through NetBase. This option applies to both local and remote users unless the <b>LOCAL</b> option or the <b>REMOTE</b> option is also specified.</li> <li><b>IMAGE</b>           The statistical tracking of processes accessing IMAGE databases through NetBase. This option applies to both local and remote users unless the <b>LOCAL</b> option or the <b>REMOTE</b> option is also specified.</li> <li><b>LOCAL</b>           The statistical tracking of local access to files or databases. This option applies to both MPE files and IMAGE databases unless the <b>FILES</b> option or the <b>IMAGE</b> option is also specified.</li> <li><b>REMOTE</b>          The statistical tracking of remote access to files or databases. This option applies to both MPE files and IMAGE databases unless the <b>FILES</b> option or the <b>IMAGE</b> option is also specified.</li> </ul>
<b>SWITCH</b>	Closes the current statistics log file, creates a new log file and starts logging to the new log file.
<b>RESUME</b>	Restarts statistical tracking after statistics have been suspended due to a file system error.
<b>SHOW</b>	Displays current log file information.

## Examples

To show the status of the current log file:

```
*> STATS SHOW

LOGGING STATUS:      ACTIVE
CURRENT FILE:       NBLG0004
RECORDS LOGGED:     1534
PERCENT FULL:       47%

LOGGING OPTIONS:
IMAGE               OFF
FILES               OFF
PROCESS             LOCAL
```

The only option currently active is the tracking of terminal response time. To start statistical tracking for local and remote users of IMAGE databases:

```
*> STATS START=LOCAL,REMOTE,IMAGE
```

Note that the above command enables the statistical tracking of the activities listed without changing the status of the options currently active. To show the changes of status of the current log file:

```
*> STATS SHOW

LOGGING STATUS:      ACTIVE
CURRENT FILE:       NBLG0004
RECORDS LOGGED:     1534
PERCENT FULL:       47%

LOGGING OPTIONS:
IMAGE               LOCAL/REMOTE
FILES               OFF
PROCESS             LOCAL
```

To stop statistical gathering without disabling the current activities being tracked:

```
*> STATS STOP
```

To then start statistics without enabling any new options:

```
*> STATS START
```

To instruct NetBase to close the current log file and open the next:

```
*> STATS SWITCH
```

Since NetBase requires exclusive access to statistic log files, this command is used when the current log file needs to be freed up to produce statistics reports. Note that **STATS STOP** will not close the current log file since processes currently collecting statistics will be posting statistics as the data becomes available.

If NetBase is unable to open the statistics log file, the statistics process will complain to the console and then suspend itself. To reactivate statistics, correct the cause of the problem (i.e. free up or purge the file that NetBase is trying to access) and use the following command:

```
*> STATS RESUME
```

To disable statistics gathering for files only:

```
*> STATS STOP=FILES
```

If **IMAGE** statistics are enabled, they will not be affected by this command.

## STOP

The **STOP** command turns off a particular service or all services to new users requesting that service, and can stop services to a specific node or all nodes. Users that have been granted access to a service before it is stopped are allowed to complete. This command requires **NODE MANAGER(NM)** or **SYSTEM SUPERVISOR(OP)** capability.

If **STOP** is issued for a specific service and a specific node, for example **STOP POST 2**, that service will remain stopped for that node until an explicit **START** command is issued. This is true even if the NetBase background job is stopped and restarted.

For the spooling and shadowing transports, this command causes the particular process stopped to go to a **DOWN** or **STOPPED** status. In addition, the process on the corresponding remote system goes to a **INACTIVE** or **QUEUE** status.

### Syntax

```
*> STOP {LOCAL      [node]}
        {REMOTE    [node]}
        {ALL
        {CLIENT
        {SPOOLING
        {SPOOL IN  [node]}
        {SPOOL OUT [node]}
        {SPOOL UNIX
        {SQL
        {SQL IN    node }
        {SQL OUT   node }
        {EXPORT   [node]}
        {IMPORT   [node]}
        {POST     [node]}
        {ILOG     [file]}
        {NETBASE  }
```

### Parameters

Parameter	Description
<b>LOCAL [node]</b>	Stops access to the specified node by local users. If the node is not specified, access to all remote nodes is disabled.
<b>REMOTE [node]</b>	Stops access of the local node by remote users on the node specified. If the node is not specified, access by all remote nodes is disabled.

*Continued*

Parameter	Description
<b>ALL</b>	Stops all processes but does not terminate the NetBase background job.
<b>CLIENT</b>	Stops the processes necessary for NetBase Client to access data on an HP 3000.
<b>SPOOLING</b>	Stops all spooling into and out of the local node.
<b>SPOOL IN [node]</b>	Stops the process that receives spooling output from the node specified. If the node is not specified, the reception of spooling output from all nodes is disabled.
<b>SPOOL OUT [node]</b>	Stops the process that transmits spooling output to the node specified. If the node is not specified, the transmission of spooling output to all nodes is disabled.
<b>SPOOL UNIX</b>	Stops the process that receives spooling output from UNIX nodes.
<b>SQL</b>	Stops SQL shadowing import and export processes.
<b>SQL IN node</b>	Stops the importing of SQL shadow transactions. Note that specifying the node is required.
<b>SQL OUT node</b>	Stops the exporting of SQL shadow transactions. Note that specifying the node is required.
<b>EXPORT [node]</b>	Stops the exporting of shadowed transactions to the node specified. If the remote node is not specified, exporting to all nodes is disabled.
<b>IMPORT [node]</b>	Stops the importing of shadowed transactions from the node specified. If the remote node is not specified, importing from all remote nodes is disabled.
<b>POST [node]</b>	Stops the posting of shadowed transactions from the node specified. If a remote node is not specified, this command stops the posting of transactions from all nodes.
<b>ILOG [file]</b>	Stops IMAGE log file shadowing for the file specified. If a file is not specified, stop IMAGE shadowing of all shadowed IMAGE log files.
<b>NETBASE</b>	Stops NetBase and all of its services.  <b>WARNING!</b> If shadow entries exist in the directory, NetBase will continue to write shadow transactions to the shadow transport file after NetBase is stopped. If NetBase is being stopped for an extended period of time, make sure that the shadow transport file <b>NBM.IPC.NETBASE</b> is large enough to handle the anticipated amount of shadow transactions, or disable shadowing before stopping NetBase.

## Examples

To stop services to and from a node, first obtain the node number by executing the **SHOW STATUS** command.

```
*> SHOW STATUS
STATUS FOR NODE 1: SYS-A

      NODE      STATUS      ----- NFA -----      -- SPOOLING --      ---- SHADOWING ----
      ACCESS LOCAL REMOTE      EXPORT  IMPORT      EXPORT  IMPORT  POST

2 SYS-B      UP      IN/OUT      22      12              ACTIVE              STOP  STOP
3 SYS-C      UP      IN/OUT      15      10              ACTIVE              STOP  STOP

LOCAL:
REMOTE USERS: 22      LIMIT = 32
LOCAL  USERS: 37      LIMIT = 64
SERVERS:      32      ALLOC = 32
ROLL FORWARD: 0

SERVICES:
STATISTICS:  INACTIVE
SQL SHADOW:  INACTIVE
```

```
*> STOP LOCAL 2
*> SHOW STATUS

STATUS FOR NODE 1: SYS-A

      NODE      STATUS      ----- NFA -----      -- SPOOLING --      ---- SHADOWING ----
      ACCESS LOCAL REMOTE      EXPORT  IMPORT      EXPORT  IMPORT  POST

2 SYS-B      UP      IN      22      12              ACTIVE              STOP  STOP
3 SYS-C      UP      IN/OUT      15      10              ACTIVE              STOP  STOP

LOCAL:
REMOTE USERS: 22      LIMIT = 32
LOCAL  USERS: 37      LIMIT = 64
SERVERS:      32      ALLOC = 32
ROLL FORWARD: 0

SERVICES:
STATISTICS:  INACTIVE
SQL SHADOW:  INACTIVE
```

## TRACE

The **TRACE** command displays the server activity as it occurs. **TRACE** displays on the terminal all **IMAGE** and **File System** intrinsic calls that a server is executing. The display consists of the following information:

- The time the call was executed
- The server ID number
- The intrinsic being serviced

For **IMAGE** calls, it also displays:

- The mode parameter
- The return status
- The fully qualified base name
- The set name or number (in brackets)

For **File System** calls, it also displays:

- The condition word
- The fully-qualified file name

Tracing to the terminal will continue until **<Ctrl-Y>** is entered or the remote user process terminates. This command requires **NODE MANAGER(NM)** or **SYSTEM MANAGER(SM)** capability.



**Note.** This command may slow down the user process associated with the server being traced. Also, the **BREAK** key is disabled while tracing.

---

### Syntax

```
*> TRACE SERVER server-id
```

### Parameters

Parameter	Description
<i>server-id</i>	Actual server number identifying the server to be traced.

## Examples

To display the activity of server 1:

```

:RUN NBCTRL
NetBase Control Program [x.x] Update n (C) QUEST Software 1987

*> SHOW REMOTE

NODE          JOBNUM SRVR SPIN  JOB NAME                PROGRAM NAME
VIXEN        #S216     1   78  TEST,OPERATOR.SYS
BANMAIN.PUB.NETBASE

*> TRACE SERVER 1
TRACING HAS BEEN ENABLED

09:19:32.3 I065 DBINFO    -202= 0   BANNER.DATA.NETBASE
09:19:32.5 I065 DBCLOSE   3   = 0   BANNER.DATA.NETBASE   REPORTS
09:19:32.5 I065 DBGET     2   = 0   BANNER.DATA.NETBASE   REPORTS
09:19:32.6 I065 DBINFO    -202= 0   BANNER.DATA.NETBASE
09:19:32.6 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   HEADERS
09:19:33.8 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   HEADERS
09:19:37.0 I065 DBGET     2   = 0   BANNER.DATA.NETBASE   REPORTS
09:19:37.0 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   HEADERS
09:19:45.0 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   HEADERS
09:19:46.4 I065 DBLOCK    3   = 0   BANNER.DATA.NETBASE
09:19:46.5 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   REPORTS
09:19:46.5 I065 DBUPDATE  1   = 0   BANNER.DATA.NETBASE
09:19:46.6 I065 DBUNLOCK  1   = 0   BANNER.DATA.NETBASE
09:19:53.5 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   REPORTS
09:19:53.5 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   HEADERS
09:20:03.4 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   HEADERS
09:20:05.9 I065 DBLOCK    3   = 0   BANNER.DATA.NETBASE
09:20:06.0 I065 DBGET     7   = 0   BANNER.DATA.NETBASE   REPORTS
09:20:06.0 I065 DBUPDATE  1   = 0   BANNER.DATA.NETBASE
09:20:06.1 I065 DBUNLOCK  1   = 0   BANNER.DATA.NETBASE
09:20:14.0 I065 DBCLOSE   1   = 0   BANNER.DATA.NETBASE

SERVER 1 HAS TERMINATED

```

## UNDEFER

This command posts all deferred updates to shadowed files. **UNDEFER** starts the posting of shadowed transactions from the specified node into shadowed files previously deferred with the **DEFER** command. Once all deferred transactions have been applied, new transactions will post directly for those files. This command requires **SYSTEM SUPERVISOR(OP)** or **SYSTEM MANAGER(SM)** capability.

### Syntax

```
*> UNDEFER node
```

### Parameters

Parameter	Description
<i>node</i>	The NetBase node number or name of the importing node.

### Examples

To restart the direct posting of files that have been previously deferred for node **FOX**:

```
:RUN NBCTRL
```

```
NetBase Control Program [x.x] Update n (C) QUEST Software 1987
```

```
*> UNDEFER FOX
```

```
DEFERRAL ROLL FORWARD STARTED
```

```
*> SHOW POST
```

```
Node  Name      Status      Backlog  Files  Closes
  1    FOX        IDLE
      Defer roll forward is in progress
  2    VIXEN      IDLE
                                     0
```

## CHAPTER 3

# Using CONFIG

Since all network environments are not the same, NetBase allows for the entry and modification of NetBase operating parameters. NBCTRL has a subsystem dedicated to installing and maintaining the configuration and operating environment of NetBase. In this chapter, the CONFIG subsystem commands and their use with configuration parameters are described.

The configuration parameters are stored in files in the NETBASE account. The subsystem will automatically create files when needed. Commands are provided that allow the configuration to be listed and changed.

### In This Chapter

- **CONFIG Command Summary**
- **Configuring NetBase**
- **Special Command Features**
- **NetBase Parameter Definitions**
- **CONFIG Commands**

## CONFIG Command Summary

The following table lists the **CONFIG** commands.

Command	Function
<b>C&gt; ADD</b>	Adds a node to the configuration.
<b>C&gt; DELETE</b>	Deletes a node from the configuration.
<b>C&gt; EXIT</b>	Exits from the configuration subsystem, and returns to <b>NBCTRL</b> .
<b>C&gt; HELP</b>	Displays a list of the <b>CONFIG</b> subsystem commands and the syntax for each command.
<b>C&gt; LIST</b>	Lists the configuration information.
<b>C&gt; MODIFY</b>	Changes the configuration parameters.
<b>C&gt; PRINT</b>	Redirects output of the <b>LIST</b> command to the printer.

## Configuring NetBase

Configuring NetBase consists of two parts, configuring the operational parameters and defining the network nodes for NetBase. The configuration subsystem can only be entered through **NBCTRL**. Configuration changes may require the creation of files that must reside in the **NETBASE** account. Access to **CONFIG** is denied if the user is not logged into the **NETBASE** account or does not have **SM** capability. Use the following steps to configure NetBase.

### To configure the operational parameters:

1. Run **NBCTRL** by typing:

```
: RUN NBCTRL.PUB.NETBASE
*-> CONFIG
C>
```

2. You may review each of the configuration parameters, or you may use the **QUICK** option to highlight the essential parameters. As NetBase takes you through to specific parameters, you can press **Enter** to keep the displayed values, or you can enter a new value.

```
C> MODIFY QUICK
```

**To define the network nodes:**

1. To define the nodes in your network for NetBase, use the **ADD** command, specifying the number of the node you wish to define. Remember to define the current node. You are prompted for details concerning each node.

```
C> ADD n
```

2. Tell NetBase which node is the local node:

```
C> MODIFY GLOBAL
```

3. Exit **CONFIG**:

```
C> EXIT
*> EXIT
```

4. Repeat steps 1 to 3 on each node.



**Note.** As an alternative, you can run **NBCTRL** with an **INFO** parameter. This allows you to access **CONFIG** when someone else is accessing **NBCTRL**.

```
:RUN NBCTRL . PUB . NETBASE ; INFO="CONFIG"
```

---

---

## Special Command Features

The subsystem command processor has some special features that provide MPE functionality within NetBase. Any MPE command that is executable in break may be issued from the subsystem. To enter an MPE command, type the command prefixed by a colon (:). This subsystem requires **NODE MANAGER(NM)** and **SYSTEM MANAGER(SM)** capabilities. The user must be logged into the **NETBASE** account.

The **REDO** command allows the last command entered to be modified and then executed. To modify and execute the last typed command, simply type **REDO** at the **C>** prompt, modify the line using standard **EDITOR** commands, and press return to execute the command.

The **CONFIG** subsystem also supports a help facility. Typing **HELP** displays a list of valid commands. The following special characters are valid while modifying a configuration entry:

Character	Description
//	Terminates command and abort changes.
/.	Backs up to previous field.
/	Terminates command and save changes.
?	Displays help information on current field.

## NetBase Parameter Definitions

The following section describes the parameters used by NetBase to function, as well as various tuning options. To guarantee a successful implementation of NetBase, it is recommended that you study this section and become familiar with these parameters.

There are two sets of parameters. The largest set of parameters defines the operational environment for NetBase on the local node. These parameters are broken down further into the following subsets: **GLOBAL**, **NFA**, **SPOOL**, **STATS**, **DIR**, **SHADOW**, **SQL**, and **PASSWORD** parameters. Another set of parameters defines specific information about each of the nodes in the network that will interact with the local node. The node level parameters also include **SPOOL** and **SHADOW** parameters.

When NetBase is installed, the parameters that define the local node are initialized to their default settings. The system may then be tuned by modifying various parameters in the configuration subsystem. The set of parameters that are defined for a system depends on which NetBase features are included in the installation.

If Network File Access is installed, the **NFA** parameters need to be defined. **NFA** parameters specify configuration limits required for remote file access. If Statistics is being installed, the **STATS** parameters may need to be specified. These parameters are used by NetBase's statistics processes. NetBase shadowing requires that the **SHADOW** parameters be defined both on the general and the node levels. Similarly, NetBase spooling also requires that the **SPOOL** parameters be defined both on the general and the node levels.

To assist in initial installation, a specialized subset of the most immediately meaningful parameters from all subsets have been grouped under the subset **QUICK**.

---

## Operational Parameters

The following parameters define general information regarding the operation of NetBase:

### GLOBAL - Local Node Number

An arbitrary number between 1 and 255 designating the number of this node. Each node in the NetBase network must be assigned a different node number. This value is used to determine which files are local and which are remote. This parameter takes effect when NetBase is started.

**Default** = 1  
**Maximum** = 255

### GLOBAL - Error Enhancements

A string specifying the type of display enhancements that will be used when sending a fatal NetBase error to the console. The string may consist of the word **NONE** or one or more of the following characters:

Character	Description
<b>H</b>	Half bright
<b>I</b>	Inverse video
<b>U</b>	Underline
<b>B</b>	Blinking
<b>R</b>	Ring the bell when printing the message

If the string consists of the word **NONE**, error messages will be displayed with no display enhancements. If the console on the local node is a non-HP terminal, only **R** will apply. This parameter takes effect when new processes are started.

**Default** = **IR**

### GLOBAL - Maximum Local Users

The maximum number of concurrent local user processes that may access NetBase at any one time. Each user process accessing at least one remote file or for who statistics are being collected is considered a local user. The maximum is **2048** user processes. Local users do not include processes that are using the **SHADOWING** subsystem only. This parameter takes effect when NetBase is started.

**Default = 0**  
**Maximum = 2048**

### NFA - Maximum Remote Users

The maximum number of concurrent remote users that may access this node at one time. Again, each remote process accessing at least one local file is considered a remote user. The maximum is **2500** remote user processes. This parameter takes effect when NetBase is started.

**Default = 0**  
**Maximum = 2500**

### NFA - Release Server on Last Close

If this parameter is set to **Y**, the server allocated to a remote process will be released when the last open remote file is closed. If this parameter is set to **N**, the server is not released until the remote process terminates. This parameter takes effect when new user processes are started.

**Default = N**

### NFA - Initial Server Allocation

The number of remote server processes to pre-allocate. When NetBase is started, it creates and suspends this number of server processes. When a new remote process attempts to access the local node, NetBase merely “wakes up” one of the suspended servers. However, when all the servers are assigned to remote processes, or if this parameter is set to zero, new server processes must be created.

Pre-allocating servers reduces system overhead. This may be set to any value between 0 and the maximum number of remote users. The **NBCTRL SET** command can be used to dynamically alter this value. This parameter takes effect when NetBase is started.

**Default = 0**

### NFA - Server Allocation Time-out

This is the number of seconds that NetBase will wait for a remote server to be allocated to a user process. The default is **60** seconds. If the system is heavily loaded, and server allocation time-outs are being experienced, this limit should be increased. This number must be between **20** and **600** seconds. This parameter takes effect when new server processes are started.

**Default = 60**

### NFA - Server Initial Priority

The priority queue in which a server process will run when a user is first activated. Entering **SAME** will cause the server to be scheduled in the same queue as the originating process. This parameter takes effect when NetBase is started.

**Default = CS**

### NFA - Server LIB= load flag

This parameter specifies the library to be loaded when the servers are created. Valid values are **S**, **P** and **G** for the system, public and group libraries respectively. Normally, the servers should be loaded with the system library. This parameter takes effect when NetBase is started.

**Default = P**

### NFA - Lock Priority Boosting

This parameter enables users locking remote files to also lock a local file. This causes MPE to boost the priority of low priority processes that hold a lock that a higher priority process is waiting for. This should be enabled if user processes are waiting because a batch job is holding a remote lock. This parameter takes effect when a user process is started.

**Default = N**

### NFA - Client Inactivity Timeout

This parameter sets the time-out value (in seconds) for the maximum time a server can be inactive before it terminates the connection to a client. Inactivity is defined as having no calls made to the server process. The range of values is **0** to **3276** seconds. Setting this value to **0** disables the time-out. This parameter takes effect when new server processes are allocated.

**Default = 0 (seconds)**

**Maximum = 3276 (seconds)**

### NFA - Client Lock Held Time-out

This parameter sets the maximum time a server will wait for any activity before it terminates the connection. Unlike the **Client Inactivity Time-out**, this parameter takes effect only when the client holds a lock. When all locks are released, the **Client Inactivity Time-out** parameter takes effect. If this parameter is set to **0**, and the **Client Inactivity Time-out** parameter is non-zero, then the later parameter will be used. This parameter takes effect when new server processes are allocated.

**Default = 0 (seconds)**  
**Maximum = 3276 (seconds)**

### NFA - Incoming Users on startup

This parameter tells NetBase whether to allow remote users access when NetBase is started. If the parameter is set to **N**, the **START REMOTE** command must be issued before remote users are allowed access to the local node. For operational simplicity, it is recommended that this parameter be set to **Y**. This parameter takes effect when NetBase is started.

**Default = Y**

### NFA - Outgoing Users on startup

This parameter tells NetBase whether to allow local users access to remote computers when NetBase is started. If it is set to **N**, the **START LOCAL** command must be issued before local users may access remote computers. For operational simplicity, it is recommended that this parameter be set to **Y**. This parameter takes effect when NetBase is started.

**Default = Y**

### NFA - Client Access on startup

This parameter instructs NetBase whether to allow remote client access to this system when NetBase is started. If this parameter is set to **N**, it will be necessary to issue a **START CLIENT** command in **NBCTRL** to allow access. This parameter takes effect when NetBase is started.

**Default = N**

### SPOOL - Remote Transfer Time-out

This parameter specifies the number of seconds that the NetBase Spooling transfer program will wait to establish communication with its remote spooling process. The minimum value is **1.0** with a maximum of **3276**. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = 120**

### SPOOL - Transfer Server Allocation

This parameter determines the number of spool file transfer servers to pre-allocate. Spool transfer servers are invoked when using the NBSpool **MOVE** or **COPY** commands to transfer spool files into this node. Pre-allocating transfer spoolers speeds up the initialization process and reduces the overhead caused by creating the spoolers at run time. This parameter should be set to the expected maximum number of incoming spool file transfer requests. This parameter takes effect when NetBase is started.

**Default = 1**  
**Maximum = 64**

### SPOOL - Maximum Block (bytes)

This parameter determines the number of bytes of spooled output that NetBase will buffer before sending the output to the spooler transport. The larger this parameter, the fewer the network transactions. The maximum value of **8192** is recommended. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = 8192**  
**Minimum = 512**

### SPOOL - Maximum Files per Node

This parameter represents the maximum number of local spool files that may concurrently receive data from remote machines. A maximum of **256** is permitted. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = 64**

### SPOOL - Use S' and J' on incoming

This parameter tells NetBase to set the job/session id numbers on incoming spool files to **J'nnn** or **S'nnn**. This convention reduces ambiguity between local and remote spool files. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = Y**

### SPOOL - Change user on incoming

Setting this parameter to **Y** preserves the user information from the originating computer. If it is set to **N**, all transferred spool files are created with the user specified as **MGR.NETBASE**. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = Y**

### SPOOL - Change jsname to node name

If the Change user on incoming flag is set to **Y** and this value is set to **Y**, the job/session name of the creating user will be changed to the node name where the creating user exists. The job/session name can be displayed with NBSpool and other spooling utilities. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = N**

### SPOOL - Using SpoolMate

This parameter should be set to **Y** when importing spool files to a system that uses *Unison's SpoolMate* software. In all other cases, set this value to **N**. This parameter takes effect when the **START SPOOLING** command is issued in **NBCTRL**.

**Default = N**

### SPOOL - Spooling Export Priority

This parameter sets the execution priority of all Spooling Export processes. The default is linear queue **152**. Any numeric value or **BS**, **CS**, **DS**, or **ES** is valid. This parameter takes effect when a Spooling Export process starts.

**Default = 152**

**Minimum = 50**

**Maximum = 255**

### SPOOL - Spooling Import Priority

This parameter sets the execution priority of all Spooling Import processes. The default is linear queue **152**. Any numeric value or **BS**, **CS**, **DS**, or **ES** is valid. This parameter takes effect when a Spooling Import process starts.

**Default** = 152  
**Minimum** = 50  
**Maximum** = 255

### SPOOL - Enable HP-UX Receiving

If this parameter is set to **Y**, NetBase Spooling from HP-UX will be available. This parameter does not effect spooling to HP-UX. This parameter takes effect when NetBase Spooling is started.

**Default** = N

### SPOOL - Enable BOOTP Daemon

This parameter causes the **BOOTP** daemon to be started when NetBase Spooling is started. This functionality is only available on MPE/iX 4.0 or later. **BOOTP** is used to configure LAN-based devices such as printers. This parameter takes effect when NetBase Spooling is started.

**Default** = N

### SPOOL - Enable LPD Daemon

This parameter causes the **LPD** daemon to be started when NetBase Spooling is started. The **LPD** daemon is used to receive spool files from various systems, including UNIX-based systems. This parameter takes effect when NetBase Spooling is started.

**Default** = N

### SPOOL - Spooling on startup

If this parameter is set to **Y**, network spooling is operational as soon as NetBase is started. If it is set to **N**, spooling must be initialized using the **START** command in **NBCTRL**. This parameter takes effect when NetBase is started.

**Default** = Y

### STATS - Max File Allocation

This parameter sets the maximum number of files accessed by a local process that can be monitored for statistics. This includes local and remote files and databases. Valid values are 1 to 145. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 64  
**Maximum** = 145

### STATS - Init File Allocation

This parameter is used to compute the initial size of the statistics data segment. To determine this number, estimate the maximum number of files and databases that any local process will usually have open concurrently. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 24  
**Maximum** = 255

### STATS - Expansion Increment

When the statistics segment fills up, it is expanded by a specified amount. This parameter, which represents the number of new files that can be monitored, is the increment by which the segment will be expanded. This number must be between 1 and 32. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 12  
**Maximum** = 32

### STATS - Log File Limit

This parameter is the limit (in number of records) of a statistics log file. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 2048  
**Maximum** = 32,767

### STATS - Read-Read Filter (ms)

This parameter is the threshold above which read-read transaction time statistics are captured. If a read-read response time is less than this value, that particular transaction is ignored for statistics gathering. The measurement is in milliseconds. A value of **0** disables the filtering. The maximum is **2,147,483,647**. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 0  
**Maximum** = 2,147,483,647

### STATS - Read-Write Filter (ms)

This parameter is the threshold above which read-write response time statistics are captured. If a read-write response time is less than this value, that particular response is ignored for statistics gathering. The measurement is in milliseconds. A value of **0** disables the filtering. The maximum is **2,147,483,647**. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 0  
**Maximum** = 2,147,483,647

### STATS - Low User Response Filter

This parameter is the threshold above which user response time statistics are captured. If the time between the start and finish of the terminal read is less than this value, that particular response is ignored for statistics gathering. The measurement is in milliseconds. A value of **0** disables the filtering. The maximum is **2,147,483,647**. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 0  
**Maximum** = 2,147,483,647

### STATS - High User Response Filter

This parameter is the threshold below which user response time statistics are captured. If the time between the start and finish of the terminal read is greater than this value, that particular response is ignored for statistics gathering. The measurement is in milliseconds. A value of **0** disables the filtering. The maximum is **2,147,483,647**. This command takes effect when the **STATS START** command is issued in **NBCTRL**.

**Default** = 0  
**Maximum** = 2,147,483,647

### DIR - Default Directory Filename

The name of the file that will be used as the NetBase directory for all processes on this node. This parameter takes effect as new user processes are started or the **START POST** command is issued in **NBCTRL**.

**Default = NBD.DATA.NETBASE**

### DIR - Allow File Equates

If set to **Y**, the NetBase directory may be redirected from the default directory with a file equation. Recommended value is **N**. This parameter takes effect as new user processes are started.

**Default = N**

### DIR - Allow Wildcards

Setting this parameter to **Y** enables the NetBase directory wildcard feature. This allows you to specify files and programs within the NetBase directory using the **@** wildcard. The **@** wildcard may only be used to replace complete file and group names. This parameter takes effect as new user processes are started.

**Default = Y**

### DIR - Map Local Files with Dir

Normally, the remapping of a filename from the **LOCAL REFERENCE** to the **REMOTE EQUIVALENT** applies only to remote files. Setting this parameter to **Y** causes NetBase to use the **REMOTE EQUIVALENT** on local files also. Wildcards may be used to remap local files.

**Default = N**

Enabling this parameter affects all file definitions. This option can be implemented more effectively through use of the **;MAP** option of the directory's **FILE** command. Turn to Chapter 4 "NBDIR" for more information on the **FILE** command and the **;MAP** option. This parameter takes effect as new user processes are started.

### SHADOW - Close Log file on suspend

This parameter should be set to **Y** if a very large (greater than **384**) number of processes are updating shadowed files. This flag will cause the shadow transaction queue file to be closed if a process suspends, thus allowing more user processes to access the input queue file. This parameter takes effect as new user processes are started.

**Default = N**

### SHADOW - File Locking Required

This parameter prevents applications from updating flat files if they are not locked or accessed exclusively. Setting this parameter to **Y** will generate a file system error if an application attempts to update a file without adequate locking. The file system error generated is **1041** or **1042**. If this error occurs, you must add the **;SYNC** option to the **FILE** statement for the file being shadowed. This will cause NetBase to use a locking strategy for you. Chronic sync errors can often be found and eliminated by enabling this parameter. This parameter takes effect as new user processes are started.

**Default = N**

### SHADOW - Flush Shadow File Posts

Setting this parameter to **Y** causes NetBase's posting process to flush all updates of files to disk after posting the transaction. This parameter is only applicable to MPE and KSAM files. If this parameter is set to **N**, user processes on the shadow machine will not immediately see updates posted by the shadow posting process. This parameter takes effect when the **START POST** command is issued in **NBCTRL**.

**Default = N**

### SHADOW - Errors Before Disable

This parameter determines the number of synchronization errors NetBase's posting process will ignore before disabling a file or data set for shadow posting. Setting this parameter to **0** causes NetBase to ignore all synchronization errors. This parameter takes effect when the **START POST** command is issued in **NBCTRL**.

**Default = 1**

**Maximum = 10000**

### SHADOW - Check KSAM record counts

This parameter enables a further sync check on KSAM files to ensure that no records are missing when writing new records to a KSAM file. This parameter should only be set to **N** if a master system has a version of NetBase earlier than 9.5. Existing users have this option off by default. It cannot be enabled until all master systems have been installed with the 9.5 NetBase shell. This parameter takes effect when the **START POST** command is issued in **NBCTRL**.

**Default = Y**

### SHADOW - Disable file autoresync

This parameter disables the auto synchronization of files that occurs when a file is purged or erased. Normally, out of sync or discarding files will be put back into sync when the master copy is erased or purged. This parameter takes effect when **POST** is started.

**Default = N**

### SHADOW - Log Post Errors

When this parameter is set to **Y**, all synchronization errors will be logged to a file **PERRnnnn.DATA.NETBASE**, where **nnnn** is the zero filled node number. This file is created by **NBPOST** when the first error occurs. In order to facilitate this, build the file ahead of time using the following syntax: **BUILD PERR####.DATA.NETBASE;MSG**. The specification for this file is in **PERRSPEC.DOC.NETBASE**. This parameter takes effect when the **START POST** command is issued in **NBCTRL**.

**Default = N**

### SHADOW - Enable Quiet Period Flush

When this parameter is set to **Y**, it causes the shadow posting processes to keep track of the number of TurboIMAGE dynamic transactions in process and to stop when the number of transactions in process for all data bases is zero. This parameter takes effect when **POST** is started.

**Default = N**

### SHADOW - Export User Exits

When this parameter is set to **Y**, the user-defined Export User Exit procedure will be called for each shadow record processed. This procedure can be used to filter exported records and to change the destination of each record. This parameter takes effect when **EXPORT** is started.

**Default = N**

### SHADOW - Export Bridge Exits

When this parameter is set to **Y**, the Export Bridge Exit procedure will be called for each shadow record processed. This procedure can be used to remap exported records and to log transactions to the Bridge logfile. This parameter takes effect when **EXPORT** is started.

**Default = N**

### SHADOW - Export Process Info

This parameter controls the exporting of Process Information records to remote nodes. It should only be enabled if the **Shadowing Post User Exit** facility is being used. Otherwise, set the value to **N**. This parameter takes effect as each user process is created.

**Default = N**

### SHADOW - Process Info to Exp Exit

This parameter causes process information to be passed to an export user exit. There is performance degradation when this option is enabled. This parameter takes effect as each user process is created.

**Default = N**

### SHADOW - Export Buffer Size (b)

This parameter defines the buffer size, in bytes, that the export process will use to send update transactions to various nodes. The optimal setting for NS LAN Link is **1457** bytes. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 1457**

**Minimum = 512                    Maximum = 8192**

### SHADOW - Export Buffer Max BF

This parameter defines the maximum number of blocks that the export process can send to a node at one time. The optimal setting for the NS LAN Link is **5** blocks. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 5**  
**Minimum = 1**                      **Maximum = 16**

### SHADOW - Export Buffer Time-out

This parameter is the number of seconds that the export process will wait before sending the buffered shadow transaction to the shadow node. Increasing this delay increases the chances of multiple shadow updates sharing the same network transaction, thus reducing networking overhead. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 10**  
**Maximum = 120**

### SHADOW - Export Ack Time-out

This parameter is the number of seconds that the export process will wait for an acknowledgment from the import process indicating that the data has been received. A value of **120** seconds has been found to be adequate. However, if time-outs occur due to heavily loaded computers or due to unusual activity, this value should be increased. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 120**  
**Maximum = 1200**

### SHADOW - Export Initial Priority

This parameter is the MPE process priority under which the export process will initially run. Valid values are **BS**, **CS**, **DS** and **ES**. This parameter takes effect when NetBase is started.

**Default = DS**

### SHADOW - Export Upshift Priority

This parameter is the MPE process priority that the export process will shift into should exporting fall behind. The export process determines it has fallen behind when the number of records in the export queue exceeds the **upshift** trigger. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = CS**

### SHADOW - Export Upshift Trigger

When the number of records in the export queue exceeds this number, the export process will automatically reschedule itself into the queue specified by the **upshift** priority. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 1500**

### SHADOW - Export Downshift Trigger

This parameter is used by NetBase when the export process is in the upshift queue due to having exceeded the **upshift** trigger. When the number of records in the export queue falls below this number, the export process will automatically reschedule itself into the queue specified by the **Export Initial Priority** parameter. This value should be between **0** and the **upshift** trigger. The **upshift** and **downshift** triggers should be set far enough apart to prevent constant priority shifting. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 100**

### SHADOW - Shift Check Interval

This parameter determines how often the export process should check the export queue to see if it is backlogged. This parameter is specified in number of records. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 50**

### SHADOW - Disable Shift Messages

This disables the upshift and downshift messages that are normally generated by the **EXPORT** process. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = N**

### SHADOW - Enable Export Logging

This flag enables the transaction logging feature of shadowing. When enabled, all shadow transactions will be logged to a series of files in the **LOG** group. These files can be used for roll forward recovery on any subset of the shadowed files. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = N**

### SHADOW - Export Log File Size

Determines the size of the shadow log files, in blocks. Each log file block is 64 sectors and can hold as many as 100 shadow transactions. When a log file fills up, a new log file will be created automatically. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 0**  
**Maximum = 2,147,483,647**

### SHADOW - Export Log File Timeout

This parameter determines how often to flush a log file's buffers to disc. This will protect a log file from system crashes. The value for this parameter is the number of transactions that will be added to the log file between flushes. The log file is also flushed to disc when the inactivity timer expires. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default = 100**

### SHADOW-Queue Full Warn Time-out

This parameter enables a timer that causes warning messages to display on the system console if the shadow queue file (**NBM**) fills up. These messages will alert the system operator to correct the problem. Each user process that encounters this condition will emit one message. This parameter takes effect when new user processes are created.

**Default = 180**

### SHADOW - Export Inactivity Time-out

This parameter is the number of seconds that the export process will wait before flushing the queued shadow updates to disk and updating the queue files EOF. This parameter takes effect when the **START EXPORT** command is issued in **NBCTRL**.

**Default** = 5  
**Maximum** = 120

### SHADOW - Import Inactivity Time-out

This parameter is the number of seconds that the import process will wait before flushing any queued shadowed updates to disk and updating the queue files EOF. This parameter takes effect when the **START IMPORT** command is issued in **NBCTRL**.

**Default** = 5  
**Maximum** = 120

### SHADOW - Post Flush Time-out

This parameter is the number of seconds that the posting process will wait before flushing any queued shadowed updates to disk and updating the queue files EOF. This parameter takes effect when the **START POST** command is issued in **NBCTRL**.

**Default** = 5  
**Maximum** = 120

### SHADOW - Export on Startup

If this parameter is set to **Y**, the export process will be started when NetBase is started. This parameter takes effect when NetBase is started.

**Default** = **Y**

### SHADOW - Import on startup

If this parameter is set to **Y**, the import process will be started when NetBase is started. This parameter takes effect when NetBase is started.

**Default** = **Y**

### SHADOW - Post on startup

If this parameter is set to **Y**, the posting process will be started when NetBase is started. This parameter takes effect when NetBase is started.

**Default = Y**

### SHADOW - Use IMAGE Log Files

This flag activates the IMAGE log file transporter. This creates additional processes used solely for the shadowing of IMAGE log files. Do not set this to **Y** unless at least one IMAGE log file is enabled for shadowing. This parameter takes effect when NetBase is started.

**Default = N**

### SQL - Enable SQL Shadowing

This flag enables the SQL shadowing feature of NetBase. This parameter takes effect when NetBase is started.

**Default = N**

### AUTORPM - Use local logon

When this parameter is set to **Y** and a **LOGON** entry is made that does not contain any passwords, NetBase will use the local users' passwords for the remote logon. This parameter takes effect when a user process is started.

**Default = N**

### PASSWORD - User Password

This parameter is used by the **START NETBASE** command in **NBCTRL** to substitute the user password in the **JOB** file before it streams the **NETBASE** job. A password can be removed by typing **NONE**.

**Default = NONE**

### **PASSWORD - Group Password**

This parameter is used by the **START NETBASE** command in **NBCTRL** to substitute the group password in the **JOB** file before it streams the **NETBASE** job. A password can be removed by typing **NONE**.

**Default = NONE**

### **PASSWORD - Account Password**

This parameter is used by the **START NETBASE** command in **NBCTRL** to substitute the account password in the **JOB** file before it streams the **NETBASE** job. A password can be removed by typing **NONE**.

**Default = QUEST**

## Node Level Parameters

The following parameters define specific information regarding each node in the network:

### System Name

This is a node level parameter that is used to describe the node in NetBase displays, reports and commands. The name can be up to eight characters long. For operational simplicity, it is recommended that this be the same as the **NS** node name.

**Default = none**

**Required Parameter**

### Startup Access (S/I/O/A/W)

This parameter tells NetBase what kind of access to allow this node upon startup. Valid values are:

Value	Description
<b>S</b>	Stopped
<b>I</b>	Incoming
<b>O</b>	Outgoing
<b>A</b>	All
<b>W</b>	Wait

It is recommended that this parameter be set to **A**. If it is not set to **A** or to **W**, the **START** command must be used to change access for the node. If this parameter is set to **W**, the communication line to this node will not be established until the first time a file on this node is accessed by another node.



**Note.** The Wait option requires that the **SPOOL - Direction** node parameter and the **SHADOW - Direction** node parameter be set to **N**. This parameter must be set to **W** if the **SHADOW - Transfer Mode Only** feature is to be used.

---

**Default = A**

## Node Name

This is the NS node name of the specified node. It should be fully qualified if its domain and organization differ from the local node.

**Default = none**                      **Required Parameter**

## Node Connection Time-out

When a communication line is down, NetBase will check periodically to see if the line has come up. Once the link between two nodes has been established, NetBase will further check the link if the link has not been used for a set period. This value sets the time-out interval (in seconds) for these two operations.

**Default = 180**  
**Minimum = 30**  
**Maximum = 600**

## SPOOL - Direction (N/E/I/B)

This parameter describes the direction that spool files are to be transported *relative to this node*. Valid values are:

Value	Description
<b>N</b>	None.
<b>E</b>	Export to this node from the local node only.
<b>I</b>	Import from this node to the local node only.
<b>B</b>	Both import and export from this node to the local node.

When defining the local node, this parameter will be set to **N**. (If this node has been defined as the local node via the **GLOBAL** parm, the spooling parms are not displayed.)

**Default = N**

## SPOOL - Allow incoming job streams

This parameter allows users on the specified node to move input spool files (i.e. job streams) to the local computer without establishing a remote session. The remote user must have **OP** capability to execute this command. Setting this parameter to **Y** may cause a breach in security.

**Default = N**

### SPOOL - Enable Export Flow Control

Normally, the Spooling Export process sends data as quickly as the network can receive the data. If a network link is not a LAN link, or if the destination node is a slower computer, this option may be necessary to prevent the link from experiencing load related failures. This parameter takes effect when **Spooling Exporting** is started.

**Default = N**

### SHADOW - Direction (N/E/I/B)

This parameter describes the direction that files are shadowed *relative to this node*. Valid values are:

Value	Description
<b>N</b>	None
<b>E</b>	Export only
<b>I</b>	Import only
<b>B</b>	Both import and export

When defining the local node this parameter is typically set to **N**, unless it is desired to shadow files back to the same node, in which case, this parameter should be set to **B**.

**Default = N**

### SHADOW - Export queue size

This parameter is used to set the limit of the export queue files built by NetBase in blocks. Note that since the queue files are message files (variable length), the limit does not necessarily correspond to the actual number of shadow transactions queued. Each block will hold 16K bytes which is approximately 50 transactions.

**Default = 3172**

### SHADOW - Import queue size

This parameter is used to set the limit of the import queue files built by NetBase in 16K blocks.

**Default = 3172**

### SHADOW - Post queue size

This parameter is used to set the limit of the posting queue files built by NetBase in 16K blocks.

**Default = 3172**

### SHADOW - Initial Post Priority

This is the MPE process priority under which the posting process will initially run. Valid options are **BS**, **CS**, **DS** or **ES**.

**Default = CS**

### SHADOW - Post Inactivity Timeout

This parameter specifies how long a file will remain open by the post process after the last update is processed. If no transactions for a file are received during this time span, the file will be closed and removed from the **SHOW POST** listing. TurboIMAGE databases are not affected by this parameter. A value of **0** disables this feature.

**Default = 0**  
**Maximum = 3600 seconds**

### SHADOW - Enable Post Logging

This enables transaction logging for the posting process. This should only be set upon the request of Quest Software since this feature is used for tracing only.

**Default = N**

### SHADOW - Log all Post Transactions

This parameter causes all transactions from a specific node to be logged by **NBPOST**. It is generally only used as a debugging tool. This parameter takes effect when posting is started.

**Default = N**

### SHADOW - Save Process Info on Post

This flag is used in conjunction with **Post User Exits** to pass process info from the originating process to the user exit procedure. The originating node must also have turned on the **Export Process Info** parameter. Do not enable this feature unless **Post User Exits** are being used. This parameter takes effect when a post process is started.

**Default = N**

### SHADOW - Abort Post on Errors

This parameter causes the posting process to terminate immediately if a sync error or an error causing a discard has occurred. This allows corrective action to be taken without discarding a transaction. If the error causing the abort is not corrected, **Post** will abort when it is restarted. This parameter takes effect when a post process is started.

**Default = N**

### SHADOW - Buffered Exporting

This parameter enables the buffering of shadowed transactions. This feature causes the export process to wait a configured amount of time before transporting in an effort to share a single network transaction across several shadow update transactions, significantly reducing Network and CPU overhead.

**Default = Y**

### SHADOW - Transfer mode only

This parameter enables **Shadow Transfer Only Mode**. This mode changes the behavior of **EXPORT** to transfer in bulk only and never go into the **ACTIVE** state. This must be set to **Y** on both nodes in order to enable this feature. This parameter is mainly useful for dial-up connections. This feature requires that the **Startup Access** node parameter be set to **W** and the **SHADOW - Direction** node parameter set to **B**. It takes effect when NetBase is started.

**Default = N**

### SHADOW - Transfer Single Queues

This parameter causes shadow **EXPORT** to transfer queued shadow data a queue file at a time. This is useful for transferring queued shadow information across network links that are likely to experience failures. This parameter takes effect when **Exporting** is started.

**Default = N**

### SHADOW - Transfer ACK Frequency

This parameter enables acknowledgments during a shadow transfer. This feature should only be used if communication problems occur frequently during transfers. The lower the value, the slower the transfer will go. Try **4** at first, then raise or lower it if desired. A value of **0** disables this feature. This parameter takes effect when **Exporting** is started.

**Default = 0**  
**Maximum = 256**

### SHADOW - Post Directory Filename

This parameter allows the posting process to use a different NetBase directory than other processes on the system.

**Default = (Whatever is specified under DIR - Default Directory Filename)**

## CONFIG Commands

The following sections describe the **CONFIG** commands.

### ADD

The **ADD** command adds a node to NetBase's configuration file. All node specific parameters such as the node name and NetBase shadowing options are specified here. NetBase should be stopped before adding a node. This command may be abbreviated to **A**.

### Syntax

```
C> ADD node#
```

### Parameters

Parameter	Description
<b>node#</b>	The number of the node to be added to the configuration file.

## Example

To add node 1 to NetBase's configuration:

```
C> A 1
System Name..... SYS-A
Startup Access (S/I/O/A/W)..... [A]
Node Name..... SYS-A
Node Connection Timeout..... [180]
SPOOL - Direction (N/E/I/B)..... [N] B
SPOOL - Allow incoming job streams. [N]
SPOOL - Enable Export Flow Control. [N]
SHADOW - Direction (N/E/I/B)..... [N] I
SHADOW - Export queue size..... [4096]
SHADOW - Import queue size..... [4096]
SHADOW - Post queue size..... [4096]
SHADOW - Initial Post Priority.... [DEFAULT]
SHADOW - Post Inactivity Timeout... [0]
SHADOW - Enable Post Logging..... [N]
SHADOW - Log all Post Transactions. [N]
SHADOW - Save Process Info on Post. [N]
SHADOW - Abort Post on Errors..... [N]
SHADOW - Buffered Exporting..... [Y]
SHADOW - Transfer mode only..... [N]
SHADOW - Transfer Single Queues... [N]
SHADOW - Transfer ACK Frequency... [0]
SHADOW - Post Directory Filename... [DEFAULT]
C> LIST NODES

Node  Sys name  Line Type  Startup  Spool  Shadow
  1    SYS-A      N         A        B      I

Currently 1 Nodes are defined

C>
```

## DELETE

The **DELETE** command removes a node from the NetBase configuration file. When a node is deleted from the configuration, files located on that node are no longer accessible by local users of NetBase. NetBase should be stopped before deleting a node. This command may be abbreviated to **D**.



**Note.** Deleting a node from the configuration does not remove the references to this node from the directory. Those references become invalid until changed, or until the node is added back.

### Syntax

```
C> DELETE node#
```

### Parameters

Parameter	Description
node#	The number of the node to be deleted from the configuration file.

### Example

To delete node 2 from the configuration:

```
C> LIST NODES
```

Node	Sys name	Line Type	Startup	Spool	Shadow
1*	SYS-A	N	A	N	N
2	SYS-B	N	A	E	N
3	SYS-C	N	A	E	N

```
C> DELETE 2
```

```
C> LIST NODES
```

Node	Sys name	Line Type	Startup	Spool	Shadow
1*	SYS-A	N	A	N	N
3	SYS-C	N	A	E	N

## LIST

The **LIST** command displays the configuration parameters for NetBase. It also lists the parameters for a specific node or for all nodes. Subsets of the global parameters may be listed separately by issuing the **LIST** command with the desired subset as a parameter. This command may be abbreviated to **L**.

### Syntax

```
C> LIST    {ALL      }
           {CONFIG   }
           {DIR      }
           {GLOBAL   }
           {NFA     }
           {NODES   }
           {QUICK   }
           {SHADOW  }
           {SPOOL   }
           {SQL     }
           {STATS   }
           {node#   }
```

### Parameters

Parameter	Description
<b>ALL</b>	Displays the parameters for NetBase and all configured nodes.
<b>CONFIG</b>	Displays all NetBase parameters.
<b>DIR</b>	Displays NetBase directory parameters.
<b>GLOBAL</b>	Displays global parameters.
<b>NFA</b>	Displays the Network File Access parameters.
<b>NODES</b>	Displays a list of the nodes configured in NetBase.
<b>QUICK</b>	Displays the "quick setup" subset of all NetBase parameters.
<b>SHADOW</b>	Displays NetBase Shadowing parameters.
<b>SPOOL</b>	Displays NetBase Spooling parameters.
<b>SQL</b>	Displays NetBase SQL parameters.
<b>STATS</b>	Displays NetBase Statistics parameters.
<i>node#</i>	Displays the parameters for a specific node.

## Example

To display the operational parameters:

```
C> L CONFIG
```

```
GLOBAL - Local Node Number..... 1
GLOBAL - Error Enhancements..... IR
GLOBAL - Maximum Local Users..... 383
NFA - Maximum Remote Users..... 64
NFA - Release Server on Last Close. N
NFA - Initial Server Allocation.... 0
NFA - Server Allocation Timeout.... 300
NFA - Server Initial Priority..... CS
NFA - Server LIB= load flag..... P
NFA - Lock Priority Boosting..... Y
NFA - Incoming users on startup... Y
NFA - Outgoing users on startup... Y
NFA - Client Access on startup.... N
SPOOL - Remote Transfer Timeout.... 900
SPOOL - Transfer Server Allocation. 0
SPOOL - Maximum Block (bytes)..... 8192
SPOOL - Maximum Files per Node.... 32
SPOOL - Use S' and J' on incoming.. Y
SPOOL - Change user on incoming... Y
SPOOL - Change jsname to node name. N
SPOOL - Using SpoolMate..... N
SPOOL - Spooling Export Priority... DEFAULT
SPOOL - Spooling Import Priority... DEFAULT
SPOOL - Enable HP-UX Receiving.... Y
SPOOL - Enable BOOTP daemon..... Y
SPOOL - Enable LPD daemon..... N
SPOOL - Spooling on startup..... Y
STATS - Max File Allocation..... 64
STATS - Init File Allocation..... 24
STATS - Expansion Increment..... 12
STATS - Log File Limit.....2048
STATS - Read-Read Filter (ms)..... 0
STATS - Read-Write Filter (ms).... 0
STATS - Low User Response Filter... 0
STATS - High User Response Filter.. 0
DIR - Default Directory Filename... NBD.DATA.NETBASE
DIR - Allow File Equates..... Y
DIR - Allow Wildcards..... Y
DIR - Map Local Files with Dir.... N
SHADOW - Close Log file on suspend. N
SHADOW - File Locking Required.... N
SHADOW - Flush Shadow File Posts... N
SHADOW - Errors Before Disable.... 1
SHADOW - Check KSAM record counts.. Y
SHADOW - Disable file autoresync .. N
SHADOW - Log Post Errors..... Y
```

---

```

SHADOW - Enable Quiet Period Flush. Y
SHADOW - Export User Exits..... N
SHADOW - Export Bridge Exits..... N
SHADOW - Export Process Info..... Y
SHADOW - Process Info to Exp Exit.. N
SHADOW - Export Buffer Size (b).... 1457
SHADOW - Export Buffer Max BF..... 5
SHADOW - Export Buffer Timeout..... 10
SHADOW - Export Ack Timeout..... 180
SHADOW - Export Initial Priority... DS
SHADOW - Export Upshift Priority... CS
SHADOW - Export Upshift Trigger... 1000
SHADOW - Export Downshift Trigger.. 100
SHADOW - Shift Check Interval..... 100
SHADOW - Disable Shift Messages.... N
SHADOW - Enable Logging..... Y
SHADOW - Log File Size..... 64
SHADOW - Export Log File Timeout... 100
SHADOW - Queue Full Warn Timeout... 180
SHADOW - Export Inactivity Timeout. 5
SHADOW - Import Inactivity Timeout. 5
SHADOW - Post Flush Timeout..... 5
SHADOW - Export on Startup..... Y
SHADOW - Import on Startup..... Y
SHADOW - Post on Startup..... N
SHADOW - Use IMAGE Log Files..... N
SQL - Enable SQL Shadowing..... Y
AUTORPM - Use local logon..... Y

```

Note that the preceding list will vary based on which subsystems are installed. It is also subject to change with future versions of NetBase.

To display the "quick setup" parameters:

```
C> L QUICK
```

```

GLOBAL - Local Node Number..... 1
GLOBAL - Maximum Local Users..... 16
NFA - Maximum Remote Users..... 8
NFA - Initial Server Allocation.... 4
NFA - Server Initial Priority..... DS
SPOOL - Use S' and J' on incoming.. Y
DIR - Default Directory Filename... NBD.DATA.NETBASE
DIR - Allow File Equates..... N
DIR - Allow Wildcards..... Y
SHADOW - Flush Shadow File Posts... Y

```

Note that the set of "quick setup" parameters is also subject to change with future versions of NetBase.

## Chapter 3

---

To display a list of all nodes configured:

```
C> L NODES
```

Node	Sys name	Line Type	Startup	Spool	Shadow
1	SYS-A	N	A	B	I
2*	SYS-B	N	A	N	N

Note that an asterisk (\*) appears next to the local node number on this display.

To display the configuration for a specific node:

```
C> L 1
```

```
Node Number..... 1
System Name..... SYS-A
Startup Access (S/I/O/A/W)..... A
Node Name..... SILVER
Node Connection Timeout..... 180
SPOOL - Direction (N/E/I/B)..... I
SPOOL - Allow incoming job streams. N
SPOOL - Enable Export Flow Control. Y
SHADOW - Direction (N/E/I/B)..... N
SHADOW - Export queue size..... 4096
SHADOW - Import queue size..... 4096
SHADOW - Post queue size..... 4096
SHADOW - Initial Post Priority.... DEFAULT
SHADOW - Post Maximum Files Open... 0
SHADOW - Enable Post Logging..... N
SHADOW - Log all Post Transactions. N
SHADOW - Save Process Info on Post. N
SHADOW - Abort Post on Errors..... N
SHADOW - Buffered Exporting..... Y
SHADOW - Transfer mode only..... N
SHADOW - Transfer Single Queues.... N
SHADOW - Transfer ACK Frequency.... 0
SHADOW - Post Directory Filename... [DEFAULT]
```

## MODIFY

The **MODIFY** command allows you to modify the configuration parameters and the node specific parameters. You can use this command to modify all the configuration parameters, or just the parameters for a specific subsystem. It can also be used to modify the configuration for a specific node. This command may be abbreviated to **M**.

### Syntax

```
C> MODIFY {ALL }
          {CONFIG }
          {DIR }
          {GLOBAL }
          {NFA }
          {NODES }
          {QUICK }
          {SHADOW }
          {SPOOL }
          {SQL }
          {STATS }
          {node# }
```

### Parameters

Parameter	Description
<b>ALL</b>	Modifies the parameters for NetBase and all configured nodes.
<b>CONFIG</b>	Modifies all NetBase parameters.
<b>DIR</b>	Modifies NetBase directory parameters.
<b>GLOBAL</b>	Modifies global parameters.
<b>NFA</b>	Modifies the Network File Access parameters.
<b>NODES</b>	Modifies a list of the nodes configured in NetBase.
<b>QUICK</b>	Modifies the "quick setup" subset of all NetBase parameters.
<b>SHADOW</b>	Modifies NetBase Shadowing parameters.
<b>SPOOL</b>	Modifies NetBase Spooling parameters.
<b>SQL</b>	Modifies NetBase SQL parameters.
<b>STATS</b>	Modifies NetBase Statistics parameters.
<i>node#</i>	Modifies the parameters for a specific node.

The **MODIFY** command recognizes the input of special characters used as commands when modifying. These commands and their meaning are:

Character	Description
//	Terminates command and abort changes
/.	Back up to previous field
/	Terminates command and save changes
?	Displays help information on current field

### Modifying Passwords

You can also change the default user, group, account passwords in **CONFIG**. To do this, enter the following at the **C>** prompt:

```
modify password
```

You will be prompted as follows:

```
PASSWORD - User password.....  
PASSWORD - Group password.....  
PASSWORD - Account password.....
```

### Examples

To change the **GLOBAL** section of the configuration:

```
C> M GLOBAL
```

```
GLOBAL - Local Node Number..... [3]  
GLOBAL - Error Enhancements..... [IR]  
GLOBAL - Maximum Local Users..... [8]
```

To modify node specific parameters:

C> M 1

```

System Name..... [SYS-A]
Startup Access (S/I/O/A/W)..... [A]
Node Name..... [SYS-A]
Node Connection Timeout..... [180]
SPOOL - Direction (N/E/I/B)..... [B]      N
SPOOL - Allow incoming job streams. [N]
SPOOL - Enable Export Flow Control. [N]
SHADOW - Direction (N/E/I/B)..... [N]
SHADOW - Export queue size..... [4096]
SHADOW - Import queue size..... [4096]
SHADOW - Post queue size..... [4096]
SHADOW - Initial Post Priority.... [DEFAULT]
SHADOW - Post Maximum Files Open... [0]
SHADOW - Enable Post Logging..... [N]
SHADOW - Log all Post Transactions. [N]
SHADOW - Save Process Info on Post. [N]
SHADOW - Abort Post on Errors..... [N]
SHADOW - Buffered Exporting..... [Y]
SHADOW - Transfer mode only..... [N]
SHADOW - Transfer Single Queues... [N]
SHADOW - Transfer ACK Frequency... [0]
SHADOW - Post Directory Filename .. [DEFAULT]

```

C> EXIT

## Help on Parameters

To get help on a specific parameter:

```

SHADOW - Close Log file on suspend. [N]
SHADOW - Check KSAM record counts.. [Y]
SHADOW - Disable file autoresync... [N] ?

```

-----  
This parameter disables the autosynchronization of files that occurs when a file is purged or erased. Normally, out of sync or discarding files will be put back into sync when the master copy is erased or purged.

This parameter takes effect when POST is started.  
-----

## PRINT

The **PRINT** command redirects the output of the **LIST** command to the output file **NBOUTPUT**. This is directed to device class **LP** by default. This command may be abbreviated to **P**.

### Syntax

```
C> PRINT      { ALL      }
               { CONFIG  }
               { DIR     }
               { GLOBAL  }
               { NFA     }
               { NODES   }
               { QUICK   }
               { SHADOW  }
               { SPOOL   }
               { SQL     }
               { STATS   }
               { node#   }
```

### Parameters

Parameter	Description
<b>ALL</b>	Prints the parameters for NetBase and all configured nodes.
<b>CONFIG</b>	Prints all NetBase parameters.
<b>DIR</b>	Prints NetBase directory parameters.
<b>GLOBAL</b>	Prints global parameters.
<b>NFA</b>	Prints the Network File Access parameters.
<b>NODES</b>	Prints a list of the nodes configured in NetBase.
<b>QUICK</b>	Prints the "quick setup" subset of all NetBase parameters.
<b>SHADOW</b>	Prints NetBase Shadowing parameters.
<b>SPOOL</b>	Prints NetBase Spooling parameters.
<b>SQL</b>	Prints NetBase SQL parameters.
<b>STATS</b>	Prints NetBase Statistics parameters.
<i>node#</i>	Prints the parameters for a specific node.

## Example

For example, to print the global parameters:

```
C> P_CONFIG  
Sending configuration listing to NBOUPTUT  
C>
```



# CHAPTER 4

## Using NBDIR

NBDIR is an interactive program that creates and maintains NetBase directories. NetBase directories are used by NetBase to determine the location of files in the network. The directories include information about file location, file access, device redirection, program access and file shadowing.

### In This Chapter

- **NBDIR Command Summary**
- **NetBase Directory Maintenance**
- **Special Command Features**
- **NBDIR Commands**

## NBDIR Command Summary

The following table lists the **NBDIR** commands.

Command	Function
<b>D&gt; ACCOUNT</b>	Changes the default account to apply to subsequent directory entries.
<b>D&gt; BUILD</b>	Builds and initializes NetBase directories.
<b>D&gt; DEVICE</b>	Enters device definitions into NetBase's directory for remote spooling.
<b>D&gt; DIR</b>	Changes the directory file to be edited.
<b>D&gt; DISPLAY</b>	Changes node name formats for <b>LIST</b> and <b>UNLOAD</b> commands, either # or name.
<b>D&gt; DSET</b>	Defines datasets for partial database shadowing.
<b>D&gt; EXIT</b>	Exits from the <b>NBDIR</b> program.
<b>D&gt; FILE</b>	Enters files into the NetBase directory for NFA or Shadowing.
<b>D&gt; GROUP</b>	Changes the default group to apply to subsequent directory entries.
<b>D&gt; HELP</b>	Displays <b>NBDIR</b> commands and their syntax.
<b>D&gt; LIMIT</b>	Changes the limit of the current directory.
<b>D&gt; LISTD</b>	Displays the device definitions stored in NetBase's directory.
<b>D&gt; LISTF</b>	Displays file entries.
<b>D&gt; LISTL</b>	Displays logon entries.
<b>D&gt; LISTP</b>	Displays program entries.
<b>D&gt; LISTS</b>	Displays shadow entries.
<b>D&gt; LISTSQLI</b>	Displays SQL import file entries.
<b>D&gt; LISTSQLO</b>	Displays SQL export file entries.
<b>D&gt; LISTX</b>	Displays user defined XL lists.
<b>D&gt; LOGON</b>	Enters logon strings used by AutoRPM.
<b>D&gt; PROGRAM</b>	Enters program records into the directory.

*Continued on the next page*

---

Command	Function
<b>D&gt; REBUILD</b>	Rebuilds the directory KSAM display file.
<b>D&gt; RESET</b>	Removes an entry from the NetBase directory.
<b>D&gt; SHADOW</b>	Enters shadow records into the directory.
<b>D&gt; SQLIN</b>	Enters SQL shadow import records into the directory.
<b>D&gt; SQLOUT</b>	Enters SQL shadow export records into the directory.
<b>D&gt; TABLE</b>	Enters the tables which are to be included in or excluded from SQL shadowing.
<b>D&gt; UNLOAD</b>	Creates an ASCII file containing the commands necessary to recreate the current directory. This file may be edited and subsequently reloaded into <b>NBDIR</b> .
<b>D&gt; XL</b>	Enters user defined XL lists into the directory.

## NBDIR Command Summary

It is a good idea to run **NBDIR** directly rather than through **NBCTRL** since only one person can run **NBCTRL** at a time. This leaves **NBCTRL** free to be run by other users. Any user may run **NBDIR**. However, some of the commands require special capabilities. See the list of capabilities in the following “Special Command Features” section.

Any **NBDIR** command may be passed via the **INFO** parameter of MPE's **:RUN** command. If the **INFO** string contains a command, **NBDIR** will start without printing the program banner, execute the command and terminate.

### To run NBDIR:

1. To run **NBDIR** from MPE, use the following command:

```
: RUN NBDIR.PUB.NETBASE
```

You can also enter **NBDIR** from **NBCTRL** using the **DIR** command as follows:

```
: RUN NBCTRL.PUB.NETBASE  
NetBase Control Program [x.x] Update n (C) QUEST Software 1987  
  
*> DIR  
D>
```

## Wildcards within NBDIR

**NBDIR** supports full use of wildcards in its **LIST** commands. However, it supports wildcards in only a limited manner for **FILE**, **SHADOW** and **PROGRAM** entries. For directory entries, you may use an unqualified **@** for either the file portion or for both the file and group specifications.

```
FILE @.group.account, HP995
FILE @.@.account, FOX
```

**NBDIR** does not support partial wildcarding for directory entries.

```
FILE M@.pub.prod is invalid.
```

Full wildcarding is not supported for directory entries because the NetBase shell performs a maximum of three directory lookups when a file is opened. Full wildcarding is supported on **LIST** commands because the **LIST** command is performed against the KSAM file associated with the directory. The shell will first look for the fully qualified file entry **filename.group.account**. If that is not found, it will search for the entry **@.group.account** for the file being opened. If that entry is not found, the shell will search once again, this time looking for **@.@.account**. By limiting the variations for the file entries in **NBDIR**, the shell can search the directory quickly, reducing overhead and improving performance.

## Editing a NetBase Directory

You can edit a directory two ways — you can edit online, or you can edit an ASCII file.

### To edit the directory online:

1. Run **NBDIR** by entering:

```
:HELLO MGR.NETBASE
:NBDIR.PUB.NETBASE
```

2. List the entry you want to modify.

```
D> LISTF MYDATAB.@.@

Node: FOX          Directory: NBD.DATA.NETBASE          O D F S P
                                                         P S L Y R L
-----LOCAL REFERENCE-----  -----REMOTE EQUIVALENT-----  E E A N O O
File      Group   Account   File      Group   Account   Node      N T G C T G
MYDATAB  DATA   PROD     ->        --Same--      <Local>  - - - - Y
                                     (Shadow)      BUGS
```

3. Remove the entry you want to modify.

```
D> RESET MYDATAB.DATA.PROD
```

4. Re-enter the entry.

```
D> FILE MYDATAB.DATA.PROD,FOX,BUGS;SYNC
```

5. Verify your modification.

```
D> LISTF MYDATAB.@.@

Node: FOX          Directory: NBD.DATA.NETBASE          O D F S P
                                                         P S L Y R L
-----LOCAL REFERENCE-----  -----REMOTE EQUIVALENT-----  E E A N O O
File      Group   Account   File      Group   Account   Node      N T G C T G
MYDATAB  DATA   PROD     ->        --Same--      <Local>  - - - Y - Y
                                     (Shadow)      BUGS
```

6. Exit by typing:

```
D> EXIT
```

### To edit the directory using an ASCII file:

1. Create an ASCII file for editing. As an example, **D> UNLOAD TEMPDIR.PUB.NETBASE**

```
D> UNLOAD filename
```

2. Edit the ASCII file.

```
:QUAD.LIB.NETBASE
/T TEMPDIR
/L ALL
  DIR NBD.DATA.NETBASE
  FILE MYDATAB.DATA.PROD,HP996,HP947;NOLOG
  PROGRAM @.PUB.BRADMARK;NONB
  PROGRAM NBPOST.NB.NETBASE;NONB
  PROGRAM @.NFSPROG.NETBASE;NONB
  PROGRAM NBADAGER.PUB.NETBASE;NONB
  PROGRAM NBGENRL.PUB.NETBASE;NONB
  PROGRAM @.PUB.REGO;NONB
  PROGRAM VTSERVER.NET.SYS;NONB
  PROGRAM CI.PUB.SYS;NONB
  PROGRAM DBRESTOR.PUB.SYS;NONB
  PROGRAM DBSCHEMA.PUB.SYS;NONB
  PROGRAM DBSTORE.PUB.SYS;NONB
  PROGRAM DBUTIL.PUB.SYS;NONB
  PROGRAM @.@.VESOFT;NONB
  SHADOW MYDATAB.DATA.PROD;PASS=CHANGE
  EXIT
.....
.....
.....
/K
  Unnumbered keep to text file TEMPDIR.PUB.NETBASE
  Purge old text file? Y
  lines kept
/E
```

3. Load the directory from the ASCII file you edited.

```
:RUN NBDIR.PUB.NETBASE < TEMPDIR.PUB.NETBASE
```

## Special Command Features

The subsystem command processor has the following special features that provide MPE functionality within NetBase.

- Any MPE command that is executable in break may be issued from the subsystem. To enter an MPE command, type the command prefixed by a colon (:).
- The **REDO** command allows the last command entered to be modified and then executed. To modify and execute the last typed command, simply type **REDO** at the **D>** prompt, modify the line using standard **EDITOR** commands, and press return to execute the command.
- **NBDIR** executes a command file called **DIRINIT.DATA.NETBASE** when a directory is created. This allows you to preload the directory with whatever you need, usually **PROGRAM** records. Additionally, this enables you to add your own commands.
- The **NBDIR** subsystem also supports a help facility. Typing **HELP** displays a list of valid commands.

## NBDIR Commands

The following sections describe the **NBDIR** commands in detail.

### ACCOUNT

The **ACCOUNT** command changes the current default account name. When the directory subsystem is entered, the default account is the logon account. Subsequent **ACCOUNT** commands change the default account name.

The default account name is used in most **NBDIR** commands. By changing the default account, the user is not required to type the account name for each command.

#### Syntax

```
D> ACCOUNT=acctname
```

#### Parameters

Parameter	Description
<i>acctname</i>	The name of the account to be used as the default.

#### Example

Using **LISTF** before using the **ACCOUNT** command:

```
D> LISTF @.DATA
-----LOCAL REFERENCE-----          O D F S P
File      Group   Account          File      Group   Account  Node P E L Y R L
-----REMOTE EQUIVALENT-----          E S A N O O
NO FILES FOUND IN FILESET
```

## Chapter 4

---

The default account is changed and file entries are made in the directory:

```
D> ACCOUNT=PROD
D> FILE AP.DATA=AP.DATA.ACTNG,ACCT
D> FILE MFGENG.DATA,MFG,SHADOW
```

The files are listed again:

```
D> LISTF @.DATA
```

-----LOCAL REFERENCE-----			-----REMOTE EQUIVALENT-----			O D F S P						
File	Group	Account	File	Group	Account	Node	N	T	G	C	T	G
AP	DATA	PROD	-> AP	DATA	ACCTNG	ACCT	-	-	-	-	-	-
MFGENG	DATA	PROD	->	--Same--		MFG	-	-	-	-	-	Y
				(Shadow)		SHADOW						

## BUILD

The **BUILD** command creates and initializes NetBase directories. The default directory size is 1024 records. One record in the directory will be used for each file entry, dataset entry, program entry, user logon, device entry, SQL entry or shadow entry defined.

The directory is a hashed file, not unlike an IMAGE master set. As such, the same capacity considerations that are applied to IMAGE master sets should be considered for the directory. Capacity should generally be maintained at or less than 75% full in order to reduce the incidence of secondaries. Since the shell procedures will access the directory each time a file or database is opened, fewer secondaries will mean improved access performance.

Note that new directories should be created in the **DATA** group of the **NETBASE** account. If the directory is built elsewhere, make sure that the access security for that group and account is correct. This command requires **NODE MANAGER (NM)** and **SYSTEM MANAGER (SM)** capability. Since this command creates files, the user must be logged on in the same account in which the new directory is to reside.

### Syntax

```
D> BUILD filename [,limit]
```

### Parameters

Parameter	Description
<i>filename</i>	Name of the new NetBase directory file.
<i>limit</i>	Maximum number of files allowed in the directory.

### Example

For example, to build a directory called **NBTEST** with a limit of 500 files:

```
D> BUILD NBTEST.DATA,500  
D>
```

## DEVICE

The **DEVICE** command adds device records to NetBase's directory. The device record is used to redirect spooled output from the local node to a remote node. This is done by redirecting the output for a local spooled device or device class to a remote spooled device or device class.

The device record can also be used to specify an alternate spooled device. The alternate device will be used if the primary device is unavailable because communication to the node it is on is down.

The **COPY** option can be used to create a local copy of the spool file while also sending the output to a remote device. In this case, if communication to the remote node goes down while a process that sends output to that node is executing, the process will continue to execute. If the **TEMP** option is specified, the local copy will be purged when the local process completes and its output has been successfully transmitted to the remote device.

Because of the additional overhead incurred by creating a local copy of the spool file, it is recommended that the **COPY** option be used only where line continuity is questionable. For instance, with systems that use certain wide area networks. Local area networks are very dependable making the **COPY** option too costly in terms of performance to use.

Device records can be listed using the **LISTD** command. They are removed by using **RESET**. This command requires **NODE MANAGER (NM)** and **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> DEVICE {class}[={class},node] [;ALT={class} [,node]] [;COPY [=device,outpri,copies]]
        {ldev}[ {ldev } ] [ {ldev } ] [;TEMP]
```

### Parameters

Parameter	Description
<b>class (local)</b> <b>ldev</b>	The device class or logical device number of the spooled device as referenced by the program.
<b>class (remote)</b> <b>ldev</b>	If specified, this parameter is the actual device class or logical device number that the spool file will be mapped to on the remote system. If omitted NetBase uses the same class or ldev as the local reference.
<b>node (remote)</b>	This parameter is the node name or number of the computer to which the actual spool device is connected.

*Continued*

Parameter	Description
<b><i>class (alternate) ldev</i></b>	If specified, this parameter indicates an alternate spool device. The alternate spool device is the device that the spool file will be redirected to in the event that the primary spool node is unavailable.
<b><i>Node (alternate)</i></b>	If specified, this parameter is the node name or number of the computer to which the alternate spool device is connected. This parameter is valid only if an alternate spool device is defined. If omitted, NetBase will use the local node.
<b>COPY</b>	If specified, this option causes NetBase to create a local copy of the spool file in addition to transporting the file to a remote spooler.
<b><i>Device</i></b>	If specified, this is the device class or ldev number to which the local copy is spooled. The default is the device class or ldev number as referenced by the program.
<b><i>outpri</i></b>	If specified, this parameter is the output priority of the spool file copy.
<b><i>Copies</i></b>	If specified, this parameter is the number of copies of the spool file that will be spooled on the local node.
<b>TEMP</b>	This option can be specified only if the COPY option is used. The TEMP option causes NetBase to purge the local copy if the entire spool file has been written to the remote spooler successfully.

### Example

To redirect all output from ldev 6 and class LP to the page printer on node 2:

```
D> DEVICE 6=PP,2
E> DEVICE LP=PP,2
```

To then specify an alternate device if communication to node 2 is unavailable.

```
D> DEVICE 6=PP,2;ALT=LP,1
This Device is already defined. Replace? (Y/N) Y
```

To redirect output for WANLP, keeping a local copy with an outpri of 1 until the spool file is complete.

```
D> DEVICE WANLP=LP,2;COPY=LP,1;TEMP
```

## Chapter 4

---

The new device records can now be listed:

```
D> LISTD
NBD.DATA.NETBASE
T U
E S
---LOCAL---  -----REMOTE-----  -----ALTERNATE-----  -----COPY-----  M E
ldev/class  ldev/class  node          ldev/class  node          ldev/class      P R
6           PP          SYS-B        LP          SYS-A
LP          PP          SYS-B        LP          SYS-A
WANLP      LP          SYS-B        --None--    LP,1,1        P -
```

To remove entry for **WANLP**:

```
D> RESET WANLP,D
```

For more information on the **DEVICE** command, turn to Chapter 7 “NetBase Spooling”.

## DIR

The **DIR** command changes which directory **NBDIR** will use. When **NBDIR** is first executed, it opens the directory that is specified in the NetBase configuration. This command closes the current directory and opens the specified directory.

### Syntax

```
D> DIR filename
```

### Parameters

Parameter	Description
<i>filename</i>	The file name of a valid NetBase directory.

### Example

For example, to change the directory to **TESTDIR**, enter the following syntax. It will look for the directory in the logon group and account.

```
D> DIR TESTDIR.DATA
```

## DISPLAY

The **DISPLAY** command allows you to display the current node name or number in the NetBase directory. This command changes the node name formats for **LIST** and **UNLOAD** commands.

### Syntax

```
D> DISPLAY {NAMES | NUMBERS}
```

### Parameters

Parameter	Description
<i>names</i>	The current node name.
<i>Numbers</i>	The current node number.

### Example

The following example demonstrates how to show the current node number in the NetBase directory.

```
NetBase Directory Program [x.x] Update 0 © QUEST Software 1996

D> listf @.@@
Node: BUGS      Directory: NBD.DATA.NETBASE                O D F S P
                                                         P S L Y R L
-----LOCAL REFERENCE-----  -----REMOTE EQUIVALENT-----  E E A N O O
File      Group   Account   File      Group   Account   Node     N T G C T G
KSGENXRF DATA  APACCT  ->          --Same--   NORWAY    - - - - -
                (Shadow)   <Local>

D> display numbers
E> listf @.@@
Node: BUGS      Directory: NBD.DATA.NETBASE                O D F S P
                                                         P S L Y R L
-----LOCAL REFERENCE-----  -----REMOTE EQUIVALENT-----  E E A N O O
File      Group   Account   File      Group   Account   Node     N T G C T G
KSGENXRF DATA  APACCT  ->          --Same--   10        - - - - -
                (Shadow)   7

D> exit
```

## DSET

The **DSET** command is used in data set shadowing. It works in conjunction with the **FILE** command to define the datasets to be shadowed. The **FILE** command is used first to define a database. The **INCLUDE** and **EXCLUDE** options on the **FILE** command determine whether the dataset defined by the **DSET** command is included for shadowing or if the dataset is to be excluded from the shadowing.

The **RESET** option is used to delete a dataset entry from the directory.

### Syntax

```
D> DSET basename,dataset[,RESET]
```

### Parameters

Parameter	Description
<b>Baseline</b>	A fully-qualified IMAGE file name that already exists in the directory.
<b>Dataset</b>	The name of a dataset that is part of the database defined by filename.
<b>RESET</b>	If specified, the specified dataset entry will be deleted from the directory.

### Example

In the following example we want to shadow only the **HEADERS** and the **DEVICES** datasets in the **BANNER** database. First, enter the **FILE** command for the database:

```
D> FILE BANNER.DATA.NETBASE,1,3;INCLUDE
```

The **INCLUDE** option means that the datasets defined with the **DSET** option will be the only datasets in the database shadowed. Next, define the datasets to be shadowed:

```
D> DSET BANNER.DATA.NETBASE,HEADERS  
E> DSET BANNER.DATA.NETBASE,DEVICES
```

## Chapter 4

---

The datasets that are defined for a database can be listed as follows:

```
D> LISTF BANNER.DATA.NETBASE,DSET

                                     D F S P
                                     P S L Y R L
-----LOCAL REFERENCE-----      -----REMOTE EQUIVALENT-----
File      Group   Account           File      Group   Account   Node
BANNER    DATA   NETBASE  ->                --Same--   NODE-A   - I - - - Y
                                     (Shadow)   NODE-B
      Data Sets:  DEVICES           HEADERS

D>
```

For more information on partial database shadowing, turn to Chapter 5 “NetBase Shadowing”.

## FILE

The **FILE** command identifies a file for NetBase. The parameters on the **FILE** command change how NetBase treats the file. One function of the command is to enable access to files on a remote computer for local processes. Another is to identify files to be shadowed, ensuring data integrity.

The formal file name, or the local reference, is the name of the file as it is referenced by local user processes. If the file is mapped to another computer, the remote name, or remote equivalent, is the name of the file that is actually opened by NetBase. This may be, and usually is, the same as the local reference. The remote name need only be specified when it differs from the local reference.

If the file is shadowed, the file that is opened by NetBase may be the master copy or the shadow copy, depending on the access mode and on whether the **LOCAL** option or the **MASTER** option was specified in the file command. The location of the master copy is designated by the first node listed. Shadowed copies are designated by listing the nodes that the file is to be shadowed on after the master node. The node can be specified either by system name or node number.

Limited wildcard use is available. The file name or group name can be replaced with an @. This allows entire groups and/or accounts to be re-mapped with a single file equation. Since NetBase first checks its directory for specific files, the @ specification can be overridden using specific file names.

The file directory entries can be listed using the **LISTF** command. The different options, that may or may not be included in the file command, are listed in the display along with the file. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

The following table lists the **FILE** options and which process to use them with.

General	NFA	Shadowing (master)	Shadowing (shadow)
<b>LOGONLY</b>	<b>ALT=</b>	<b>CLOSE</b>	<b>DUALACC</b>
<b>NOLOG</b>	<b>UPDPASS=</b>	<b>EXCLUDE</b>	<b>LOCAL</b>
<b>MAP</b>	<b>READPASS=</b>	<b>INCLUDE</b>	<b>LOCALACC</b>
	<b>BUFONLY</b>	<b>IMAGELOG</b>	<b>MASTER</b>
	<b>INFO</b>	<b>NOKFILES</b>	<b>READ</b>
	<b>NOPEND</b>	<b>SYNC</b>	<b>LOCK</b>
			<b>SHADOW</b>
			<b>SYNCERR=FAIL</b>
			<b>SYNCERR=REMOTE</b>

## Syntax

```
D> FILE filename[=remote-name],{nodename}[,shadow[,...]][;options]
      {node# }
```

where

```
option =| DUALACC | LOCAL | LOCALACC | MASTER | READ
        ALT=node[,node]
        BUFONLY
        CLOSE
        EXCLUDE | INCLUDE
        IMAGELOG
        INFO
        LOCK
        LOGONLY | NOLOG
        MAP
        NOKFILES
        NOPEND
        READPASS=password
        SHADOW
        SYNC
        SYNCERR=FAIL
        SYNCERR=REMOTE
        UPDPASS=password
```

## Parameters

Parameter	Description
<b>filename</b>	The fully-qualified formal file name.
<b>Remote-name</b>	If specified, this parameter is the actual file name on the remote computer.
<b>Nodename</b>	The system name of the node that the file resides on.
<b>Node#</b>	The system number of the node that the file resides on.
<b>Shadow</b>	If specified, this parameter is the system name or number of a node on which a shadowed copy resides. This parameter may be repeated, separated by commas, to list all the shadow nodes. If an @ is entered, the file will be shadowed on all other nodes in the NetBase configuration.

*Continued*

Parameter	Description
<b>DUALACC</b>	Enables NetBase to switch from shadow (local) to remote on locks, staying remote until unlock is enabled. Pending locks must be disabled on the shadow system.
<b>LOCAL</b>	If a file is defined with the <b>LOCAL</b> option, NetBase always opens the local copy of the file. If the local copy is a shadow copy, access to the file is limited to read only.
<b>LOCALACC</b>	Same as <b>LOCAL</b> option (above), except that NetBase returns the original access options when requested.
<b>MASTER</b>	If a file is defined with the <b>MASTER</b> option, NetBase always opens the master copy of the file even if a shadow copy exists locally.
<b>READ</b>	Specifying this parameter allows read access only to the shadow copy of the file. If access other than read access is requested, access to the file or database is denied.
<b>ALT=node [,node]</b>	Alternate node for remote access. Any alternate node specified may be specified by node number or by node name.
<b>BUFONLY</b>	The parameter prevents <b>NOBUF/NOMR</b> access to the file. When <b>BUFONLY</b> is set to <b>Y</b> and a file is opened without <b>MR</b> , NetBase sets the <b>BUF(A)</b> to zero, therefore insuring buffering.
<b>CLOSE</b>	If specified, the shadow copy of the file is closed when the posting of shadow transactions is complete.
<b>EXCLUDE</b>	This option is only applicable for data set shadowing of <b>IMAGE</b> files. It specifies that all the datasets in the database, except those specified with the <b>DSET</b> command, will be shadowed.
<b>INCLUDE</b>	This option is only applicable for data set shadowing of <b>IMAGE</b> files. It specifies the only those datasets in this database that are defined with the <b>DSET</b> command will be shadowed.
<b>IMAGELOG</b>	This option identifies the file as an <b>IMAGE</b> log file to be shadowed.
<b>INFO</b>	This option is only applicable for <b>IMAGE</b> files. It causes NetBase to look for and open a local copy of the root file to satisfy <b>DBINFO</b> calls. This option greatly reduces the networking overhead for programs like <b>QUERY</b> and 4th GLs that make extensive use of <b>DBINFO</b> .

Parameter	Description
<b>LOCK</b>	This option is only applicable for non-IMAGE files. It causes the shadow posting process to open the shadow copy with <b>LOCK</b> access. This allows user processes to open the file with lock access.
<b>LOGONLY</b>	This option is only applicable for files that are not shadowed. If logging is enabled, specifying this option causes the export process to log all updates to this file.
<b>NOLOG</b>	This option is only applicable for files that are shadowed. Specifying this option turns off the logging of updates to this file. However, the file will still be shadowed.
<b>MAP</b>	Specifying this option causes NetBase to map the file name for a local open to the remote name as specified in the <b>FILE</b> record. This option is applicable for all file types.
<b>NOKFILES</b>	This option is only applicable for <b>FILE</b> records that use wildcards to specify the shadowing of a whole group or account. If this option is added to such a record, work files with names that match the <b>:LISTF</b> pattern <b>?#####</b> will not be shadowed.
<b>NOPEND</b>	Disables pending locks on this file.
<b>PROTECT</b>	Specifying this option causes NetBase to encrypt the database passwords before calling <b>DBOPEN</b> . Use of this parameter forces programs to be run through NetBase in order to access databases.
<b>READPASS=<i>password</i></b>	<p>When specified for a remote database, this option defines a specific password for NetBase to use when opening the database for read access. It also causes NetBase to use the defined password instead of the password specified by the user process. The password is case sensitive.</p> <p>When specified for a database to be shadowed, this option defines a specific password for NetBase to use when retrieving shadow transactions for <b>IMAGE</b> databases. It also causes NetBase to use the defined password instead of the password specified by the user process. This is necessary when the user password does not grant access to all items in the record. The password is case sensitive.</p>
<b>SHADOW</b>	This option is only applicable when both the file and its shadow copy reside on the same node. Specifying this option causes NetBase to treat this file as the shadow copy.

Parameter	Description
<b>SYNC</b>	This option is only applicable for IMAGE files. It must be specified if item level locking ( <b>DBLOCK</b> modes 5 and 6) is used on the file. It guarantees synchronization of all updates to shadowed files.
<b>SYNCERR=FAIL</b>	This option is only applicable to shadow files. It causes NetBase to disable all access to the shadow file if the file loses synchronization with its master copy.
<b>SYNCERR=REMOTE</b>	This option is only applicable to shadow files. It causes NetBase to open the master file when the shadow file loses synchronization with its master copy.
<b>UPDPASS=<i>password</i></b>	Defines a specific password for NetBase to use when opening a remote database for update access. Causes NetBase to use the defined password instead of the password specified by the user process. The password is case sensitive.

### NFA Example

To set up an entry for NFA:

```
D> FILE OPCUST.DATA.DIST,FOX
```

This makes it possible for local processes to access file **OPCUST.DATA.DIST** on node **FOX**. The actual file can have a different name than the local reference:

```
D> FILE AP.DATA.PROD=AP.DATA.ACCTNG,FOX
```

### Shadowing Example

To specify a file that is to be shadowed:

```
D> FILE ARCUST.DATA.CUST,FOX,VIXEN
```

The master copy of **ARCUST.DATA.CUST** is on node **FOX** and it is shadowed to node **VIXEN**.

## GROUP

The **GROUP** command changes the current default group. When the directory subsystem is entered, the default group is the logon group. Subsequent **GROUP** commands change the default group name.

The default group name is used in most **NBDIR** commands. By changing the default group, the user is not required to type the group name for each command.

### Syntax

D> **GROUP=groupname**

### Parameters

Parameter	Description
<i>groupname</i>	The new default group name.

### Example

For example, to change the default group to **DATA**:

D> **GROUP=DATA**

## LIMIT

This command builds a new directory to the new size specified in the command. It then copies the file entries from the old directory to the new directory and purges the old directory. It also recreates the (KSAM) directory index file and loads it from the new directory. If the filesize parameter is omitted, this command displays the current directory statistics only.

To execute this command, *exclusive access* of the directory must be obtained. The user must also be signed onto the same account as the directory.

### Syntax

```
D> LIMIT [filesize]
```

### Parameters

Parameter	Description
<i>filesize</i>	The new directory capacity.

### Example

For example, to check the current capacity:

```
D> LIMIT

Directory:  NBD.DATA.NETBASE
Limit:      1024
Entries:    14
Secondaries: 0
% Full:     1.36
```

To then change the capacity to 2500 files:

```
D> LIMIT 2500
Creating new directory...
Creating KSAM index files...

Directory:  NBD.DATA.NETBASE
Limit:      2500
Entries:    14
% Full:     .68
```

## LISTD

The **LISTD** command is used to display the devices that are defined in the NetBase directory. The display shows the mapping of spool files across the network. For example, the display header appears as follows:

```

---LOCAL---  -----REMOTE-----  -----ALTERNATE-----  -----COPY-----M E
ldev/class  ldev/class node      ldev/class node      ldev/class  P R

```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>LOCAL</b> ldev/class	The <b>ldev</b> number or class name of the device as it is referenced by local user processes.
<b>REMOTE</b> ldev/class node	The <b>ldev</b> number or class name of the actual device and the node on which the actual device resides.
<b>ALTERNATE</b> ldev/class node	The name and location of an alternate device where NetBase will route the spool file if the <b>REMOTE</b> node is not available.
<b>COPY</b> ldev/class	The device information for a local device to which a copy of the spool file will be sent.
<b>TEMP</b>	If set to <b>T</b> , this flag indicates that the local copy of the spool file is purged after transfer of the file to the remote node is complete.

### Syntax

```
D> LISTD [deviceset]
```

### Parameters

Parameter	Description
<i>deviceset</i>	The logical device number or the device class of the record to be listed. If the device set is omitted, all devices are listed. Wildcards may be used.

## Example

To display all spooled device records in the directory:

```
D> LISTD
NBD.DATA.NETBASE
                                     T U
                                     E S
---LOCAL---  -----REMOTE-----  -----ALTERNATE-----  -----COPY----- M E
ldev/class  ldev/class  node      ldev/class  node      ldev/class  P R
105         PP         SYS-B     --None--
LP          --Same--   SYS-C     --None--
LP88       LP         SYS-B     LP         SYS-A     LP,2,1     T U

D>
```

To display all the devices whose device class starts with the string LP:

```
D> LISTD LP@
NBD.DATA.NETBASE
                                     T U
                                     E S
---LOCAL---  -----REMOTE-----  -----ALTERNATE-----  -----COPY----- M E
ldev/class  ldev/class  node      ldev/class  node      ldev/class  P R
LP          --Same--   SYS-C     --None--
LP88       LP         SYS-B     LP         SYS-A     LP,2,1     T U

D>
```

## LISTF

The **LISTF** command is used to display files that are in the NetBase directory. The display includes all the information about the file. The flags depict the options that were specified when the file was defined with the **FILE** command. The header for the display appears as follows:

```

                                O D F S P
                                P S L Y R L
                                E E A N O O
-----LOCAL REFERENCE-----  -----REMOTE EQUIVALENT-----
File   Group  Account  File   Group  Account  Node N T G C T G

```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>LOCAL REFERENCE</b> File Group Account	The name of the file as it is referenced by local processes.
<b>REMOTE EQUIVALENT</b> File Group Account	The actual name of the file.
<b>Node</b>	The node on which the actual file exists.
<b>OPEN</b>	<p>This flag is used by NetBase in conjunction with shadowing. If a shadowed copy exists locally, this flag determines which copy to open.</p> <p><b>DEFAULT</b> <math>\frac{3}{4}</math> If the flag is not set, NetBase will use the default option. If the file is being opened with read only access, and a shadow copy exists locally, the local copy is used. Otherwise the master copy is accessed.</p> <p><b>C</b> — <b>CLOSE</b> indicates that the posting process will close the shadow copy of the file when the master file is closed.</p> <p><b>L</b> — <b>LOCAL</b> access opens the local copy if it exists. If the local copy is a shadow copy, file access is limited to read only.</p> <p><b>M</b> — <b>MASTER</b> access always opens the master copy regardless of whether a shadow copy exists locally. The requested open mode is always granted.</p>

*Continued*

Header Item	Description
<b>DSET</b>	<p><b>R</b> — Only <b>READ</b> access opens are allowed on the shadow copy of the file or database.</p> <p>This flag applies to dataset shadowing only. Valid values are:</p> <p><b>E</b> % NetBase will shadow the whole database except for the datasets defined. The defined datasets for this file are listed using the <b>DSET</b> or <b>ALL</b> options.</p> <p><b>I</b> % Only the datasets defined will be shadowed. The defined datasets for this file are listed using the <b>DSET</b> or <b>ALL</b> options.</p>
<b>FLAG</b>	<p>If this flag is set, its valid values are:</p> <p><b>I</b> % This flag applies to <b>IMAGE</b> files only. It directs NetBase to open a local copy of the database root file to satisfy <b>DBINFO</b> calls. This is useful if the database is accessed extensively using <b>QUERY</b> or 4th GLs.</p> <p><b>L</b> % This flag is only valid for non-<b>IMAGE</b> files. It causes the NetBase shadow posting process to open the shadow copy with <b>LOCK</b> access.</p>
<b>SYNC</b>	<p>If this flag is set, its value is:</p> <p><b>Y</b> — This flag is used if you are shadowing <b>IMAGE</b> databases and <i>not</i> using <b>TPS</b>, or for non-image files that are accessed by programs with a poor locking strategy.</p>
<b>LOG</b>	<p>If this flag is set, its valid values are:</p> <p><b>I</b> — This flag is only valid for <b>IMAGE</b> log files. It causes NetBase to shadow the <b>IMAGE</b> log file. See the discussion of <b>IMAGELOG</b> under the <b>FILE</b> command in this section for more information.</p> <p><b>Y</b> — This flag causes the export process to log all updates to this file. It is automatically set for all files defined for shadowing, and can be defined for non-shadowed files with the <b>LOGONLY</b> option.</p>

### Syntax

```
D> LISTF [fileset [, {ALL|DSET}]]
```

### Parameters

Parameter	Description
<i>fileset</i>	The file set to be listed. If the files are not in the default group and/or account, the group and/or account must be specified. Wildcards are valid.
<b>DSET</b>	If specified, the dataset records associated with the fileset will be included in the listing.
<b>ALL</b>	If specified, the <b>FILE</b> parameters and dataset records associated with the fileset will be included in the listing.

### Example

To display all files in the directory:

D> LISTF @.@@

```

-----LOCAL REFERENCE-----          -----REMOTE EQUIVALENT-----
File      Group    Account    File      Group    Account    Node
O D F S P
P S L Y R L
E E A N O O
N T G C T G

AP        DATA    PROD      -> AP        DATA    ACCTNG     ACCT
AR        DATA    PROD      -> AR        DATA    ACCTNG     ACCT
GL        DATA    PROD      -> GL        DATA    ACCTNG     ACCT
MFGENG    DATA    PROD      ->          --Same--   MFG        M - I - P -
          (Shadow)   ACCT
MFGORD    DATA    PROD      ->          --Same--   MFG        - - - Y P Y
          (Shadow)   ACCT
MFGPUR    DATA    PROD      ->          --Same--   MFG        - - - Y P Y
          (Shadow)   ACCT
MFGWIP    DATA    PROD      ->          --Same--   MFG        L - - - P -
          (Shadow)   ACCT

```

Or to display a subset of the files in the default group and account:

```
D> GROUP=DATA
D> ACCOUNT=PROD
D> LISTF MFG@
```

-----LOCAL REFERENCE-----			-----REMOTE EQUIVALENT-----				O D F S P	
File	Group	Account		File	Group	Account	Node	P S L Y R L
								E E A N O O
								N T G C T G
MFGENG	DATA	PROD	->	--Same--			MFG	M - I - P -
				(Shadow)			ACCT	
MFGORD	DATA	PROD	->	--Same--			MFG	- - - Y P Y
				(Shadow)			ACCT	
MFGPUR	DATA	PROD	->	--Same--			MFG	- - - Y P Y
				(Shadow)			ACCT	
MFGWIP	DATA	PROD	->	--Same--			MFG	L - - - P -
				(Shadow)			ACCT	

To display the same subset, but with any datasets that are defined:

```
D> LISTF MFG@,DSET
```

-----LOCAL REFERENCE-----			-----REMOTE EQUIVALENT-----				O D F S P	
File	Group	Account		File	Group	Account	Node	P S L Y R L
								E E A N O O
								N T G C T G
MFGENG	DATA	PROD	->	--Same--			MFG	M - I - P -
				(Shadow)			ACCT	
MFGORD	DATA	PROD	->	--Same--			MFG	- E - Y P Y
				(Shadow)			ACCT	
Data Sets: OLDCUSTS								
MFGPUR	DATA	PROD	->	--Same--			MFG	- - - Y P Y
				(Shadow)			ACCT	
MFGWIP	DATA	PROD	->	--Same--			MFG	L - - - P -
				(Shadow)			ACCT	

## LISTL

The **LISTL** command displays remote logon strings used with AutoRPM that have been entered into the NetBase directory. The header for the display appears as follows:

```
-- LOCAL LOGON -- -- TO -- ----- REMOTE LOGON -----
User      Acct      Node      Logon String
```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>LOCAL LOGON</b> User Account	The user and account names of a user running a program through AutoRPM.
<b>TO</b> Node	The remote node on which a program is running.
<b>REMOTE LOGON</b> Logon String	The logon string used by NetBase to establish a communication line with the remote computer. If passwords were specified when the <b>LOGON</b> record was entered, they are indicated by a / character.

### Syntax

```
D> LISTL [user.acct][:node]
```

### Parameters

Parameter	Description
<i>user</i>	The name of the local user or users whose logon strings will be displayed. Full wildcard use is valid. If specified, the node must also be specified. If no set of users is specified, <b>NBDIR</b> will default to @:@.
<i>Acct</i>	If specified, this is the account of the local user or users whose logon strings will be displayed. Full wildcard use is valid. If the account is not specified, the default account will be used.
<i>node</i>	If specified, this is the remote node that the set of users specified will be logged on to by NetBase. If @ is entered, the set of users logging on to all nodes will be displayed.

## Example

To list all logon strings in the directory:

D> LISTL @.:@

Directory: NBD.DATA.NETBASE

LOCAL LOGON	TO	REMOTE LOGON	
User	Acct	Node	Logon String
@	ACCT	NODE-A	DESKUSER,MGR.NETBASE,PUB
MGR	DIST	NODE-B	RPM,MGR.NETBASE,PUB
TOM	ACCT	NODE-A	TOM,MGR/.NETBASE/,GRP/

D>

To list the logon strings for user **MGR@** in the default account, logging on to all nodes:

D> LISTL MGR@:@

LOCAL LOGON	TO	REMOTE LOGON	
User	Acct	Node	Logon String
MGR	ACCT	NODE-B	MGR,MGR.ACCT,PUB
MGRAR	ACCT	NODE-A	MGRAR,MGR.ACCT,DATA

D>

## LISTP

The **LISTP** command displays records that are in the NetBase directory. Records are used for two different purposes. One function is to disable NetBase options for a program or a set of programs. They are also used for AutoRPM.

The display shows the NetBase options that are to be disabled for this file or file set. It also shows the AutoRPM options that are set for this file set. The header for the display appears as follows:

```

Node: FOX          Directory: NBD.DATA.NETBASE          B P S
                                           L U E H
-----LOCAL PROGRAM-----  ----REMOTE EQUIVALENT----  N O F N U
Program Group  Account  Program Group  Account  Node  B C F D T
    
```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>LOCAL PROGRAM</b> <b>Program Group Account</b>	If <b>Node</b> is <Local>, this is the name of a file that has NetBase options disabled.  If <b>Node</b> is not <Local>, this is the name of a local file that will, through AutoRPM, run a file on a remote node.
<b>REMOTE EQUIVALENT</b> <b>Program Group Account</b>	The name of the file to be run by AutoRPM.
<b>Node</b>	The node running the file.
<b>NB</b>	If set to <b>N</b> , this file set will not use any NetBase features.
<b>LOC</b>	If set to <b>L</b> , all shadow files will be opened by the program. Will be opened locally.
<b>BUFF</b>	If set to <b>B</b> , this set of programs will not use NetBase's network buffering for NFA.
<b>PEND</b>	If set to <b>N</b> , it indicates that pending locks have been disabled for the set of programs. This causes unconditional locks to be performed as individual network transactions, rather than coupling them with the next call to the database.
<b>SHUT</b>	If set to <b>S</b> , the remote line will be closed when the remote file terminates. (This flag applies to programs run by AutoRPM.)

If a file is defined with load parameters for AutoRPM, the load parameters are displayed on the second line. This line displays the load library and the run parameter, the XL list and the **DEBUG** flag.

## Syntax

```
D> LISTP [fileset]
```

## Parameters

Parameter	Description
<i>fileset</i>	Indicates the file or set of files to be listed. Wildcards are allowed. If the group or account is not specified, the default group or account is used.

## Example

To display all the files in the directory:

```
D> LISTP @.@.@
Node: FOX          Directory: NBD.DATA.NETBASE

-----LOCAL PROGRAM-----      ----REMOTE EQUIVALENT----
Program  Group   Account  Program  Group   Account  Node   B P S
                                L U E H
                                N O F N U
                                B C F D T

TESTRPM  NFA     NBACCT  ->      --Same--      BUGS      - - - - -
NBCOP    NB      NETBASE ->      --Same--      local     N - - - -
BASIC    PUB     NETBASE ->      --Same--      local     - - - - -
LIB=S
NBADAGER PUB     NETBASE ->      --Same--      local     N - - - -
NBPASS   PUB     NETBASE ->      --Same--      local     T - - - -
NBRPT01  PUB     NETBASE ->      --Same--      local     N - - - -
NBSPOOLR PUB     NETBASE ->      --Same--      local     - - - - -
LIB=D
SORTSUM  PUB     NETBASE ->      --Same--      local     N - - - -
VISTACLN PUB     NETBASE ->      --Same--      local     N - - - -
@        PUB     REGO    ->      --Same--      local     N - - - -
DBRESTOR PUB     SYS      ->      --Same--      local     N - - - -
DBSCHEMA PUB     SYS      ->      --Same--      local     N - - - -
DBSTORE  PUB     SYS      ->      --Same--      local     N - - - -
DBUTIL   PUB     SYS      ->      --Same--      local     N - - - -
FORMSPEC PUB     SYS      ->      --Same--      local     N - - - -
```

## LISTS

The **LISTS** command displays shadow records that are in the NetBase directory. A shadow record identifies a shadow file that exists on the local node. It is used by the posting process to identify shadow records as they come in from remote nodes. They are also used for information about the current state of shadowing for the file. The header for the shadow file display appears as follows:

```

-----TRANSPORT REFERENCE-----          ---LOCAL COPY REFERENCE---          S Q U D S L
File      Group  Account  Node          File      Group  Account  N E R S R G

```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>TRANSPORT REFERENCE</b> <b>File Group Account Node</b>	The name of the file as it is received by the import process and the node from which it is received.
<b>LOCAL COPY REFERENCE</b> <b>File Group Account</b>	The actual name of the shadow file.
<b>SYN = S</b>	If this flag is set, a synchronization error has occurred on this file and it is time to resynchronize the shadow file. Turn to Chapter 5 “NetBase Shadowing” for more information on how to resynchronize shadow files.
<b>QUE = Q, P, R</b>	<p>This flag indicates that updates to the file are queued by the posting process. This occurs when the shadow file is unavailable or the <b>DEFER</b> command has been issued for the file.</p> <p><b>Q ¼ Queued</b> indicates that a post queue has been created for this file. If a post queue has been created, the <b>ROLL FORWARD</b> command should be executed on this file as soon as the shadow copy is available.</p> <p><b>P ¼ Pending</b> indicates that the posting for this file has been deferred with the <b>DEFER</b> command.</p>

*Continued*

Header Item	Description
<b>USR = U</b>	<b>R %4 Roll forward</b> indicates that a roll forward recovery is in progress.  This flag indicates that <b>Post User Exits</b> are enabled for this file.
<b>DIS = D</b>	This flag indicates that NetBase is discarding updates to this particular file. If it is set, the <b>SYN</b> flag may also be set, indicating why this is occurring. Turn to Chapter 5 “NetBase Shadowing” for more information on error recovery.
<b>SHR = Y</b>	This flag indicates that the shadow copy is being opened in shared access mode for the posting of updates. This happens when <b>;MODE 1</b> is specified on a <b>SHADOW</b> entry.
<b>LOG = P, R</b>	<b>P %4</b> indicates post logging is enabled (for diagnostic use only).  <b>R %4</b> roll forward in progress.

Some additional information may be listed on a second line if there have been synchronization errors. The **Reset** time is the time that the file was last resynchronized. The **Sync Lost** time is the time the file went out of sync. The **Last Update** time is the time stamp of the last update that was applied (the time stamp is relevant to the time that the update was made to the master copy of the file) before the file was last closed by the posting process.

## Syntax

```
D> LISTS [fileset[, {ALL|ERR|PASS}]]
```

## Parameters

Parameter	Description
<i>fileset</i>	The file set to be listed. Wildcards are allowed. If the group or account is not specified, the default group or account will be used.
<b>ALL</b>	If specified, synchronization and update information for the file will be displayed along with the file entries.
<b>INFO</b>	This option can be used in place of <b>ALL</b> . It performs exactly the same function.

*Continued*

Parameter	Description
ERR	Shows only those records that are in one of the following states: <b>Q</b> , <b>S</b> , or <b>D</b> .
PASS	Displays passwords on <b>SHADOW</b> statements.

### Example

To list all the shadow files on this node:

```
D> LISTS @.@@
Directory: NBD.DATA.NETBASE

-----TRANSPORT REFERENCE-----      ---LOCAL COPY REFERENCE---      S Q U D S L
File      Group      Account      Node      File      Group      Account      Y U S I H O
MFGENG    DATA      PROD         MFG      ->        --Same--      - - - - -
MFGORD    DATA      PROD         MFG      ->        --Same--      - - - - -
MFGPUR    DATA      PROD         MFG      ->        --Same--      - - - - -
MFGWIP    DATA      PROD         MFG      ->        --Same--      - - - - -
```

To display the shadow files along with their synchronization and update statistics:

```
D> LISTS @.@@,ALL
Directory: NBD.DATA.NETBASE

-----TRANSPORT REFERENCE-----      ---LOCAL COPY REFERENCE---      S Q U D S L
File      Group      Account      Node      File      Group      Account      Y U S I H O
MFGENG    DATA      PROD         MFG      ->        --Same--      - - - - -
  Resync Time is 09/22/96 11:14:26
  Last Update at 10/02/96 09:01:10
MFGORD    DATA      PROD         MFG      ->        --Same--      - - - - -
  Last Update at 09/30/96 16:40:57
MFGPUR    DATA      PROD         MFG      ->        --Same--      - - - - -
  Last Update at 10/02/96 13:26:27
MFGWIP    DATA      PROD         MFG      ->        --Same--      - - - - -
  Resync Time is 10/02/96 14:13:58
  Last Update at 10/02/96 15:26:10
```

## LISTSQLI

The **LISTSQLI** command displays DBE's that are defined for inbound SQL shadowing in the NetBase directory. The flags depict the options that were specified when the file was defined with the **SQLIN** command. The header for the inbound SQL shadowing file display appears as follows:

```

-----LOCAL DBE NAME-----          -----ACTUAL DBE NAME-----          E
File   Group  Account      File   Group  Account      Node      T  Pri

```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>LOCAL DBE NAME</b> File Group Account	The name of the file as it is controlled by NetBase.
<b>ACTUAL DBE NAME</b> File Group Account	The actual name of the file on this system.
<b>Node</b>	The node from which transactions are received.
<b>EXIT</b>	If this flag is set, the SQL importing process will call a user exit for each transaction received for this file. See the <i>NetBase SQL Shadowing Handbook</i> for more information.
<b>PRI</b>	This flag displays the priority defined for the import process to use for the transactions to be applied. See the <i>NetBase SQL Shadowing Handbook</i> for more information.

### Syntax

```
D> LISTSQLI [fileset]
```

### Parameters

Parameter	Description
<i>fileset</i>	The file set to be listed. Wildcards are allowed. If the group or account is not specified, the default group or account will be used.

### Example

## Chapter 4

---

To list all the SQL shadow files on this node:

```
D> LISTSQLI @.@@
Directory: NBD.DATA.NETBASE
-----LOCAL DBE NAME-----
File      Group   Account
-----ACTUAL DBE NAME-----
File      Group   Account  Node   T  Pri
PAYDBE4   DATA   PROD     ->     --Same--  VIXEN   Y    0
D>
```

## LISTSQLO

The **LISTSQLO** command displays DBE's that are defined for outbound SQL shadowing in the NetBase directory. The flags depict the options that were specified when the file was defined with the **SQLOUT** command. The header for the outbound SQL shadowing file display appears as follows:

```

-----LOCAL DBE NAME-----          -----ACTUAL DBE NAME-----          E
File   Group   Account   File   Group   Account   Node   I # of
T Part Freq

```

The following table provides descriptions of the items in the display header.

Header Item	Description
<b>LOCAL DBE NAME</b> File Group Account	The name of the file as it is referenced by NetBase processes.
<b>ACTUAL DBE NAME</b> File Group Account	The actual name of the DBE.
<b>Node</b>	The node to which transactions are being sent.
<b>EXIT</b>	If this flag is set, the SQL exporting process will call a user exit for each transaction sent for this file. See the <i>NetBase SQL Shadowing Handbook</i> for more information.
<b># OF PART</b>	This flag displays the number of partitions of the DBE being shadowed.
<b>FREQ</b>	This flag indicates that length of time the export process will pause between checks to see if new transactions have been created.

### Syntax

```
D> LISTSQLO [fileset]
```

### Parameters

Parameter	Description
<i>fileset</i>	The file set to be listed. Wildcards are allowed. If the group or account is not specified, the default group or account will be used.

### Example

To list all the shadow files on this node:

```
D> LISTSQLO @.@@
Directory: NBD.DATA.NETBASE
E
X
I # of
T Part Freq
-----LOCAL DBE NAME-----
File      Group   Account   ->
-----ACTUAL DBE NAME-----
File      Group   Account   Node
PAYDBE2   DATA   PROD      ->      --Same--   FOX       Y       1       0
PAYDBE4   DATA   PROD      ->      --Same--   FOX       -       3       0
PAYDBE4   DATA   PROD      ->      --Same--   VIXEN     Y       3       0
Partitions:
D>
```

## LISTX

The **LISTX** command is used to display user-defined XL lists in the directory. The XL list is used by NetBase when running a remote process or creating a local process. The **XL=** option of the **PROGRAM** command attaches an XL list to a program to be run by NetBase.

### Syntax

```
D> LISTX [listname]
```

### Parameters

Parameter	Description
<i>listname</i>	User-defined name of an XL list. Wildcards are allowed.

### Example

```
D> LISTX @
Directory: NBD.DATA.NETBASE

Name      XL List
TESTXL    NMXL1.PUB,NMXL2.PUB

D>
```

For more information, turn to Chapter 8 “AutoRPM”.

## LOGON

The **LOGON** command enters logon records into the directory. The logon record is used by AutoRPM when logging a user onto a remote node. Wildcards are valid only for the **localuser** parameter. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> LOGON localuser[:node]=logonstring
```

### Parameters

Parameter	Description
<b>localuser</b>	The user name and account name of the local user. The @ wildcard is valid only for user.
<b>node</b>	The remote node on which to log the local user using the remote logon. The @ wildcard is <i>not</i> allowed.
<b>logonstring</b>	The remote logon string. The logon string must be in the same format as the <b>MPE HELLO</b> command ( <b>sname,uname/ pass.aname/pass,gname</b> ). Only the session name is optional. If it is omitted, and a preceding comma (,) is supplied ( <b>,uname.aname,gname</b> ) the executing user's local session name will be inserted. The group name is required.

### Example

To set up a logon string for all users logged in to the **NETBASE** account:

```
D> LOGON @.NETBASE:SYSB=NETUSER,MGR.NETBASE,PUB
```

To set up a logon string for **MGR** in the **DIST** account, using the default account and specifying that the users local session name should be used on the remote logon:

```
D> ACCOUNT=DIST  
D> LOGON MGR:SYS-C=,MGR.DIST,PUB
```

---

To display the logon records after they have been entered:

```
D> LISTL @.:@
```

```
Directory: NBD.DATA.NETBASE
```

```
-- LOCAL LOGON -- -- TO -- ----- REMOTE LOGON -----  
User      Acct      Node      Logon String  
  
@         NETBASE   SYSB      NETUSER,MGR.NETBASE,PUB  
MGR       DIST      SYSC      RPM,MGR.DIST,PUB
```

For more information, turn to Chapter 8 “AutoRPM”.

## PROGRAM

This command writes the program records to the open NetBase directory. There are three purposes for adding program records to the directory, they are:

- Disable NetBase functions on one program or on a set of programs.
- Override certain run time options from those specified at process creation.
- Define processes that NetBase will run on a remote node using NS's RPM.

If the command is issued without specifying a remote program name or remote node, it is used to disable NetBase functions. The **NONB** option disables all NetBase functions for the program or set of programs. If it is selected, it is the only option that may be selected. The rest of the options for this mode of the command may be specified in almost any combination. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

If a remote node name or number is specified in the **PROGRAM** command, the command is used to set up a program that NetBase will run on a remote machine using NS's RPM. The options for this mode of the command modify how the program on the remote node is executed.

### Syntax

```
D> PROGRAM progname [ ;MAP ] <Disable NetBase options>
                   [ ;NOBUF ]
                   [ ;NONB ]
                   [ ;NOREL ]
                   [ ;NOSTAT ]
```

```
D> PROGRAM progname [ ;DEBUG ] <Override local runtime options>
                   [ ;LOCAL ]
                   [ ;LIB= ]
                   [ ;NOPEND ]
                   [ ;PARAM= ]
                   [ ;XL= ]
```

```
D> PROGRAM progname [=remoteprog][,nodeid][ ;LIB= ] <AutoRPM>
                   [ ;PARAM= ]
                   [ ;NOEQ ]
                   [ ;NOJCW ]
                   [ ;XL= ]
```

```
D> PROGRAM progname [ ;SHADOW ] <Special options>
                   [ ;TRACE ]
```

## Parameters

Parameter	Description
<i>progname</i>	The programs running under NetBase whose NetBase functions will be disabled.
<b>MAP</b>	Uses a <b>remoteprog</b> name instead of a local name for local creations.
<b>NOBUF</b>	Disables NFA buffering for this program set.
<b>NONB</b>	Disables all NetBase features for this set of programs. If this option is selected, all other options are invalid.
<b>NOREL</b>	If this option is selected, the server for a process in this program set is not released on the last file close. This option overrides the <b>Release Server on Last Close</b> configuration parameter.
<b>NOSTAT</b>	Disables the collection of statistics for this program set.
<b>DEBUG</b>	Runs the remote program with the debugger.
<b>LOCAL</b>	Makes all access by a program to shadowed files local, regardless of whether the program is trying to open a file for update access.
<b>LIB=</b>	The load library that is to be used when the process is created on the local node. If not specified, NetBase uses <b>LIB=P</b> . To override this default and require that the program be run without NetBase overriding the load library specified at process creation, enter <b>LIB=D</b> .
<b>NOPEND</b>	Disables pending locks handling, causing unconditional locks to be transported across the network as individual transactions.
<b>PARAM=</b>	Uses this parameter string as an override to the parm specified at process creation.
<b>XL=</b>	The name of a user defined XL list entered in the directory with the <b>XL</b> command. The XL list is used as a local override for an XL list specified at process creation.
<i>progname</i>	The name of the program on the local node. If the wildcard option is enabled, the @ wildcard is valid for file name and group name.
<i>remoteprog</i>	If specified, this is the name of the program that NetBase will run on the remote node. If not specified, the remote program name is the same as the local program name. If the wildcard option is enabled, the @ wildcard is valid for file name and group name.

Continued

Parameter	Description
<i>nodeid</i>	The remote node where NetBase will run the remote program.
<b>NOEQ</b>	This option disables sending the file equates for the local process to the remote node for the remote process.
<b>NOJCW</b>	If this option is selected, NetBase will not transmit the JCW from the remote process to the local process when the remote process terminates.
<b>SHADOW</b>	Causes NetBase to preallocate all shadowing buffers before first file open. Required for some <b>VPLUS</b> applications.
<b>TRACE</b>	This option enables NetBase to do an application trace of the individual program specified in the directory entry.

### Example

To exclude all the programs in the **REGO** account from using NetBase services:

```
D> PROGRAM @.@.VESOFT;NONB
```

To disable **QUERY** from using NetBase statistics:

```
D> PROGRAM QUERY.PUB.NETBASE;NOSTATS
```

To substitute one program for another on the local node:

```
D> PROGRAM EDITOR.PUB.SYS=QUERY.PUB.SYS,thisnode;MAP
```

To run a set of utility programs on a remote node:

```
D> PROGRAM @.PUB.DESK,UTILS
```

To run a program on a remote node, specifying the load library and a parameter string:

```
D> PROGRAM LHPMAIL.PUB.REMOTE=HPMAIL.PUB.SYS,3;LIB=G;PARAM=1
```

To run a native mode program on the remote node, specifying an XL list and the **DEBUG** parameter, first enter the XL list:

```
D> XL TESTXL=NMXL1.PUB,NMXL2.PUB
D> PROGRAM XLPROG,3;XL=TESTXL
```

To display all the program records in the directory:

```
D> LISTP @.@.@

Directory: NBD.DATA.NETBASE

-----LOCAL PROGRAM-----      ----REMOTE EQUIVALENT--
Program Group  Account      Program Group  Account  Node
QUERY  PUB      NETBASE  ->      --Same--      local
@      PUB      DESK     ->      --Same--      UTILS
LHPMAIL PUB      REMOTE   ->HPMAIL  PUB      SYS      SYSB
LIB=G;PARM=1
XLPROG  TEST     NETBASE  ->      --Same--      SYSB
XL=TESTXL
```

If the XL list record is deleted while still defined to a program, the program would display as follows:

```
D> RESET XLTEST,XL
D> LISTP XLPROG.TEST

Directory: NBD.DATA.NETBASE

-----LOCAL PROGRAM-----      ----REMOTE EQUIVALENT--
Program Group  Account      Program Group  Account  Node
XLPROG  TEST     NETBASE  ->      --Same--      SYSB
XL=TESTXL (undef)  DEBUG
```

## REBUILD

The **REBUILD** command rebuilds the KSAM file associated with the current directory. This command is normally used if the integrity of the KSAM directory is suspect. To execute this command, you must have *exclusive access* of the directory KSAM file. You must also be signed onto the same account as the directory.

### Syntax

```
D> REBUILD
```

### Parameters

This command does not have any parameters.

### Example

To rebuild the KSAM file associated with the current directory:

```
D> REBUILD
```

To rebuild the KSAM file for a directory other than the default directory:

```
D> DIR NEWNBD.DATA.NETBASE  
D> REBUILD
```

## RESET

The **RESET** command removes a record from the NetBase directory. Once an entry is removed, it will no longer be referenced the next time that file is opened. Active processes, with previously defined files opened, will not be affected. **DSET** records are removed with the **DSET** command. **SQLOUT** and **SQLIN** records are removed by the **SQLOUT** and **SQLIN** commands. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> RESET filename[,recordtype]
```

where

```
recordtype = F or FILE
             S or SHADOW
             P or PROGRAM
             D or DEVICE
             L or LOGON
             X or XL
```

### Parameters

Parameter	Description
<i>filename</i>	The file name to be removed from the directory. If the group name or the account name is not specified, the default group or account will be used.
<i>recordtype</i>	Specifies the type of record to be removed from the directory. Record type is expressed as one of the following: <ul style="list-style-type: none"> <li><b>FILE</b> <math>\frac{3}{4}</math> Removes a file record from the directory. This is the default when the command is issued without specifying a record type.</li> <li><b>SHADOW</b> <math>\frac{3}{4}</math> Removes a shadow record from the directory.</li> <li><b>PROGRAM</b> <math>\frac{3}{4}</math> Removes a program record from the directory.</li> <li><b>DEVICE</b> <math>\frac{3}{4}</math> Removes a device record from the directory.</li> <li><b>LOGON</b> <math>\frac{3}{4}</math> Removes a logon record from the directory.</li> <li><b>XL</b> <math>\frac{3}{4}</math> Removes an XL list record from the directory.</li> </ul>

### Example

To remove a file record from the directory:

```
D> RESET MFGENG.DATA.PROD
```

To remove a file record from the directory, using the default group and account:

```
D> GROUP=DATA  
D> ACCOUNT=DIST  
D> RESET OPCUST
```

To remove a program record from the directory:

```
D> RESET QUERY.PUB.NETBASE,PROGRAM
```

To remove a device record from the directory, using the short form of the record type flag:

```
D> RESET 88,D
```

To remove all the **File** statements for files that start with the letter **K**:

```
D> RESET K@.@.@,F
```

## SHADOW

The **SHADOW** command adds shadow records to NetBase's directory. The shadow record defines shadowing for a file that is already defined in the network. **SHADOW** records are entered on the shadow system only.

The posting processes uses the shadow record when opening the file for updates and to mark the status of updates to the shadow file. Turn to Chapter 5 “NetBase Shadowing” for more information on how these records are used. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> SHADOW filename[=copyfilename][;option[;option[;...]]]
      [;DBX      ]
      [;LOG      ]
      [;MODE1    ]
      [;MULTISYS]
      [;OVERRIDE]
      [;USEREXIT]
      [;PASS=_____]
```

### Parameters

Parameter	Description
<i>filename</i>	The name of the shadow file as it is passed to the posting process. The file name must already exist in a file record in the directory or be covered by a wildcard entry in the directory.
<i>copyfilename</i>	If specified, this is the actual name of the shadow file.
<b>DBX</b>	Only required if TPS shadowing is <i>not</i> used and dynamic transactions are used within the application on the master. Turn to the “Shadowing with Dynamic Rollback Recovery” section in Chapter 5 for implementation steps.
<b>LOG</b>	If specified, all updates to the shadow file will be logged to a post log file. Requires the node configuration parameter <b>SHADOW - Enable Post Logging</b> be set to <b>Y</b> .
<b>MODE1</b>	If specified, the shadow copy will be opened in shared access mode for the posting of updates.

*Continued*

Parameter	Description
<b>OVERRIDE</b>	Applicable to shadowed MPE files only. If specified, this parameter causes the post process to ignore update related synchronization errors.
<b>PASS=password</b>	This option applies only to IMAGE databases and is required. It specifies the password for a database to be used by the posting process to open the shadow file for updating. The password must allow update access to all sets and items to be shadowed.  <b>Note:</b> Remember that all IMAGE passwords are case sensitive.
<b>USEREXIT</b>	This parameter specifies that post should call a user posting subroutine when posting updates to this file. Turn to the "Post User Exits" section in Chapter 5 for more information.

### Example

To add a shadow entry for a database file:

```
D> SHADOW AP.DATA.PROD;PASS=DOUBLE
```

To add a shadow record for a file where the actual shadow file name is different:

```
D> SHADOW CUSTFL.DIST.DIST=CUSTFL.DATA.ACCT
```

To list the shadow records from the directory:

```
D> LISTS @.@@
Directory: NBD.DATA.NETBASE

-----TRANSPORT REFERENCE-----      ---LOCAL COPY REFERENCE---      S Q E D S
File      Group      Account      Node      File      Group      Account      Y U R I H
N E R S R

AP        DATA      PROD        SYS-A      ->        --Same--      - - - - -
CUSTFL    DIST       DIST        DIST       CUSTFL    DATA      ACCT        - - - - -
```

To list a shadow file that has gone out of sync:

```
D> LISTS AP.DATA.PROD
Directory: NBD.DATA.NETBASE

-----TRANSPORT REFERENCE-----          S Q E D S
File      Group    Account  Node          File      Group    Account  N E R S R
AP        DATA    PROD     SYS-A    ->          --Same--          S - - D -
                                     Sync Lost at 02/13/96 14:00:00
```

## SQLIN

The **SQLIN** command is used to define which DBE's are importing updates from which nodes. See the *NetBase SQL Shadowing Handbook* for more information on how these records are used. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> SQLIN dbname node [;PRI=transaction-priority]
      [;EXECPRI={BS|CS|DS|ES}]
      [;ORIGDBE=dbname]
      [;USEREXIT]
      [;RESET]
```

To set up multiple import processes for one DBE:

```
D> SQLIN fakedbe1=dbname,node;PART=part1,[part2[,...]][...]
D> SQLIN fakedbe2=dbname,node;PART=...
```

To remove a **SQLIN** command:

```
D> SQLIN dbname, node;RESET
```

### Parameters

Parameter	Description
<b>dbname</b>	The name of the SQL DBE.
<b>node</b>	The node from which SQL transactions are received.
<b>transaction-priority</b>	The PRI option is used to set the priority on the <b>BEGIN WORK</b> for the transaction being applied. This is important in the case of deadlocks. The import process should have the highest priority of all the processes on the system. This is because HP resolves a deadlock by aborting the process with the lowest priority. The valid range is 0 to 255 with the highest priority going to the lowest number. The default is 0.

*Continued*

Parameter	Description
<b>EXECPRI=BS CS DS ES</b>	Specifies the execution priority for the SQL import process.
<b>ORIGDBE</b>	Specifies the DBE name on the master.
<b>USEREXIT</b>	Specifies that a user exit routine should be called with each transaction received for the defined file.
<b><i>fakedbe1, fakedbe2</i></b>	The pseudo name of the database to be used to administer the partition. Use this name when starting and stopping the import process for the partition.
<b><i>PART=part1, part2</i></b>	The partitions to be shadowed via this import process.

### Example

To add a SQL DBE entry for a database file, accepting the default priority:

```
D> SQLIN PAYDBE1.DATA.PROD,4
```

To set up separate import processes per partition:

```
D> SQLIN DBE.PART1.PROD=DBEHERE.DATA.PROD,masternode;PART=1;ORIGDBE=MASTDBE;  
EXECPRI=CS
```

To list the **SQLIN** records from the directory:

```
D> LISTSQLI @.@.@
Node: SHADOW      Directory: NBD.DATA.NETBASE
-----LOCAL DBE NAME-----      -----REMOTE EQUIVALENT-----
File      Group      Account      File      Group      Account      Node      T      Pri
MMDBE     PART1     PRODACT -> MMDBE     DATAB     PRODACT     MASTER   -      0
Exec Pri: CS Partitions: 1
MMDBE     PART4     PRODACT -> MMDBE     DATAB     PRODACT     MASTER   -      0
Exec Pri: DS Partitions: 4
EEDBE     DATAG     PRODTWO ->          --Same--          MASTER   -      0
Orig DBE: CCDBE
```

## SQLOUT

The **SQLOUT** command is used to define which DBEs are to have transactions exported and to which nodes. See the *NetBase SQL Shadowing Handbook* for more information on how these records are used. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> SQLOUT dbname,node
      ;HOME=home-partition[,part2[,...]]
      [ ;EXECPRI={BS|CS|DS|ES} ]
      [ ;FREQ=wakeup-seconds ]
      [ ;{INCLUDE|EXCLUDE} ]
      [ ;RESET ]
      [ ;USEREXIT ]

D> SQLOUT dbname,@;PART=partition1[,partition2[,...]]
      [ ;EXECPRI={BS|CS|DS|ES} ]
      [ ;FREQ=wakeup-seconds ]
      [ ;{INCLUDE|EXCLUDE} ]
```

### Parameters

Parameter	Description
<i>dbname</i>	The name of the SQL DBE.
<i>node</i>	The node to which SQL transactions are transported.
<i>home-partition</i>	The home partition number is only entered on the export record ( <b>SQLOUT</b> ). This is the home partition number of the local copy of the DBE.
<i>PART=partition1, partition2</i>	The list of partitions to export from the DBE to a user exit (indicated by @ as the node).
<i>EXECPRI=BS CS DS ES</i>	Specifies the execution priority for the SQL export process.
<i>wakeup-seconds</i>	Specifies the length of time that the export process will pause between checks to see if new records have been added to the log file. The pause only takes effect after a call to get the latest record from the log file fails. The default is 20 seconds.

*Continued*

Parameter	Description
<b>EXCLUDE</b>	Excludes (from shadowing) the tables specified in the <b>TABLE</b> command for this database.
<b>INCLUDE</b>	Shadows the tables specified in the <b>TABLE</b> command for this database.
<b>RESET</b>	Resets the <b>SQLOUT</b> record for this database.
<b>USEREXIT</b>	Specifies that a user exit routine should be called with each transaction sent for the defined file.
<i>fakedbe1, fakedbe2</i>	The pseudo name of the database to be used to administer the partition. Use this name when starting and stopping the export process for the partition.
<i>PART=part1, part2</i>	The partition ID used when <i>fakedbe=dbename</i> syntax is used to create separate processes per partition.

### Example

To add a SQL DBE entry for a database file:

```
D> SQLOUT PAYDBE4 .DATA .PROD , FOX ; HOME=3
D> SQLOUT PAYDBE4 .DATA .PROD , VIXEN ; HOME=3
```

Below is an example of some SQL shadowing using multiple export processes for one DBE:

```

D> SQLOUT GGDBE.PART1.PRODUCT=GGDBE.DATAB.PRODUCT,shadownode;HOME=1;EXECPRI=CS
D> SQLOUT GGDBE.PART4.PRODUCT=GGDBE.DATAB.PRODUCT,shadownode;HOME=4;EXECPRI=DS
D> SQLOUT CCDBE.DATAB.PRODUCT,shadownode;HOME=1;EXECPRI=DS;EXCLUDE
D> TABLE CCDBE.DATAB.PRODUCT,shadownode,TUPLEONE
D> TABLE CCDBE.DATAB.PRODUCT,shadownode,TUPLETWO
D> TABLE CCDBE.DATAB.PRODUCT,shadownode,TUPLETHREE
D> listsqlo @.@@
Node: MASTER      Directory: NBD.DATA.NETBASE

```

-----LOCAL DBE NAME-----			-----REMOTE EQUIVALENT-----				E	X	I # of
File	Group	Account	File	Group	Account	Node		T Part Freq	
GGDBE	PART1	PRODUCT	->	GGDBE	DATAB	PRODUCT	SHADOW	- 1 0	
Exec Pri: CS Partitions: 1									
GGDBE	PART4	PRODUCT	->	GGDBE	DATAB	PRODUCT	SHADOW	- 1 0	
Exec Pri: DS Partitions: 4									
CCDBE	DATAB	PRODUCT	->	--Same--			SHADOW	- 1 0	
Exec Pri: DS Partitions: 1									
Don't Shadow: TUPLEONE									
TUPLETHREE									
TUPLETWO									

To set up multiple export processes for one DBE:

```

D> SQLOUT fakedbe1=dbename,node;PART=part1,[part2[,...]][...]
D> SQLOUT fakedbe2=dbename,node;PART=...

```

## TABLE

The **TABLE** command works in conjunction with an **SQLOUT** command. The **SQLOUT** command indicates whether the associated **TABLE** commands specify tables to be included or excluded from shadowing. See the *NetBase SQL Shadowing Handbook* for more information.

### Syntax

```
D> TABLE dbname,node,tablename[,RESET]
```

### Parameters

Parameter	Description
<b>dbname</b>	The name of the Allbase database.
<b>node</b>	The name of the node to which this table should or should not be shadowed.
<b>tablename</b>	The name of the table being included or excluded (depending on the associated <b>SQLOUT</b> command) from shadowing.
<b>RESET</b>	If this option is specified, the table entry is removed from the directory.

### Example

Assuming **DBE1** includes five tables and we are shadowing it from **SYSA** to **SYSB**. To exclude **TABLE4** (shadowing the rest), you could enter:

On **SYSA**:

```
:NBDIR.PUB.NETBASE
D> SQLOUT DBE1.DATA.PROD,SYSB;HOME=1;EXCLUDE
D> TABLE DBE1.DATA.PROD,SYSB,TABLE4
```

On **SYSB**:

```
:NBDIR.PUB.NETBASE
D> SQLIN DBE1.DATA.PROD,SYSA
```

## Chapter 4

---

If **DBE2** includes 40 tables, and you want to shadow only the first three, the **INCLUDE** option is your best choice:

On **SYSA**:

```
:NBDIR.PUB.NETBASE  
D> SQLOUT DBE2.DATA.PROD,SYSB;HOME=1;INCLUDE  
D> TABLE DBE2.DATA.PROD,SYSB,TABLE1  
D> TABLE DBE2.DATA.PROD,SYSB,TABLE2  
D> TABLE DBE2.DATA.PROD,SYSB,TABLE3
```

On **SYSB**:

```
:NBDIR.PUB.NETBASE  
D> SQLIN DBE2.DATA.PROD,SYSA
```

## UNLOAD

This command is used to create an ASCII file containing the entries in NetBase's directory. The **UNLOAD** command creates a file which contains the commands necessary to rebuild the NetBase directory. This file may be edited and used to reload the directory. The **UNLOAD** command does not unload SQL Shadowing commands. This command requires **ACCOUNT MANAGER (AM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> UNLOAD filename
```

### Parameters

Parameter	Description
<i>filename</i>	The user-defined name of the MPE (ASCII) file containing the directory entries. If the file exists, it will be purged if requested.

### Example

To unload the directory into a file named **DIRBACK**:

```
D> UNLOAD DIRBACK
```

To reload the directory, based on the entries in the file named **DIRBACK**:

```
: RUN NBDIR.PUB.NETBASE;STDIN=DIRBACK
```

## XL

The **XL** command enters XL list records into the directory. The XL list is used in conjunction with the **PROGRAM** command to define the run string for a program to be run remotely or created locally through NetBase. This command requires **NODE MANAGER (NM)** or **SYSTEM MANAGER (SM)** capability.

### Syntax

```
D> XL name=[file[,file[,...]]]
```

### Parameters

Name	Description
<i>name</i>	The user-defined name of the XL list.
<i>file</i>	The name of an XL file as it should appear in the run command.

### Example

To define an XL list:

```
D> XL TESTXL=NMXL1.PUB,NMXL2.PUB  
D> LISTX TESTXL  
Directory: NBD.DATA.NETBASE
```

```
Name      XL List  
  
TESTXL    NMXL1.PUB,NMXL2.PUB
```

For more information, turn to Chapter 8 “AutoRPM”.

## CHAPTER 5

# NetBase Shadowing

NetBase shadowing provides the ability to maintain concurrent multiple copies of files and databases across one or more systems in a network. Since it does not rely on IMAGE logging, NetBase can shadow all types of disk files, not only databases.

## In This Chapter

- **Process Overview**
- **Configuration Parameters**
- **Defining a File for Shadowing - NBDIR Entries**
- **Setting Up Shadowing**
- **Third Party Shadowing (TPS) for TurboIMAGE databases**
- **Monitoring Shadowing**
- **Synchronization Loss**
- **Structural Changes for Shadowed Databases**
- **Flushing Shadow Updates**
- **User Exits**
- **Dataset Shadowing**
- **Shadowing With TurboIMAGE Log Files**
- **Shadowing with Dynamic Rollback Recovery**

## Process Overview

NetBase operates on the principle of master and shadow copies of files and databases. Each file that is to be shadowed will have a master copy to which all updates are performed. The updates are then transported and posted to the shadow copies. This guarantees that all copies will be exact duplicates of each other. In addition, updates occur to the shadow copy in the exact order that they occurred against the master.

By default, shadow copies of files and databases are available for read-only access. When an application is run on the shadow system and it opens the shadow copy in a read mode, then the application will read the data locally. If an application opens the shadow copy in an update mode and NetBase NFA has been purchased, NetBase will automatically NFA the access to the master system since all updates must occur on the master copy of the data. This default access can be overridden with options specified on the **FILE** statement.

NetBase shadowing is designed to function through all modes of operation and failure, including system backups, posting interruptions, and hardware or communication failures. If the shadow computer is unavailable, NetBase's export process will automatically queue all updates on the master computer until the shadow machine becomes available.

User exits are also available as a part of shadowing. Custom user exit procedures can be written for both the exporting and posting of data. Export user exits are available on the master system to allow the customization of shadowing by preprocessing the data before it is sent. Post user exits are available on the shadow system and provide the ability to do customized shadowing. Turn to the "Export User Exits" and "Post User Exits" sections in this chapter for more information.

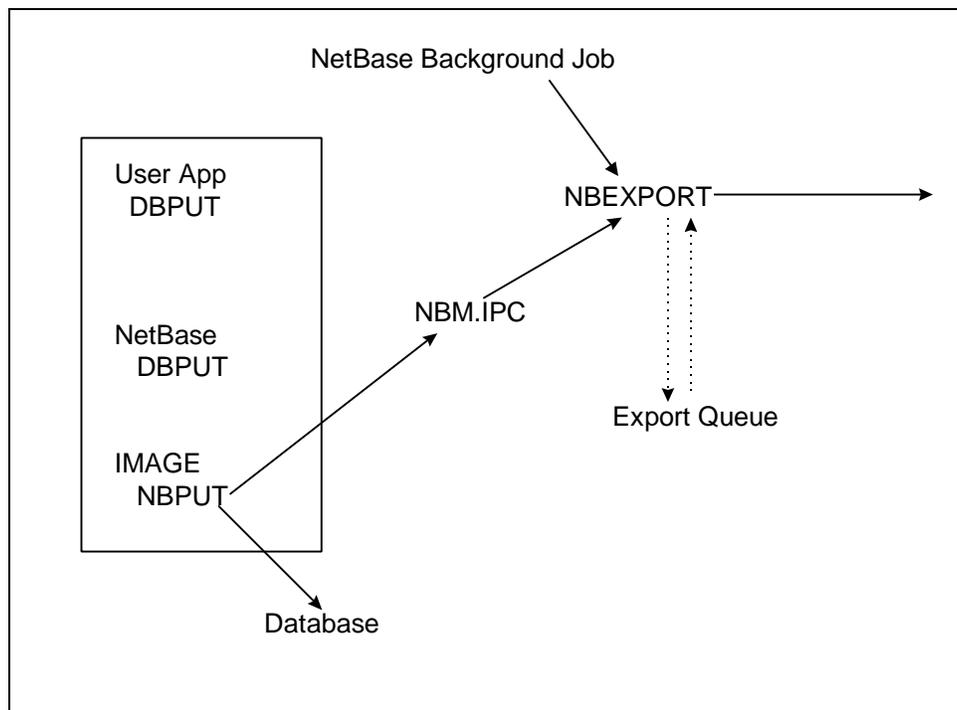
## Master System

The NetBase Shell captures updates on the master system as they occur. These updates are written to the shadow transport file, **NBM.IPC.NETBASE**. There are two methods for writing the captured transactions into the shadow transport file.

NetBase has been integrated with TurboIMAGE so that transactions to IMAGE databases are written to the shadow transport file during the IMAGE transactions by IMAGE. This is known as third party shadowing (TPS). The IMAGE procedure will call a NetBase routine from **XLNETBSE.PUB.SYS** during the IMAGE transaction which will write the transaction into the shadow transport file. For non-IMAGE files, the NetBase Shell will write the transactions to the shadow transport file.

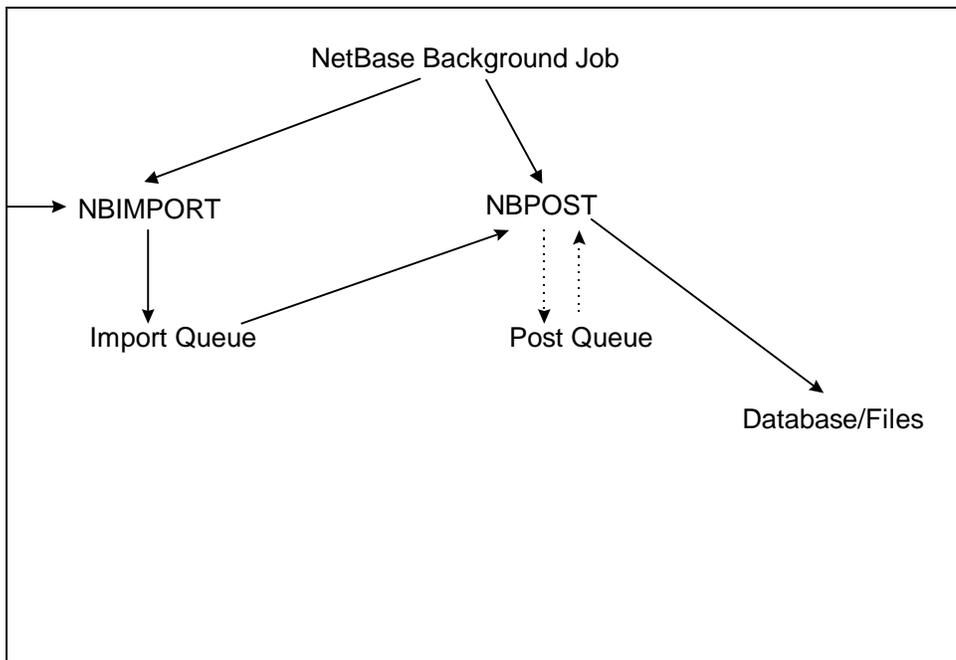
The process of reading the transactions from the **NBM** file is done by **NBEXPORT**. This is a process under the NetBase background job. If the NetBase background job is not running, the export process cannot read the transactions from the **NBM** file and send them to the shadow system.

The **NBEXPORT** process reads the transactions from the shadow transport file and sends them to the shadow system. If **NBEXPORT** cannot send the transactions to the shadow system, it will write the transactions into export queue files. Export queue files are contained in the **QUEUE.NETBASE** group. When communication is established with the shadow system, **NBEXPORT** will send any export queue files over to the shadow system. This could cause the transport file to fill up. If this happens, processes on the master system could begin to suspend while they wait to write to the **NBM** file.



## Shadow System

The shadow transactions are received by the **NBIMPORT** process. This process writes the transactions into import queue files. These files are contained in the **QUEUE.NETBASE** group. Next, the posting process, **NBPOST**, reads the transactions from the import queue files and applies them to the shadow copies. If the posting process is not running, the transactions will accumulate in the import queue files. If the posting process encounters a “non-fatal” error when trying to open the shadow copy, it will write the transactions for that file into a post queue file. These transactions can be recovered later once the problem has been resolved. There will be one **NBIMPORT** and **NBPOST** process for each master system that sends transaction to the shadow system. Turn to the “Queue State” section, under “Checking the State of Shadowed Files”, in this chapter for more information. Also see Appendix B “Troubleshooting NetBase”.



## Configuration Parameters

The operation of NetBase, including its various features, is defined by NetBase's configuration parameters. Configuration maintenance is a subsystem of the NetBase control program, **NBCTRL**. The configuration parameters can be displayed and modified using the configuration subsystem. The systems manager should become familiar with the control program, the configuration parameters, and the configuration maintenance commands in order to set up the shadowing facility.

Effective with version 9.6, shadow records are automatically added to the directory if the following conditions are met:

- The directory is large enough to hold all the records
- A shadow record with wildcards exists for the shadowed file



**Note.** Although shadow records for IMAGE databases may be auto-added, they will not work. This is because the password must be included in the shadow records for the database to be opened without password violations.

---

The parameters used in shadowing can be listed in the configuration maintenance program by entering the following commands. Normally, the default for these parameters is sufficient. If you suspect a problem with shadowing or want to change these parameters, please contact the Technical Support group before making any changes.

```
:NBCTRL.PUB.NETBASE
NetBase Control Program [x.x] Update 0 (C) QUEST Software 1996
```

```
*> CONFIG
C>L SHADOW
```

```
SHADOW - Close Log file on suspend. N
SHADOW - Flush Shadow File Posts... N
SHADOW - Errors Before Disable..... 1
SHADOW - Check KSAM record counts.. Y
SHADOW - Disable file autoresync .. N
SHADOW - Log Post Errors..... N
SHADOW - Enable Quiet Period Flush. N
SHADOW - Export User Exits..... N
SHADOW - Export Bridge Exits..... N
SHADOW - Export Process Info..... N
SHADOW - Process Info to Exp Exit.. N
SHADOW - Export Buffer Size (b).... 1457
SHADOW - Export Buffer Max BF..... 5
SHADOW - Export Buffer Timeout..... 10
SHADOW - Export Ack Timeout..... 120
SHADOW - Export Initial Priority... DEFAULT
SHADOW - Export Upshift Priority... CS
SHADOW - Export Upshift Trigger... 1500
SHADOW - Export Downshift Trigger.. 100
SHADOW - Shift Check Interval..... 50
SHADOW - Disable Shift Messages.... N
SHADOW - Enable Export Logging..... N
SHADOW - Export Log File Size..... 0
SHADOW - Export Log File Timeout... 100
SHADOW - Queue Full Warn Timeout... 180
SHADOW - Export Inactivity Timeout. 5
SHADOW - Import Inactivity Timeout. 5
SHADOW - Post Flush Timeout..... 5
SHADOW - Export on Startup..... Y
SHADOW - Import on Startup..... Y
SHADOW - Post on Startup..... Y
SHADOW - Use IMAGE Log Files..... N
```

```
C> exit
*> exit
```

The **Export Buffer** parameters change the way update transactions are buffered for transport across the network. You can fine tune this feature for maximum efficiency on your system. The **Export Initial Priority**, **Upshift Priority**, **Upshift Trigger**, **Downshift Trigger** and **Shift Check Interval** determine when and how the export process will change priority in response to changes in shadowed file activity.

There are also node level parameters associated with shadowing. These parameters affect each node in the system. Before modifying any of the node level shadowing parameters, please contact the Technical Support group. The node level parameters may be listed by entering the list command **L** followed by the node number or node name, in this case, **1**:



**Note.** If you have a node name that matches one of the **CONFIG** categories (**SHADOW** or **NFA**), you will have to use the node number.

C> L 1

```

Node Number..... 1
System Name..... MASTER
Startup Access (S/I/O/A/W)..... A
Node Name..... MASTER.QUESTS.COM
Node Connection Timeout..... 180
SPOOL - Direction (N/E/I/B)..... B
SPOOL - Allow incoming job streams. Y
SPOOL - Enable Export Flow Control. N
SHADOW - Direction (N/E/I/B)..... B
SHADOW - Export queue size..... 4096
SHADOW - Import queue size..... 4096
SHADOW - Post queue size..... 4096
SHADOW - Initial Post Priority.... DEFAULT
SHADOW - Post Inactivity Timeout .. 0
SHADOW - Enable Post Logging..... N
SHADOW - Log all Post Transactions. N
SHADOW - Save Process Info on Post. N
SHADOW - Abort Post on Errors..... N
SHADOW - Buffered Exporting..... Y
SHADOW - Transfer mode only..... N
SHADOW - Transfer Single Queues... N
SHADOW - Transfer ACK Frequency... 0
SHADOW - Post Directory Filename .. DEFAULT
    
```

The node level **SHADOW** parameters affect the queuing and logging of updates, process priority and the buffering feature. The rest of the parameters enable and modify other features of NetBase shadowing. Turn to Chapter 3 “Using CONFIG” for a description of each of these parameters, their default values and when they take effect in NetBase.

## Defining a File for Shadowing - NBDIR Entries

To define a file for shadowing, NetBase directory entries must be made on both the master and shadow systems. These directory entries tell NetBase which files to shadow and to which nodes. A **FILE** statement is entered in the directory on the master system, and a **FILE** and **SHADOW** statement is required on the shadow system.

This section details the steps required to start the shadowing process for a file or database. The set up on the master and shadow computers should be done at a quiet time, when user processes are not accessing the files. If configuration changes are required, read the parameter documentation to determine whether processes must be stopped for the parameter change to take effect.

NetBase uses a directory to locate files in the system that are to be shadowed. The directory maintenance subsystem is used to define a file or database for shadowing. Read the **NBDIR** chapter to learn how to run the subsystem and to familiarize yourself with the **FILE** and **SHADOW** directory commands.

### Directory Entries for Shadowing

The **FILE** directory command provides a number of options that affect the way processes access the master and shadow copies of a file or database. Some of the options are specific to the master copy of the file or database. Other options affect only the shadow copy. Turn to Chapter 4 “Using NBDIR” for the complete syntax for the **FILE** command.

```
D> FILE filename[=remote-name],masternode,shadownode[,shadownode...][;OPTIONS]
```



**Note.** No matter which system the **FILE** statement is used on, the nodes are always specified **masternode**, **shadownode**.

---

### Valid FILE Options for the Master System

The following table lists **FILE** command options for affecting processes that access the master file. For a full description of these options, turn to Chapter 4 “Using NBDIR”.

Option	Description
<b>SYNC</b>	Use for non-IMAGE files that do not have a locking strategy or with IMAGE databases if not using TPS shadowing.
<b>NOKFILES</b>	Do not shadow files with names that match the pattern <b>?#####</b> .
<b>EXCLUDE</b>	Do not shadow defined datasets.
<b>INCLUDE</b>	Shadow defined datasets only.
<b>READPASS</b>	Defines the shadow read password-not necessary if using TPS.
<b>CLOSE</b>	If this parameter is specified, the shadow copy of the file is closed when the posting of shadow transactions are completed.
<b>IMAGELOG</b>	This option identifies the master file as an IMAGE log file. Only one log file need be specified if automatic <b>CHANGELOGS</b> are enabled. Turn to the “Shadowing with TurboIMAGE Log Files” section in this chapter for more information on how <b>IMAGELOG</b> is used.
<b>NOLOG</b>	This option turns off the ability to log updates for this file.
<b>LOGONLY</b>	If this option is specified, all updates to the master file are logged. However, the updates are not sent to the shadow computer. The transactions are written to <b>ELOG####.LOG.NETBASE</b> where <b>####</b> is a sequence number.

### Valid FILE Options for the Shadow System

The following table lists **FILE** command options for affecting processes that access the shadow copy of the file. For a full description of these options, turn to Chapter 4 “Using NBDIR”.

Option	Description
<b>LOCAL</b>	Access local shadow copy only
<b>MASTER</b>	Access master copy only (requires NFA)
<b>READ</b>	Requires opens to be for read access

*Continued*

Option	Description
<b>LOCK</b>	For non-IMAGE files, open shadow file with lock access
<b>SYNCERR=FAIL</b>	Fail open on shadow file if out of sync
<b>SYNCERR=REMOTE</b>	Access master copy if out of sync (requires NFA)
<b>SHADOW</b>	This option is only valid if the shadow copy of a file exists on the same computer as the master copy. Specifying this option causes NetBase to treat this file as the shadow copy.
<b>DUALACC</b>	When this option is specified, both the shadow copy of the database and the master copy will be opened and accessed. When a call to <b>DBLOCK</b> occurs, the process is sent to the master system until a <b>DBUNLOCK</b> is called. This option can be used for an application that does heavy reading against a shadowed database, a small amount of updating, and is surrounded by <b>DBLOCK</b> and <b>DBUNLOCK</b> . This option requires an NFA license. Pending locks must be disabled on the shadow system.

## SHADOW Command Options

The **SHADOW** command provides a number of options that affect the way the shadow copy of a file or database is accessed or updated by the NetBase posting process. The posting process uses the **SHADOW** directory record when opening the file for updates and to mark the status of those updates to the shadow file. The **SHADOW** command is only issued on the system where the shadow copy of a file or database resides. The **SHADOW** command has the following format:

```
D> SHADOW filename [=copyfilename][;options]
```

The following table lists **SHADOW** command options for affecting the way the shadow copy of the file is posted. For a full description of these options, turn to Chapter 4 “Using NBDIR”.

Option	Description
<b>PASS</b>	Database password that grants write access
<b>DBX</b>	Needed if TPS is not used and using DBX Transactions
<b>LOG</b>	Enables Post Logging for a file
<b>USEREXIT</b>	Enables Post User Exits for this file
<b>MODE1</b>	Open shadow database in mode 1
<b>OVERRIDE</b>	This option applies only to MPE files. It directs the posting process to post file updates to a shadowed disk file ignoring certain synchronization errors, such as those that occur with the <b>FUPDATE</b> intrinsic.

## Examples of Directory Entries for Shadowing

This section provides three examples of making directory entries for shadowing from one system to another system.

### Example 1

Set up the directory entries necessary to shadow **BANNER.DATA.NETBASE** from **SYSA** to **SYSB**. In this example the remote name does not need to be specified since we are shadowing to the same group and account on **SYSB**.

On the master system, enter:

```
RUN NBDIR.PUB.NETBASE
D> FILE BANNER.DATA.NETBASE,SYSA,SYSB
```

On the shadow system, enter:

```
RUN NBDIR.PUB.NETBASE
D> FILE BANNER.DATA.NETBASE,SYSA,SYSB
D>
SHADOW BANNER.DATA.NETBASE;PASS=WRITER
```

### Example 2:

Set up the directory entries necessary to shadow **BANNER.DATA.NETBASE** on **SYSA** to **BANNER.DATA.BACKUP** on **SYSB**.

On the master system, enter:

```
RUN NBDIR.PUB.NETBASE
D> FILE BANNER.DATA.NETBASE=BANNER.DATA.BACKUP,SYSA,SYSB
```

On the shadow system, enter:

```
RUN NBDIR.PUB.NETBASE
D> FILE BANNER.DATA.BACKUP=BANNER.DATA.NETBASE,SYSA,SYSB
D> SHADOW BANNER.DATA.BACKUP;PASS=WRITER
```

**Example 3:**

Set up the directory entries necessary to shadow all the files in **DATA.PROD** from **SYSA** to **SYSB**. When shadowing entire groups or accounts, you must still specify an individual **SHADOW** statement for IMAGE databases. This is to provide **NBPOST** with the correct password to use for opening the shadow copy.

On the master system, enter:

```
RUN NBDIR.PUB.NETBASE
D> FILE @.DATA.PROD,SYSA,SYSB
```

On the shadow system, enter:

```
RUN NBDIR.PUB.NETBASE
D> FILE @.DATA.PROD,SYSA,SYSB
D> SHADOW @.DATA.PROD
D> SHADOW PAYDB.DATA.PROD;PASS=WRITER
```

Effective with version 9.6, shadow records are automatically added to the directory if the following conditions are met:

- The directory is large enough to hold all the records
- A shadow record with wildcards exists for the shadowed file



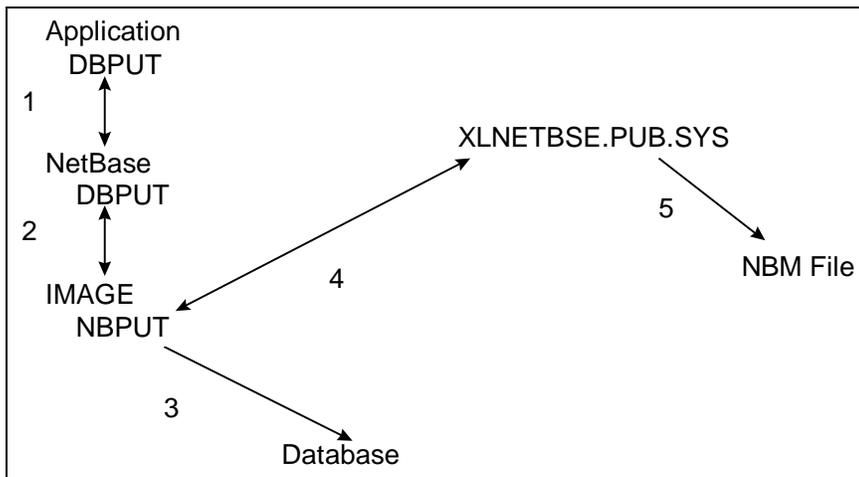
**Note.** Although shadow records for IMAGE databases may be auto-added, they will not work. This is because the password must be included in the shadow records for the database to be opened without password violations.

---

See the following section for instruction on adding Third Party Shadowing (TPS) information after the above examples.

## Third Party Shadowing (TPS) for TurboIMAGE databases

Hewlett Packard has added TPS (Third Party Shadowing) to IMAGE. This means that for every IMAGE **DBPUT**, **DBUPDATE** or **DBDELETE** NetBase will be called from within the IMAGE transaction. The advantage to NetBase is that the write to the **NBM** file is a part of the IMAGE transaction. The NetBase TPS procedures are in an XL named **XLNETBSE.PUB.SYS**. The file is created during the NetBase installation job.



**DBUTIL** has been changed to make it possible to enable or disable a database for TPS. If TPS is enabled on a database, but the database is not in the NetBase directory, shadowing will not occur. If a database is in the NetBase directory, but it is not enabled for TPS, shadowing will occur as it did in previous versions of NetBase. The NetBase shell will capture the transactions and write them to the NBM file.

The *first time* that a database is enabled for TPS, it must be enabled programmatically via a utility called **NBTPSUTL**. The utility, **NBTPSUTL**, will TPS enable the database specified in the **INFO** string, or TPS enable all the databases listed in a file that is specified in the **INFO** string. This utility only enables a base for TPS. After a database has been enabled via the **NBTPSUTL** program, **DBUTIL** must be used to disable a database for TPS. **DBUTIL** can also be used to re-enable it.

Run the utility by entering:

```
RUN NBTPSUTL;INFO=" [ ^ ]name "
```

**Name** is the name of a database, or if preceded by a carat (^), the name of a file containing a list of databases. The program prints a message for every database that the program attempts to TPS enable, for example:

```
RUN NBTPSUTL;INFO="^BASELIST"  
  
VISTA.DATA now enabled for TPS  
BANNER now enabled for TPS  
SWADB already enabled for TPS  
  
END OF PROGRAM
```

A database can be RE-enabled for TPS using **DBUTIL**. The command in **DBUTIL** is **ENABLE <database> FOR SHADOWING**. To disable a database for shadowing enter **DISABLE <database> for shadowing**. To display the current state of a database with regard to TPS enter **SHOW <database> FLAGS**.

The following display shows the **DBUTIL.PUB.SYS** for the **BANNER** database.

```
>>SHOW BANNER FLAGS  
For database BANNER  
  
Access is enabled.  
Autodefer is disabled.  
Dumping is disabled.  
Rollback recovery is disabled.  
Recovery is disabled.  
ILR is disabled.  
Mustrecover is disabled.  
Logging is disabled.  
Prefetch is disabled.  
Indexing is disabled.  
HWMPUT is disabled.  
Restart is disabled.  
Database last stored on THU, OCT 17, 1996, 8:06 AM.  
Database has not been modified since last store date.  
Shadowing is enabled for Netbase.  
Subsystem access is READ/WRITE.
```

## Setting Up Shadowing

Use the following steps to set up NetBase Shadowing on your master and shadow systems.

### To set up shadowing:

1. On the *shadow* system, stop the posting process.

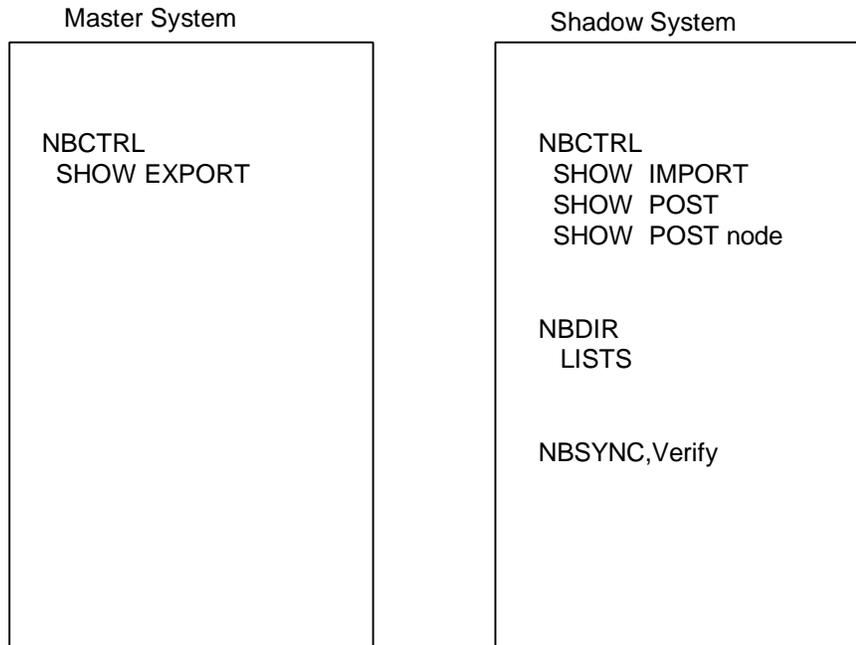
```
:NBCTRL.PUB.NETBASE;INFO="STOP POST"
```

2. On the *master* system, store the data.
3. On the *master* system, add directory entries for shadowing using **NBDIR**.
4. On the *master* system, enable databases for TPS using **NBTPSUTL**.
5. On the *master* system, allow users access to the data.
6. On the *shadow* system, restore the data.
7. On the *shadow* system, add directory entries using **NBDIR**.
8. On the *shadow* system, start the posting process.

```
:NBCTRL.PUB.NETBASE;INFO="START POST"
```

## Monitoring Shadowing

Within **NBCTRL**, the NetBase shadowing processes can be monitored. This section describes the commands that can be used for monitoring the master and shadow systems. See the following diagram.



On the master system, the **SHOW EXPORT** command gives a count of the total number of transactions sent to each shadow system. These values gets reset whenever the NetBase background job is stopped and restarted or whenever **EXPORT** is stopped and restarted.

```
:RUN NBCTRL.PUB.NETBASE
NetBase Control Program [x.x] Update 0 (C) QUEST Software 1996
```

```
*> SHOW EXPORT
```

```
No Backlog at this time. Priority is CS, shifted 1 times.
```

#	NODE	STATE	TRANSACTIONS SENT	TRANSACTIONS QUEUED
8	DAFFY	ACTIVE	105989	0
10	NORWAY	QUEUE	0	250

On the shadow system, the **SHOW IMPORT** command gives a count of the total number of transactions received from each master system. These values gets reset whenever the NetBase background job is stopped and restarted or whenever **IMPORT** is stopped and restarted.

```
:RUN NBCTRL.PUB.NETBASE
NetBase Control Program [x.x] Update 0 (C) QUEST Software 1996

*> SHOW IMPORT

# Node      State      Transactions  Transfer
           Received    % Done

7 BUGS     ACTIVE     105989
```

The **SHOW POST** command lists each master node for which it is posting transactions, and the current status of each node. It also lists whether or not there is a backlog and the current number of shadowed files that are open.

```
:RUN NBCTRL.PUB.NETBASE
NetBase Control Program [x.x] Update 0 (C) QUEST Software 1996

*> SHOW POST

Node Name      Status      Backlog    Files    Closes

7  BUGS        IDLE                1
```

The **SHOW POST node** command (where *node* is the node number of the master) gives detailed information on each shadowed file that is receiving post updates. It also gives a count of the number of updates and the current status.

```
:RUN NBCTRL.PUB.NETBASE
NetBase Control Program [x.x] Update 0 (C) QUEST Software 1987

*> SHOW POST 7

File                                     Type      Updates    Time Stamp of Last
                                           Record Processed    S Q D
                                           Y U I
                                           N E S

BANNER.DATA.NETBASE                     IMAGE     105989    08/22/96 13:44:37    - - -
TESTDB.DATA.NETBASE                     IMAGE         2001    08/22/96 14:40:50    S - -
MYFILE.TEST.NETBASE                     IMAGE     10005    08/22/96 13:44:37    - - -
```

In the previous display, the column **SYN** indicates whether or not the file is in sync. If there is an **S** in the column, that indicates a sync error. The column **QUE** indicates whether or not the transactions for a particular file are in a queue state. If a **Q** is present in this column, the posting process is writing all the transactions for that particular file into **Post Queues** for later recovery. Turn to Appendix B “Troubleshooting NetBase” for information on recovering files in a queue state. The last column, **DIS**, indicates that the file is in a discard state. If this flag is set, NetBase is discarding all transactions for that file.

## Shadow File States

The following table lists the four states that a shadow file can be in.

State	Description
<b>In sync (--)</b>	The master and shadow copy match. NetBase has not detected any errors.
<b>Queuing (Q)</b>	The posting process detected a non-fatal error when it tried to open the shadow copy. Updates for this file are being saved in a post queue and can be recovered.
<b>Out of sync (S)</b>	The posting process, <b>NBPOST</b> , has detected a sync error between the master and shadow copy. If this is an <b>IMAGE</b> database, the posting process will be discarding the transactions for the dataset that received the sync error and will continue posting the datasets that are in sync. For non- <b>IMAGE</b> files, all transactions for the file will be discarded.
<b>Discarding (D)</b>	NetBase has detected an error that is causing the posting process to discard all transactions for the shadowed file. If this is an <b>IMAGE</b> database, all transactions for all datasets are being discarded.

## Checking the State of Shadowed Files

The following sections describe two ways to check the state of shadowed files, one using **NBDIR.PUB.NETBASE** with the **LISTS** command, and the other using **NBSYNC.PUB.NETBASE, VERIFY**.

### NBDIR.PUB.NETBASE - LISTS Command

Using **NBDIR.PUB.NETBASE**, issue the **LISTS** command and check the status flags. If an **S** appears in the **SYN** column, that indicates the file is out of sync. The **Sync Lost at ...** message is the date and time of the most recent sync error.

```
:RUN NBDIR.PUB.NETBASE
D> lists @. @. ,info
Node: DAFFY      Directory: NBD.DATA.NETBASE

-----TRANSPORT REFERENCE-----          S Q U D S L
File      Group      Account      Node      ---LOCAL COPY REFERENCE--- Y U S I H O
N E R S R G
DIR1000X DATA      NFSDEV      BUGS      ->          --Same--          S - - D - -
Sync Lost at 03/03/96 13:27:08
  Last Update at 02/22/96 12:15:39
FURP0000 DATA      NFSDEV      BUGS      ->          --Same--          - - - - -
  Last Update at 01/18/96 14:54:19
HPUX0000 DATA      NFSDEV      BUGS      ->          --Same--          - - - - -
  Last Update at 04/20/96 12:17:20
INODE      DATA      NFSDEV      BUGS      ->          --Same--          S - - - - -
Sync Lost at 03/03/96 13:19:47
  Last Update at 04/20/96 12:17:20
```

### NBSYNC.PUB.NETBASE, VERIFY

Using the utility **NBSYNC.PUB.NETBASE** with the **VERIFY** entry point is another method for checking the sync status. This option on **NBSYNC** causes a **JCW, VERJCW**, to be set to a particular value depending on the status of the shadowed file.

```
:RUN NBSYNC.PUB.NETBASE,VERIFY;INFO="PAYDB.FILE.PAYROLL"
```



**Note.** If the **VERJCW** value is  $\neq 0$ , then you have some kind of problem with this file. Wildcards can be used in the **INFO** string.

---

### Checking for Synchronization:

The **SHADOW** statements in the NetBase directory on the shadow system are used by the posting process to flag whether or not a shadow file is in sync. The **LISTS** command in **NBDIR** can be used to see which files are out of sync or discarding. However, if you are shadowing numerous files, this display can be cumbersome and hard to read.

The utility, **NBSYNC**, can be used to check to see the state of shadowed files.

```
RUN NBSYNC.PUB.NETBASE,VERIFY;INFO="@. @.@"
```

The **VERIFY** entry point will instruct **NBSYNC** to go out and read the **SHADOW** statements in the NetBase directory. It will print the name of each file that is in a queue, discard or sync state. It will also set a JCW called **VERJWCW** which can be checked to see if files are in sync. The following table lists the **VERJWCW** values.

Value	Description
<b>WARN</b>	A file entry exists in the directory, but the file does not exist on the system.
<b>0</b>	Everything is okay with this file.
<b>1</b>	File is in the queue state.
<b>2</b>	File is out of sync.
<b>4</b>	File is in the discard state.

You can have a combination of these values. For example, if a file is out of sync and in a discard state, the value of **VERJWCW** would be 6.

### Example

```
RUN NBSYNC.PUB.NETBASE,VERIFY;INFO=' @. @. @'
```

```
NetBase Shadowing Reset Program [x.x] Update 0 (C) QUEST Software 1996
```

```
Enter shadow filename: @. @. @
CUSTDB.GROUP1.CLASS does not exist on this system
CUSTDB.GROUP3.CLASS does not exist on this system
CUSTDB.GROUP5.CLASS does not exist on this system
BANNER.KIM.NETBASE is DISCARDING
DB.TEST.NETBASE is NOT IN SYNC
```

```
END OF PROGRAM
BUGS:SHOWJCW
CIERROR = 0
JCW = 0
VERJWCW = 6
```

## Sample NBSYNC Job

```
!JOB CHCKSYNC,MGR.NETBASE
!COMMENT Checks for out of sync conditions
!COMMENT
!COMMENT Step 1 - Clean directory - the REMOVE entry
!COMMENT           point will remove any SHADOW statements
!COMMENT           from the directory that do not have a physical
!COMMENT           file on the shadow system
!RUN NBSYNC.PUB.NETBASE,REMOVE
!COMMENT
!COMMENT Step 2 - Check the state of shadowed files
!RUN NBSYNC.PUB.NETBASE,VERIFY;INFO="@.@.@"
!IF VERJCW <> 0 THEN
!  TELLOP ***WARNING*** - Shadowed files are not in Sync
!  TELLOP Check the STDLIST for the filenames
!ENDIF
!EOJ
```

## Synchronization Loss

When synchronization loss occurs, the shadow copy of a file or database will no longer match the original copy. Posting to the shadow copy may continue, depending on the **Errors Before Disable** configuration parameter, which determines how many sync problems will be tolerated before shadowing to the file is disabled and a full discard of updates is done. For databases, only datasets that experience sync problems will be disabled, while the remaining sets will continue to be updated as normal.

Once the shadow copy of a file or database has been flagged as no longer in sync with its master, synchronization can only be restored by taking a copy of the master file and restoring, or **DSCOPY**ing it onto the shadow system. This insures that the two copies match. Several methods exist that can be used to check if a file is in sync.

## Queue State

A file in a queue state can be recovered. The posting process is saving the transactions in **Post Queue** files. To recover a file in this state, do the following steps. For more detailed steps and an example, turn to Appendix B "Troubleshooting NetBase".

1. Check the NetBase **stdlist** for the error that the post encountered.
2. Fix the error.
3. Notify the posting process that the error has been fixed using the **ROLL FORWARD** command in **NBCTRL**.

## Sync or Discard State

Shadow files that are in either an **S** (out of sync) or **D** (discard) state, must be resynced. Resyncing a shadow copy involves refreshing the data on the shadow system with a copy of the data from the master system.

### To resync:

1. On the master system, stop access to the data. You must do this to create a store of the data.

2. On the shadow system, stop the posting process:

```
:RUN NBCTRL.PUB.NETBASE;INFO="STOP POST"
```

This needs to be done to ensure that all subsequent updates to the master are preserved in **Import Queue** files.

3. On the master system, store copy of the data. This will be used to refresh the shadow copy.
4. On the master system, allow access to the data. Updates will now be sent to the shadow system
5. On the shadow system, restore the data either using **NBSYNC** or the **RESTORE Utility**. If you use **NBSYNC** to do the **RESTORE**, it will automatically specify the **OLDDATE** option when doing the **RESTORE**. If using **RESTORE** or a **Third Party Utility**, be sure to specify the **OLDDATE** option. The **LAST MODIFY TIMESTAMP** on the file label is used as the **RESYNC** time. If the **OLDDATE** option is not used, the wrong resync time will be used and the shadow file will go out of sync again.

6. On the shadow system, run **NBSYNC** to reset directory flag and set resync timestamp:

```
:RUN NBSYNC.PUB.NETBASE,RESET
```

If **NBSYNC** was used to do the restore, this step is not necessary. However, if **RESTORE** or a **Third Party Utility** was used, this step must be done in order to reset the sync flag and set the resync time.

7. On the shadow system, start the posting process:

```
:RUN NBCTRL.PUB.NETBASE;INFO="START POST"
```

The posting process will automatically discard any transactions that are prior to the **RESYNC** time since it is assumed that those transactions were included in the data that was restored. Any transactions after the resync time are applied to the shadow copy. Transactions times are from the master system, the time on the shadow system is irrelevant.

### Example 1

This example uses **NBSYNC** to perform the **RESTORE** on the shadow system. Notice that **NBSYNC** issues the appropriate **STORE/RESTORE** command. Once the restore is complete, **NBSYNC** resets the shadow directory entry for the file **FT.TEST.QUEST**.

```
:RUN NBSYNC.PUB.NETBASE
```

```
NetBase Shadowing Reset Program [x.x] Update n (C) QUEST Software 1996
```

```
Enter fileset to restore: FT@.TEST.QUEST
```

```
STORE/RESTORE, VERSION 2 (C) 1996 HEWLETT-PACKARD CO.  
THU, AUG 23, 1996, 10:15 AM
```

```
WILL RESTORE 7 FILES; NUMBER OF FILES ON TAPE = 7
```

FILENAME.GROUP	.ACCOUNT	LDN	ADDRESS	REEL	SECTORS	CODE
FT	.TEST	.QUEST	RESET TO: TUE, AUG 23, 1996, 9:10 AM			
FT01	.TEST	.QUEST	1%00552746	1	180	PRIV
FT02	.TEST	.QUEST	2%00160043	1	162	PRIV
FT03	.TEST	.QUEST	1%00160624	1	28	PRIV
FT04	.TEST	.QUEST	2%00174560	1	104	PRIV
FT05	.TEST	.QUEST	1%00155262	1	16	PRIV
FT06	.TEST	.QUEST	2%00174730	1	84	PRIV

```
FILES RESTORED: 7
```

```
END OF PROGRAM
```

To verify that the **SYNC** flag has been reset, use the **NBDIR** utility:

```

:RUN NBDIR.PUB.NETBASE
NetBase Directory Program [x.x] Update 0 (C) QUEST Software 1996

D> LISTS FT.TEST.QUEST,INFO
Node: BUGS      Directory: ALS.DATA.NETBASE

-----TRANSPORT REFERENCE-----      ---LOCAL COPY REFERENCE---
File      Group      Account      Node      File      Group      Account      N E R S R G
S Q U D S L
Y U S I H O

FT        TEST        QUEST        DAFFY      ->        --Same--      - - - - -
Resync Time is 08/23/96 09:10:54

```

As described above, **NBSYNC** has two basic functions. In normal mode, **NBSYNC** invokes the appropriate **:RESTORE** command (**OLDDATE** option specified) and then proceeds to reset all directory status flags for any files restored. It also notes the last modify time for each file restored.



**Note.** Also, if you wish to resync a database using a tape drive other than the default (**DEV=TAPE**), you can issue a file equation prior to running **NBSYNC**:

```

:FILE NBSYNC;DEV=DAT
:RUN NBSYNC.PUB.NETBASE, entry point

```

---

With the above equation, **NBSYNC** will expect the **DAT** tape drive to contain a tape with the data being restored.

To have the restore list sent to the device **LP**:

```

:FILE NBSYNCLP;DEV=LP
:RUN NBSYNC.PUB.NETBASE

```

## Example 2

If you use a backup software product at your site, you do not need to have **NBSYNC** do the restore for you. If you have some method of getting the files from the master to shadow system, then the only step required of **NBSYNC** is to reset the directory flags for the shadowed file.

If the utility that you use for the restore/copy has an **OLDDATE** option, then you can run **NBSYNC** with the **RESET** entry point. This tells NetBase to get the **LAST MODIFY** time stamp from the file and use this as the resync time.

1. Perform **RESTORE** with the backup software, specifying the **OLDDATE** option.
2. Reset the status flags for a file that has already been restored with the **OLDDATE** option:

```
:RUN NBSYNC.PUB.NETBASE,RESET
```

```
NetBase Shadowing Reset Program [x.x] Update n (C) QUEST Software 1996
```

```
Enter shadow filename: CUSTS.CUSTOMER.RPACCT
```

```
CUSTS.CUSTOMER.RPACCT has been reset to WED, APR 18, 1996, 11:16 AM
```

```
Enter shadow filename: //
```

### Example 3

You may also resync a file using a copy utility such as **DSCOPY**. These utilities, however, typically copy the file into a **NEW** file, rather than retaining the **OLD** file label information of the original file. In this case, it is necessary to tell NetBase the original modify date and time information from the master copy of the file.

For example, to resync the file **BANNER.DATA.NETBASE** that has been **DSCOPY**ed to the shadow system, it is first necessary to determine the last modify date and time of the master file:

```
:LISTF BANNER.DATA.NETBASE,3
*****
FILE: BANNER.DATA.NETBASE

FILE CODE : -400                FOPTIONS: BINARY, FIXED, NOCCTL, STD
BLK FACTOR: 1                   CREATOR  : **
REC SIZE:  256 (BYTES)          LOCKWORD: **
BLK SIZE:  256 (BYTES)          SECURITY--READ   : ANY
EXT SIZE:  18 (SECT)            WRITE          : ANY
NUM REC:   11                   APPEND        : ANY
NUM SEC:   32                   LOCK           : ANY
NUM EXT:    1                   EXECUTE       : ANY
MAX REC:   11                   **SECURITY IS ON
MAX EXT:    1                   FLAGS         : NO ACCESSORS
NUM LABELS: 6                   CREATED      : FRI, FEB 15, 1996,  3:33 PM
MAX LABELS: 6                   MODIFIED    : THU, MAR 19, 1996,  8:28 AM
DISC DEV #: 2                   ACCESSED    : THU, MAR 19, 1996,  8:28 AM
CLASS      : DISC               LABEL ADDR: **
SEC OFFSET: 1536
```

This shows that the last modify date and time for **BANNER.DATA.NETBASE** on the master system is Thursday, March 19, 1996 at 8:28 a.m. Using this information, run the **NBSYNC** utility with the **ENTER** entry point on the shadow system:

```
:REMOTE
SYS-B:RUN NBSYNC.PUB.NETBASE,ENTER

NetBase Shadowing Reset Program [x.x] Update n (C) QUEST Software 1996

Enter shadow filename: BANNER.DATA.NETBASE
Reset date/time [03/19/96 17:51]? 03/19/96 08:28
BANNER.DATA.NETBASE has been reset to THU, MAR 19, 1996,  8:28 AM
Enter shadow filename: //

END OF PROGRAM
:
```

## NBSYNC Syntax

The syntax for running **NBSYNC** is as follows:

```
:RUN NBSYNC.PUB.NETBASE[,entry point]
where
entry point = CLEAR
              CREATE
              ENTER
              DISCARD
              NOW
              REMOVE
              REMOVEALL
              RESET
              VERIFY
```

The optional alternate entry points work as follows:

Entry Point	Description
<b>CLEAR</b>	Has the same effect as resetting and reentering a shadow entry for a file.
<b>CREATE</b>	Creates shadow records for any permanent file on the system that is covered by a wildcard shadow record. These entries are used by a subsequent run of <b>NBSYNC</b> to set sync times for these files.
<b>ENTER</b>	Causes <b>NBSYNC</b> to prompt for the file name to be reset and for the last modify date and time of the master file. The existing modify date and time from the file label of the shadow file is displayed and you are prompted for the new date/time. The new date and time must be entered in the same format as the date and time that is displayed. This entry point is used whenever a shadow file is <b>resynced</b> using a copy utility ( <b>DSCOPY</b> ) or if the <b>NEWDATE</b> parm was used on the <b>RESTORE</b> .
<b>DISCARD</b>	Sets a file to the discard set in order to allow posting to continue after an abort.
<b>NOW</b>	Sets the reset time to the current date/time instead of the last modified date/time in the file label.
<b>REMOVE</b>	Deletes all automatically generated shadow records that do not have a corresponding permanent file on the shadow system. The <b>REMOVE</b> entry point can be run at any time. It does not delete shadow records for files in the queue or discard state.

*Continued*

Entry Point	Description
<b>REMOVEALL</b>	Deletes all automatically added entries, including ones in the queue state.
<b>RESET</b>	<p>Running <b>NBSYNC</b> using this entry point causes <b>NBSYNC</b> to not invoke the restore command, but to just prompt for the files to be reset. This is often used when it is desired to restore the files with a utility other than the MPE <b>RESTORE</b> command.</p> <p><b>Note!</b> The <b>RESET</b> entry point assumes the file has been restored using an <b>OLDDATE</b> option, which preserves the last modify time information of the original file. If the restore operation built the file new or otherwise changed the last modify time information, use the <b>ENTER</b> entry point described below.</p>
<b>VERIFY</b>	<p>Allows jobs and UDCs to check to see if a file is in the discard, sync or queue state by setting a JCW. When you run <b>NBSYNC</b> with the <b>VERIFY</b> entry point, specify the file (or fileset) in the <b>INFO</b> string. The following example shows the values for a JCW named <b>VERJCW</b>.</p> <pre data-bbox="591 932 1328 1003">:RUN NBSYNC.PUB.NETBASE,VERIFY;INFO="@.GROUP.ACCT" :SHOWJCW</pre> <p><b>VERJCW</b> values:</p> <ul style="list-style-type: none"> <li><b>0</b>      Everything is okay</li> <li><b>1</b>      File is in the <b>QUEUE</b> state</li> <li><b>2</b>      File is in the <b>SYNC</b> error state</li> <li><b>4</b>      File is in the <b>DISCARD</b> state</li> </ul> <p><b>Note:</b> If more than one value applies, they will be added.</p>

## Structural Changes for Shadowed Databases

NetBase maintains exact physical copies of files that are being shadowed. With TurboIMAGE databases, whenever a structural change (such as a capacity change, dataset packing or the creating of new datasets) is made to the master system, it must also be applied to the shadow database.

Handling structural changes can be done three ways: through the 1) the **ADAGER** interface 2) the **DBGENERAL** interface, or by 3) manually coordinating updates. NetBase has an interface with both **ADAGER** and **DBGENERAL**. If you are using a utility other than **ADAGER** or **DBGENERAL**, you will need to manually coordinate the structural changes.

### 1) NetBase ADAGER Batch Interface

NetBase can now shadow **ADAGER** changes. It works by capturing the input to **ADAGER** on the master system and sending the input to the shadow system. A special command file **ADAGER.PUB.NETBASE** is executed on the master side which captures all of the **ADAGER** commands and puts them into a special job stream. The **NBADAGER.PUB.NETBASE** program runs when this job stream executes to execute the real **ADAGER** program. Once **ADAGER** has successfully completed the changes, the **NBADAGER** program then writes all of the **ADAGER** commands into the shadow transport file, **NBM.IPC.NETBASE**, and they are sent to the shadow system.

When the posting process **NBPOST** receives the **ADAGER** commands, it automatically puts the database into a queue state. Next, **NBPOST** invokes **ADAGER** on the shadow system to make the changes. Once the **ADAGER** changes successfully complete, **NBPOST** does a roll forward on the database.



**Requirements.** You must be running a version of **ADAGER** equal to or greater than 941201 and NetBase 9.6.

---

#### On the master system:

1. Execute the command file **ADAGER.PUB.NETBASE**. This runs **ADAGER.PUB.REGO**. You will be prompted for the name of a job stream, the database name and the changes to be made. When this step is complete, a job stream will have been created.
2. Stream the job created in step 1. This executes the **NBADAGER** program which in turn executes the **ADAGER** changes and sends them to the shadow system.

## Notes

- The command file, **ADAGER.PUB.NETBASE**, must be used.
- Only one database can be modified per execution of the command file.
- Online changes made with **ADAGER** are not shadowed.
- **ADAGER.PUB.NETBASE** will work when the data set is out of sync or queuing. The out of sync or queue state will not change, but the capacity change will happen.

## 2) NetBase DBGGENERAL Interface

NetBase also has an interface with **DBGGENERAL**. When **DBGGENERAL** is used to do maintenance to a master copy of a file, then changes will automatically be sent to the shadow system and performed there as well. This maintenance operation may be performed online or in batch. The interface works by capturing the input to **DBGGENERAL** on the master system and sending it over to the shadow system. A special program **NBGENRL.PUB.NETBASE** is used to write the **DBGGENERAL** changes to the shadow transport file, **NBM.IPC.NETBASE**.

When the posting process **NBPOST** receives the **DBGGENERAL** commands, it automatically puts the database into a queue state. Next, **NBPOST** invokes **DBGGENERAL** on the shadow system to perform the changes. Once **DBGGENERAL** has successfully completed on the shadow system, **NBPOST** does a **ROLL FORWARD** automatically on the database. The **STDLIST** of this **DBGGENERAL** process is created as a SPOOL file with the name **DBGENOUT.out.hpspool**.

## 3) Manually Coordinating the Structural Changes

There are several ways that this can be accomplished. One method is to make the changes on the master database and then restore that database to the shadow machine. This in effect produces a “resync” of the database. Another method outlined below produces the least amount of user downtime.

**On the master system:**

1. Get all users out of the master database.
2. Flush all of the records through to the shadow database and then stop posting in the shadow node. This can be accomplished by running the following program on the master machine:

```
RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM=node
```

When this program finishes, all the records for the shadow machine will be flushed and posted and then the posting process will be stopped. The node number specified is the node number of the shadow node.

3. Make a backup copy of the database. This is only a precautionary measure should the capacity change fail.
4. Make your database changes with your favorite database tool. (Remember these tools should not go through NetBase.)
5. Let the users back onto the master database.

**On the shadow system:**

1. Make your capacity changes (the same ones made to the master).
2. Run **NBCTRL** and enter the **START POST** command. This restarts the posting process.

## Flushing Shadow Updates

It can be useful to know when all updates have been posted to the shadow copies after the update processes have completed. NetBase provides a process called **NBFLUSH** that can be used to accomplish this task. The **NBFLUSH** utility is used to flush updates to the shadow copy of a file so it can be released by the posting process.

**NBFLUSH** is run on the machine where the master copy of the file resides. The program writes a special record to the shadow log file. When that record is read by the posting process on the shadow machine, the posting process closes the shadow copy of the file and/or stops completely, based on the entry point specified when run.

### NBFLUSH Syntax

```
:RUN NBFLUSH.PUB.NETBASE[,entry point][;INFO="filename"];PARM=node
where
entry point = CHECKPOINT
              CHECKWAIT
              STOP
              STOPWAIT
              QUIET
              QUIETWAIT
              WAIT
```

If **NBFLUSH** is run without an entry point, writes a stop record for the file specified on the node specified and then completes. When the posting process receives the stop record, it closes the file specified. The alternate entry points and the values for **INFO** and **PARM** operate as described in the following table.

Entry Point	Description
<b>CHECKPOINT</b>	The <b>CHECKPOINT</b> entry point is used to flush all logged updates to a file that is shadowed using IMAGE log files.
<b>CHECKWAIT</b>	The <b>CHECKWAIT</b> entry point is also used to flush updates to a file that is shadowed using IMAGE log files. However, <b>NBFLUSH</b> will wait for an acknowledgment from the posting process on the remote node before terminating.

*Continued*

Entry Point	Description
<b>STOP</b>	The <b>STOP</b> entry point is used to flush updates for all the files on a node. When using this entry point, the <b>INFO</b> parameter is not required, <b>NBFLUSH</b> will write a stop record for the node specified and then terminate. When the posting process receives the stop record, all files will be closed and the post process will enter the <b>STOP</b> state.
<b>STOPWAIT</b>	The <b>STOPWAIT</b> entry point is used to perform the same function as the <b>STOP</b> entry point except that <b>NBFLUSH</b> will wait for a message from the posting process stating that the task is complete before terminating.
<b>QUIET</b>	The <b>QUIET</b> entry point is used to flush updates for all the files on a node. It works just like the <b>STOP</b> entry point except that <b>NBPOST</b> will stop only when the count of DBX transactions reaches zero.  <b>Note:</b> This option must be enabled by the <b>NBCTRL</b> option <b>SHADOW - Enable Quiet Period Flush</b> .
<b>QUIETWAIT</b>	The <b>QUIETWAIT</b> entry point is used to perform the same function as the <b>QUIET</b> entry point except that <b>NBFLUSH</b> will wait for a message from the posting process stating that the task is complete before terminating.  <b>Note:</b> This option must be enabled by the <b>NBCTRL</b> option <b>SHADOW - Enable Quiet Period Flush</b> .
<b>WAIT</b>	An alternate entry point into <b>NBFLUSH</b> . Using this entry point causes <b>NBFLUSH</b> to wait for a message from the posting process stating that the task is complete before terminating.
<i>filename</i>	The file name of the shadowed file or log file as it appears in the NetBase directory. The <b>INFO</b> string, including this parameter, should not be used with the <b>STOP</b> or the <b>STOPWAIT</b> entry points.
<i>node</i>	This is the node number on which the shadow copy exists. Because a file can be shadowed to more than one node, this parameter is useful in directing <b>NBFLUSH</b> to the required node. If used with the <b>STOP</b> or the <b>STOPWAIT</b> entry points, this is the node number for which all updates will be flushed.

**NBFLUSH** acts as a means of marking the end of updates to the shadow copy of a file or database. The program sends a message to the posting process for the specified shadowed file. The message causes the posting process to close the shadow copy.

The program may be run with alternate entry points to provide optional features. If the program is run with the **WAIT** entry point specified, the posting process sends a message back to **NBFLUSH** before terminating. If any of the processes that propagate the message between **NBFLUSH** and the posting process are down, an error message is returned to **NBFLUSH**. This works slightly differently if shadowing is being performed using IMAGE log files. First, use the alternate entry point **CHECKPOINT** or **CHECKWAIT**. Second, the name of the log file as it appears in the directory should be passed in the **INFO** string. Also, **NBFLUSH** does not terminate until the log file has been completely read and the process for that log file has stopped.

The **CHECKWAIT** entry point is valid only if shadowing is being performed using IMAGE log files. It can be used to check that the updates for a particular process have completed. In this case, the flush message is passed to the posting process only when all records have been read that were written to the log file up to the time that **NBFLUSH** was run.

Running **NBFLUSH** with the **STOP** or **STOPWAIT** entry points is used to flush updates to all the shadow copies on a particular node and stop the posting process. This operates much the same as flushing updates to one particular file, the difference being that the posting process is directed to stop itself after the stop record is received. This option is commonly used prior to performing a backup of the shadow copy of files since it ensures that all updates are posted, relative to the time that **NBFLUSH** was run, and it directs the posting process to stop, thus freeing the files for the backup. This also causes all subsequent updates to the master files to be queued until the backup is complete and the posting process restarted.

The **QUIET** and **QUIETWAIT** entry points work just like the **STOP/STOPWAIT** entry points except that **NBPOST** stops only when the count of DBX transactions has reached zero (no DBX transactions in process). This option needs to be enabled by the **NBCTRL, CONFIG** option of **SHADOW - Enable Quiet Period Flush**.



**Note:** The counts are stored in the file **TICOUNTS.DATA.NETBASE**. If something goes wrong with the counts, this file can be purged. If a sync error is detected, this feature will disable itself.

---

In all cases, when **NBFLUSH** completes, it sets a JCW named **NBFLUSH**. If the value of **NBFLUSH** is other than zero, an error has occurred. The possible values for **NBFLUSH** and their meaning are shown in the following table.

Value	Description
0	Successful, no errors. If a wait option was specified, the task is complete on the shadow system.
1	NetBase is not running.
2	NetBase log control process is not running.
3	The NetBase log process is not running.
4	The shadow export process is not running.
5	The link to the remote computer is not active.
6	The posting process is not running on the remote node.
7	Unable to retrieve log file information.

## Backups Using Shadow Data

NetBase Shadowing provides a flexible and easy system for doing backups without shutting down access to files or databases on the master system. Backups can be made of the shadow copies of files without interrupting access to the original files. All that needs to be done is to stop one of the processes that transfer updates between the original files and the shadow files. Since NetBase queues all updates to the files while the process is down, updates made while the files are being backed up are not lost.

Whether the shadow copies reside on the same machine as the original copies, or whether files in the network are shadowed to a remote system, the steps are the same. Use the following steps for backing up files with NetBase Shadowing

**To backup files:**

1. Run **NBCTRL** and enter the **STOP POST** command on the system where the shadow files reside. The import process will start queuing updates to the shadow files.
2. Store the shadow files to tape.
3. Run **NBCTRL** and enter the **START POST** command. This will restart the posting process.

When the **START POST** command is issued, NetBase restarts all the posting processes (one process per import node). Posting automatically resumes for all files that were queued while posting was down.

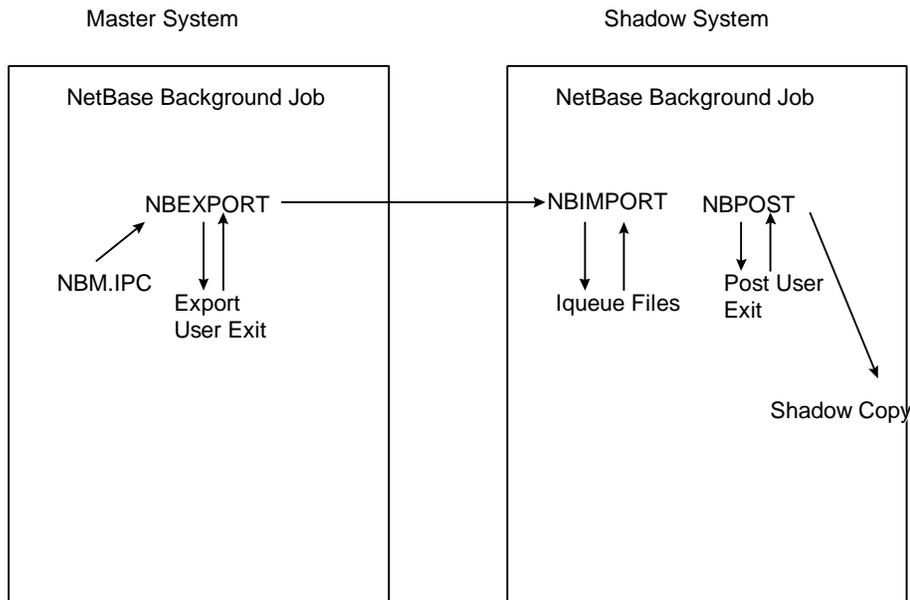


**Note.** If it is important that the backup reflect the state of the files as of a certain point in time (after online activity is completed, but before batch processing begins), the **NBFLUSH** utility can be used to generate a checkpoint. This checkpoint can be used to establish a point in time prior to which all transactions have been posted to the shadow files. The **NBFLUSH** utility is described in the “Flushing Shadow Updates” section.

---

## User Exits

Two types of user exits are available: export user exits or post user exits. Export user exits can be used to preprocess data before sending it to shadow system or to implement selective shadowing based on the value of the data. Post user exits are called by the posting process prior to writing to the shadow copy. They can be used to implement custom shadowing where a single transaction can be used to update multiple files.



## Export User Exits

Export user exits are procedures called by the NetBase shadow exporting process **NBEXPORT** to allow processing of shadow data beyond the normal shadowing processes. To use the export exit facility, the configuration parameter **SHADOW - Export User Exits** must be enabled.

The procedure called by **NBEXPORT** must be named **EXPORTEXIT**. Currently, the user exit procedure must be in either the **XL.NB.NETBASE**, **XL.PUB.NETBASE** or **XL.PUB.SYS** libraries. The user exit procedure will be dynamically loaded by the **NBEXPORT** program when needed. In order to update the user exit procedure, exporting must be stopped to unload the procedure.



**Note.** Since only one user exit procedure name is currently supported, the procedure must have the logic necessary to process all files simultaneously. This typically requires simple switching logic to other procedures.

---

All records that pass through the Shadow Export Transport are sent to the **EXPORTEXTIT**. The procedure can determine the disposition of each record and can individually control logging if necessary. The following section describes the calling conventions for the **EXPORTEXTIT** procedures.

### EXPORTEXTIT Syntax

**EXPORTEXTIT**(comarea, userarea, validnodes, exportnodes, header, orig, new)

## Parameters

Parameter	Description																														
<b>comarea</b>	<p>An array of various information used to control the posting of shadow information. Some fields are reserved and cannot be modified. The format is:</p> <table border="1"> <thead> <tr> <th data-bbox="483 533 553 562"><u>Bytes</u></th> <th data-bbox="581 533 678 562"><u>Contents</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="483 583 532 613"><b>0..1</b></td> <td data-bbox="581 583 1344 646"><b>Action flag.</b> Set by the user exit procedure to determine the action to be taken by the exporting process. The valid values are:</td> </tr> <tr> <td data-bbox="581 667 597 697"><b>0</b></td> <td data-bbox="678 667 945 697">Process record normally.</td> </tr> <tr> <td data-bbox="581 718 597 747"><b>1</b></td> <td data-bbox="678 718 954 747">Ignore record completely.</td> </tr> <tr> <td data-bbox="581 768 597 798"><b>2</b></td> <td data-bbox="678 768 1230 798">Disable all subsequent calls to the user exit facility.</td> </tr> <tr> <td data-bbox="581 819 597 848"><b>3</b></td> <td data-bbox="678 819 1091 848">Send record to ALL configured nodes.</td> </tr> <tr> <td data-bbox="581 869 597 898"><b>4</b></td> <td data-bbox="678 869 1318 932">Change node list from default. This must be specified if the <b>exportnodes</b> parameter is changed.</td> </tr> <tr> <td data-bbox="483 953 516 982"><b>-1</b></td> <td data-bbox="581 953 1302 1016">Abort exporting. This is a drastic situation which can cause later sync errors.</td> </tr> <tr> <td data-bbox="483 1037 532 1066"><b>2..3</b></td> <td data-bbox="581 1037 1344 1100"><b>Status.</b> Additional information used by the export program if an action such as <b>ABORT</b> is set. Used to display informative messages.</td> </tr> <tr> <td colspan="2" data-bbox="483 1121 1188 1150"><b>Note:</b> The above fields are set to <b>0</b> when the procedure is entered.</td> </tr> <tr> <td data-bbox="483 1171 532 1201"><b>4..5</b></td> <td data-bbox="581 1171 1334 1234"><b>Logging flag.</b> Set to <b>1</b> if logging for this record is enabled. Set to <b>0</b> if it is disabled.</td> </tr> <tr> <td data-bbox="483 1255 532 1285"><b>6..7</b></td> <td data-bbox="581 1255 1084 1285"><b>Local node number.</b> Node number of this node.</td> </tr> <tr> <td data-bbox="483 1306 532 1335"><b>8..9</b></td> <td data-bbox="581 1306 1075 1335"><b>Maximum node number possible.</b> Currently <b>64</b>.</td> </tr> <tr> <td data-bbox="483 1356 553 1386"><b>10..11</b></td> <td data-bbox="581 1356 1302 1419"><b>Size of exportnodes array.</b> Do not attempt to exceed this number of nodes (currently <b>12</b>).</td> </tr> <tr> <td data-bbox="483 1440 553 1470"><b>12..13</b></td> <td data-bbox="581 1440 766 1470">Unused. Set to <b>0</b>.</td> </tr> </tbody> </table>	<u>Bytes</u>	<u>Contents</u>	<b>0..1</b>	<b>Action flag.</b> Set by the user exit procedure to determine the action to be taken by the exporting process. The valid values are:	<b>0</b>	Process record normally.	<b>1</b>	Ignore record completely.	<b>2</b>	Disable all subsequent calls to the user exit facility.	<b>3</b>	Send record to ALL configured nodes.	<b>4</b>	Change node list from default. This must be specified if the <b>exportnodes</b> parameter is changed.	<b>-1</b>	Abort exporting. This is a drastic situation which can cause later sync errors.	<b>2..3</b>	<b>Status.</b> Additional information used by the export program if an action such as <b>ABORT</b> is set. Used to display informative messages.	<b>Note:</b> The above fields are set to <b>0</b> when the procedure is entered.		<b>4..5</b>	<b>Logging flag.</b> Set to <b>1</b> if logging for this record is enabled. Set to <b>0</b> if it is disabled.	<b>6..7</b>	<b>Local node number.</b> Node number of this node.	<b>8..9</b>	<b>Maximum node number possible.</b> Currently <b>64</b> .	<b>10..11</b>	<b>Size of exportnodes array.</b> Do not attempt to exceed this number of nodes (currently <b>12</b> ).	<b>12..13</b>	Unused. Set to <b>0</b> .
<u>Bytes</u>	<u>Contents</u>																														
<b>0..1</b>	<b>Action flag.</b> Set by the user exit procedure to determine the action to be taken by the exporting process. The valid values are:																														
<b>0</b>	Process record normally.																														
<b>1</b>	Ignore record completely.																														
<b>2</b>	Disable all subsequent calls to the user exit facility.																														
<b>3</b>	Send record to ALL configured nodes.																														
<b>4</b>	Change node list from default. This must be specified if the <b>exportnodes</b> parameter is changed.																														
<b>-1</b>	Abort exporting. This is a drastic situation which can cause later sync errors.																														
<b>2..3</b>	<b>Status.</b> Additional information used by the export program if an action such as <b>ABORT</b> is set. Used to display informative messages.																														
<b>Note:</b> The above fields are set to <b>0</b> when the procedure is entered.																															
<b>4..5</b>	<b>Logging flag.</b> Set to <b>1</b> if logging for this record is enabled. Set to <b>0</b> if it is disabled.																														
<b>6..7</b>	<b>Local node number.</b> Node number of this node.																														
<b>8..9</b>	<b>Maximum node number possible.</b> Currently <b>64</b> .																														
<b>10..11</b>	<b>Size of exportnodes array.</b> Do not attempt to exceed this number of nodes (currently <b>12</b> ).																														
<b>12..13</b>	Unused. Set to <b>0</b> .																														

The following additional fields are available only if the **Export Process Info** configuration option is set on the master machine. The **Process Info to Exp Exit** option must be set in **NBCTRL, CONFIG**. All of these fields will be set to binary zeros (**0**) if these options are either not enabled or not available.

Parameter	Description
<b>comarea</b>	<p><b>Bytes</b>    <b>Contents</b></p> <p><b>14..37</b>    <b>Program name.</b> Program name of the updating program.</p> <p><b>38..45</b>    <b>Session/Job name.</b> The session/job name of the updating user.</p> <p><b>46..53</b>    <b>User name.</b> The user name of the updating user.</p> <p><b>54..61</b>    <b>Logon group.</b> The logon group of the updating user.</p> <p><b>62..69</b>    <b>Account name.</b> The account name of the updating user.</p> <p><b>70..71</b>    <b>LDEV.</b> The LDEV of the updating user's <b>\$STDIN</b>.</p> <p><b>72..73</b>    <b>MODE.</b> The <b>MODE</b> from the <b>WHO</b> intrinsic.</p> <p><b>74..77</b>    <b>Job/Session number.</b> The job/session number of the updating user.</p> <p><b>78..81</b>    <b>Capability mask.</b> The capability mask from the <b>WHO</b> intrinsic.</p>
<b>userarea</b>	A 128 byte field initialized to binary zeros. This field is used by the user exit procedure. It retains its values between calls.
<b>validnodes</b>	A list of statuses for all possible node numbers. Each 16 bit integer will either be <b>0</b> (for invalid node) or a positive value, which signifies that the node is valid. The first integer is the status for node 1, the second for node 2, and so on.
<b>exportnodes</b>	<p>A list of nodes specifying where this record is to be sent. It is terminated by a <b>0</b> or by specifying the maximum number of nodes (see <b>comarea</b> field, <b>10..11</b>). If the <b>Send to ALL nodes</b> action is set, this is ignored.</p> <p><b>Note:</b> Unlike <b>validnodes</b>, this array contains actual node numbers. When using Action 4, be sure the node list is terminated by a <b>0</b>, or you will have problems.</p>

*Continued*

Parameter	Description																				
<b>header</b>	<p>Contains all non-data information about the original update. This field contains the following:</p> <table border="0"> <thead> <tr> <th><u>Bytes</u></th> <th><u>Contents</u></th> </tr> </thead> <tbody> <tr> <td>0..23</td> <td><b>File name.</b> This is the file name of the original, master copy of the file. The name is formatted as <b>FILE, GROUP, ACCOUNT</b> with no delimiters.</td> </tr> <tr> <td>24..25</td> <td><b>Flag word #1.</b> Currently reserved.</td> </tr> <tr> <td>26..27</td> <td><b>Date.</b> CALENDAR format of original date of the update.</td> </tr> <tr> <td>28..31</td> <td><b>Time.</b> CLOCK format of original time of the update.</td> </tr> <tr> <td>32..33</td> <td><b>Type of update.</b> Will be one of the following values:            IB      IMAGE DBBEGIN            IE      IMAGE DBEND            IO      IMAGE DBOPEN            IP      IMAGE DBPUT            IU      IMAGE DBUPDATE            ID      IMAGE DBDELETE            FC      File close            FD      File delete (RIO)            FN      File rename            FP      File put (KSAM write)            FR      File remove (KSAM)            FU      File update            FW      File write (new record added)</td> </tr> <tr> <td>34..35</td> <td><b>Reserved field.</b></td> </tr> <tr> <td>36..39</td> <td><b>Record number.</b></td> </tr> <tr> <td>40..41</td> <td><b>Data length.</b> Length of data records in 16 bit words.</td> </tr> <tr> <td>42..43</td> <td><b>IMAGE:</b>            Dataset number. Files: Original <b>TCOUNT</b> parameter.</td> </tr> </tbody> </table>	<u>Bytes</u>	<u>Contents</u>	0..23	<b>File name.</b> This is the file name of the original, master copy of the file. The name is formatted as <b>FILE, GROUP, ACCOUNT</b> with no delimiters.	24..25	<b>Flag word #1.</b> Currently reserved.	26..27	<b>Date.</b> CALENDAR format of original date of the update.	28..31	<b>Time.</b> CLOCK format of original time of the update.	32..33	<b>Type of update.</b> Will be one of the following values: IB      IMAGE DBBEGIN IE      IMAGE DBEND IO      IMAGE DBOPEN IP      IMAGE DBPUT IU      IMAGE DBUPDATE ID      IMAGE DBDELETE FC      File close FD      File delete (RIO) FN      File rename FP      File put (KSAM write) FR      File remove (KSAM) FU      File update FW      File write (new record added)	34..35	<b>Reserved field.</b>	36..39	<b>Record number.</b>	40..41	<b>Data length.</b> Length of data records in 16 bit words.	42..43	<b>IMAGE:</b> Dataset number. Files: Original <b>TCOUNT</b> parameter.
<u>Bytes</u>	<u>Contents</u>																				
0..23	<b>File name.</b> This is the file name of the original, master copy of the file. The name is formatted as <b>FILE, GROUP, ACCOUNT</b> with no delimiters.																				
24..25	<b>Flag word #1.</b> Currently reserved.																				
26..27	<b>Date.</b> CALENDAR format of original date of the update.																				
28..31	<b>Time.</b> CLOCK format of original time of the update.																				
32..33	<b>Type of update.</b> Will be one of the following values: IB      IMAGE DBBEGIN IE      IMAGE DBEND IO      IMAGE DBOPEN IP      IMAGE DBPUT IU      IMAGE DBUPDATE ID      IMAGE DBDELETE FC      File close FD      File delete (RIO) FN      File rename FP      File put (KSAM write) FR      File remove (KSAM) FU      File update FW      File write (new record added)																				
34..35	<b>Reserved field.</b>																				
36..39	<b>Record number.</b>																				
40..41	<b>Data length.</b> Length of data records in 16 bit words.																				
42..43	<b>IMAGE:</b> Dataset number. Files: Original <b>TCOUNT</b> parameter.																				
<b>orig</b>	Original data before update. Only applicable for "update" and "delete" type records. Note that this will be the <i>entire</i> dataset entry or file record.																				
<b>new data</b>	New data after update. Only applicable for "update" type records. Same format as the <b>orig</b> .																				

---

## Post User Exits

The posting process posts updates to a file or a database that is being shadowed. Sometimes, however, additional processing is required. For instance, updates to a particular set of a database may need to be made to a second database. NetBase's posting process has a method of handling these special cases. When additional processing is required, the posting process calls a user-supplied subroutine. The procedure called by the posting process is called **POSTUSEREXIT**. Currently, the user exit procedure must be in either the **XL.NB.NETBASE**, **XL.PUB.NETBASE** or **XL.PUB.SYS** library.

The user exit procedure will be dynamically loaded by the **NBPOST** program when needed. In order to update the user exit procedure, posting must be stopped to unload the procedure.



**Note.** Since only one user exit procedure name is currently supported, the procedure must have the logic necessary to process all files simultaneously. This typically requires simple switching logic to other procedures.

---

Any shadow file may be selected for the user exit facility. Once a file has been selected, all shadow records will be passed to the user exit. The user exit then has the ability to process the data at will. When the user exit procedure returns, an action code set by the user exit procedure will direct **NBPOST** to further process the update record. These actions are described below.

To select a file for user exit processing, a **SHADOW** command such as the following must be entered in the **NBDIR** program:

```
D> SHADOW ARCUST.DATA.PROD;PASS=CUSTPASS;USEREXIT
```

This will inform the **NBPOST** program that the file or database must be processed by the user exit procedure.

### POSTUSERINIT

This section describes the calling conventions for the **POSTUSERINIT** procedure. This procedure is called the first time a transaction is received for a shadowed file. It allows initialization to take place before the **POSTUSEREXIT** procedure is called.

### Syntax

```
POSTUSERINIT (comarea,userarea,filename)
```

## Parameters

Parameter	Description																														
<b>comarea</b>	<p>An array of various information used to control the posting of shadow information. Some fields are reserved and must not be modified. The format is:</p> <table border="1"> <thead> <tr> <th data-bbox="483 533 553 562"><u>Bytes</u></th> <th data-bbox="581 533 678 562"><u>Contents</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="483 583 553 613"><b>0..1</b></td> <td data-bbox="581 583 1360 646"><b>Action Flag.</b> Set by the user exit procedure to determine the action taken by the posting process. The valid values are:</td> </tr> <tr> <td data-bbox="581 667 602 697"><b>0</b></td> <td data-bbox="678 667 1122 697">Successful call. Do not open shadow file.</td> </tr> <tr> <td data-bbox="581 718 602 747"><b>2</b></td> <td data-bbox="678 718 1252 747">Successful call. Open shadow file for normal posting.</td> </tr> <tr> <td data-bbox="581 768 602 798"><b>3</b></td> <td data-bbox="678 768 1252 831">Disable subsequent user exit calls. File will be posted normally.</td> </tr> <tr> <td data-bbox="581 852 602 882"><b>-1</b></td> <td data-bbox="678 852 932 882">Set discard flag for file.</td> </tr> <tr> <td data-bbox="581 903 602 932"><b>-2</b></td> <td data-bbox="678 903 1289 932">Set <b>SYNC LOST</b> flag. Reason goes in <b>STATUS</b> field below.</td> </tr> <tr> <td data-bbox="581 953 602 982"><b>-3</b></td> <td data-bbox="678 953 1300 982">Set <b>QUEUE</b> status. All records will be sent to a post queue.</td> </tr> <tr> <td data-bbox="483 1024 553 1054"><b>2..3</b></td> <td data-bbox="581 1024 1321 1087"><b>Status.</b> Additional information used by the post program if an action such as <b>SYNC LOST</b> is set. Used to display informative messages.</td> </tr> <tr> <td colspan="2" data-bbox="483 1108 1187 1138"><b>Note:</b> The above fields are set to <b>0</b> when the procedure is entered.</td> </tr> <tr> <td data-bbox="483 1159 553 1188"><b>4..5</b></td> <td data-bbox="581 1159 1101 1188"><b>Node number.</b> Original node number of this file.</td> </tr> <tr> <td data-bbox="483 1209 553 1239"><b>6..7</b></td> <td data-bbox="581 1209 1084 1239"><b>Local node number.</b> Node number of this node.</td> </tr> <tr> <td data-bbox="483 1260 553 1323"><b>8..11</b> used</td> <td data-bbox="581 1260 1305 1323"><b>File number.</b> Set by the user if action <b>0</b> is taken and action 2 will be in subsequent calls of <b>POSTUSEREXIT</b>.</td> </tr> <tr> <td data-bbox="483 1344 553 1449"><b>12</b>  zero</td> <td data-bbox="581 1344 1321 1449"><b>File type.</b> Must be <b>F</b> for MPE file, <b>K</b> for KSAM file and <b>I</b> for IMAGE database. Set by the user only if the file number is also set to a non-zero value.</td> </tr> <tr> <td data-bbox="483 1470 553 1499"><b>13</b></td> <td data-bbox="581 1470 667 1499">Unused</td> </tr> </tbody> </table>	<u>Bytes</u>	<u>Contents</u>	<b>0..1</b>	<b>Action Flag.</b> Set by the user exit procedure to determine the action taken by the posting process. The valid values are:	<b>0</b>	Successful call. Do not open shadow file.	<b>2</b>	Successful call. Open shadow file for normal posting.	<b>3</b>	Disable subsequent user exit calls. File will be posted normally.	<b>-1</b>	Set discard flag for file.	<b>-2</b>	Set <b>SYNC LOST</b> flag. Reason goes in <b>STATUS</b> field below.	<b>-3</b>	Set <b>QUEUE</b> status. All records will be sent to a post queue.	<b>2..3</b>	<b>Status.</b> Additional information used by the post program if an action such as <b>SYNC LOST</b> is set. Used to display informative messages.	<b>Note:</b> The above fields are set to <b>0</b> when the procedure is entered.		<b>4..5</b>	<b>Node number.</b> Original node number of this file.	<b>6..7</b>	<b>Local node number.</b> Node number of this node.	<b>8..11</b> used	<b>File number.</b> Set by the user if action <b>0</b> is taken and action 2 will be in subsequent calls of <b>POSTUSEREXIT</b> .	<b>12</b>  zero	<b>File type.</b> Must be <b>F</b> for MPE file, <b>K</b> for KSAM file and <b>I</b> for IMAGE database. Set by the user only if the file number is also set to a non-zero value.	<b>13</b>	Unused
<u>Bytes</u>	<u>Contents</u>																														
<b>0..1</b>	<b>Action Flag.</b> Set by the user exit procedure to determine the action taken by the posting process. The valid values are:																														
<b>0</b>	Successful call. Do not open shadow file.																														
<b>2</b>	Successful call. Open shadow file for normal posting.																														
<b>3</b>	Disable subsequent user exit calls. File will be posted normally.																														
<b>-1</b>	Set discard flag for file.																														
<b>-2</b>	Set <b>SYNC LOST</b> flag. Reason goes in <b>STATUS</b> field below.																														
<b>-3</b>	Set <b>QUEUE</b> status. All records will be sent to a post queue.																														
<b>2..3</b>	<b>Status.</b> Additional information used by the post program if an action such as <b>SYNC LOST</b> is set. Used to display informative messages.																														
<b>Note:</b> The above fields are set to <b>0</b> when the procedure is entered.																															
<b>4..5</b>	<b>Node number.</b> Original node number of this file.																														
<b>6..7</b>	<b>Local node number.</b> Node number of this node.																														
<b>8..11</b> used	<b>File number.</b> Set by the user if action <b>0</b> is taken and action 2 will be in subsequent calls of <b>POSTUSEREXIT</b> .																														
<b>12</b>  zero	<b>File type.</b> Must be <b>F</b> for MPE file, <b>K</b> for KSAM file and <b>I</b> for IMAGE database. Set by the user only if the file number is also set to a non-zero value.																														
<b>13</b>	Unused																														

The following **comarea** fields are available only if the **Export Process Info** configuration option is set on the master system and the **Save Process Info on Post** option is set on the shadow system. They will be set to all binary zeroes (0) if not enabled or not available.

Field	Description
<b>comarea</b>	<p><b>14..37</b> Program name of updating program.</p> <p><b>38..45</b> Session/jobname of updating user.</p> <p><b>46..53</b> User name of updating user</p> <p><b>54..61</b> Logon group of updating user</p> <p><b>62..69</b> Account name of updating user</p> <p><b>70..71</b> LDEV of updating user's \$STDIN</p> <p><b>72..73</b> MODE from the WHO intrinsic</p> <p><b>74..77</b> Job/session number of updating user</p> <p><b>78..81</b> Capability mask from the WHO intrinsic</p>
<b>userarea</b>	A 1024 byte field initialized to binary 0's. This field is for the use of the user exit procedure. It retains its values between calls to <b>POSTUSEREXIT</b> and <b>POSTUSERINIT</b> .
<b>filename</b>	Name of the master file. Name is formatted in 24 bytes without periods, as in <b>QUERY PUB SYS</b> . The next 8 bytes will be the contents of the <b>PASS</b> field specified in the <b>SHADOW</b> command, as detailed above.

## POSTUSEREXIT

This section describes the calling conventions for the **POSTUSEREXIT** procedure. This procedure is called each time a transaction is received for a shadowed file.

### Syntax

**POSTUSEREXIT (comarea,userarea,header,olddata,newdata)**

## Parameters

Parameter	Description																																		
<b>comarea</b>	<p>An array of various information used to control the posting of shadow information. Some fields are reserved and must not be modified. The format is:</p> <table border="1"> <thead> <tr> <th data-bbox="483 569 553 594"><u>Bytes</u></th> <th data-bbox="581 569 678 594"><u>Contents</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="483 621 526 646"><b>0..1</b></td> <td data-bbox="581 621 1349 680"><b>Action flag.</b> Set by the user exit procedure to determine the action taken by the posting process. The valid values are:</td> </tr> <tr> <td data-bbox="483 695 509 720">to <b>0</b></td> <td data-bbox="581 695 1328 753">Update successful. Increment update counter but do not post shadow file.</td> </tr> <tr> <td data-bbox="483 768 509 793"><b>1</b></td> <td data-bbox="581 768 1235 800">Record ignored. Do not increment the update count.</td> </tr> <tr> <td data-bbox="483 814 509 840"><b>2</b></td> <td data-bbox="581 814 834 846">Post normally.</td> </tr> <tr> <td data-bbox="483 856 509 882"><b>3</b></td> <td data-bbox="581 856 1252 915">Disable subsequent user exit calls. File will be posted normally.</td> </tr> <tr> <td data-bbox="483 930 509 955"><b>-1</b></td> <td data-bbox="581 930 932 961">Set discard flag for file.</td> </tr> <tr> <td data-bbox="483 976 509 1001"><b>-2</b></td> <td data-bbox="581 976 1305 1008">Set <b>SYNC LOST</b> flag. Reason goes in <b>STATUS</b> field (below).</td> </tr> <tr> <td data-bbox="483 1022 509 1047"><b>-4</b></td> <td data-bbox="581 1022 1341 1108">Remove the file from the <b>SHOW POST</b>. It is the responsibility of the user exit procedure to close any files or data bases opened by the user exit procedure.</td> </tr> <tr> <td data-bbox="483 1123 526 1148"><b>2..3</b></td> <td data-bbox="581 1123 1328 1182"><b>Status.</b> Additional information used by the post program if an action such as <b>SYNC LOST</b> is set. Used to display informative messages.</td> </tr> <tr> <td colspan="2" data-bbox="483 1209 1187 1241"><b>Note:</b> The above fields are set to 0 when the procedure is entered.</td> </tr> <tr> <td data-bbox="483 1262 526 1287"><b>4..5</b></td> <td data-bbox="581 1262 1101 1293"><b>Node number.</b> Original node number of this file.</td> </tr> <tr> <td data-bbox="483 1314 526 1339"><b>6..7</b></td> <td data-bbox="581 1314 1084 1346"><b>Local node number.</b> Node number of this node.</td> </tr> <tr> <td data-bbox="483 1367 537 1392"><b>8..11</b></td> <td data-bbox="581 1367 1341 1434"><b>File number.</b> For databases, bytes <b>10</b> and <b>11</b> will contain the base ID of the shadow copy of the database (if open).</td> </tr> <tr> <td data-bbox="483 1455 509 1480"><b>12</b></td> <td data-bbox="581 1455 1349 1522"><b>File type.</b> Will be <b>F</b> for MPE file, <b>K</b> for KSAM file or <b>I</b> for IMAGE data base. Set to <b>U</b> if no shadow file is open.</td> </tr> <tr> <td data-bbox="483 1543 509 1568"><b>13</b></td> <td data-bbox="581 1543 672 1575">Unused.</td> </tr> <tr> <td data-bbox="483 1596 548 1621"><b>14..81</b></td> <td data-bbox="581 1596 1328 1682">The updating user's process info (see <b>POSTUSERINIT</b> for details). Will only be present if certain configuration options are enabled. Will be all binary <b>0</b> if not enabled.</td> </tr> </tbody> </table>	<u>Bytes</u>	<u>Contents</u>	<b>0..1</b>	<b>Action flag.</b> Set by the user exit procedure to determine the action taken by the posting process. The valid values are:	to <b>0</b>	Update successful. Increment update counter but do not post shadow file.	<b>1</b>	Record ignored. Do not increment the update count.	<b>2</b>	Post normally.	<b>3</b>	Disable subsequent user exit calls. File will be posted normally.	<b>-1</b>	Set discard flag for file.	<b>-2</b>	Set <b>SYNC LOST</b> flag. Reason goes in <b>STATUS</b> field (below).	<b>-4</b>	Remove the file from the <b>SHOW POST</b> . It is the responsibility of the user exit procedure to close any files or data bases opened by the user exit procedure.	<b>2..3</b>	<b>Status.</b> Additional information used by the post program if an action such as <b>SYNC LOST</b> is set. Used to display informative messages.	<b>Note:</b> The above fields are set to 0 when the procedure is entered.		<b>4..5</b>	<b>Node number.</b> Original node number of this file.	<b>6..7</b>	<b>Local node number.</b> Node number of this node.	<b>8..11</b>	<b>File number.</b> For databases, bytes <b>10</b> and <b>11</b> will contain the base ID of the shadow copy of the database (if open).	<b>12</b>	<b>File type.</b> Will be <b>F</b> for MPE file, <b>K</b> for KSAM file or <b>I</b> for IMAGE data base. Set to <b>U</b> if no shadow file is open.	<b>13</b>	Unused.	<b>14..81</b>	The updating user's process info (see <b>POSTUSERINIT</b> for details). Will only be present if certain configuration options are enabled. Will be all binary <b>0</b> if not enabled.
<u>Bytes</u>	<u>Contents</u>																																		
<b>0..1</b>	<b>Action flag.</b> Set by the user exit procedure to determine the action taken by the posting process. The valid values are:																																		
to <b>0</b>	Update successful. Increment update counter but do not post shadow file.																																		
<b>1</b>	Record ignored. Do not increment the update count.																																		
<b>2</b>	Post normally.																																		
<b>3</b>	Disable subsequent user exit calls. File will be posted normally.																																		
<b>-1</b>	Set discard flag for file.																																		
<b>-2</b>	Set <b>SYNC LOST</b> flag. Reason goes in <b>STATUS</b> field (below).																																		
<b>-4</b>	Remove the file from the <b>SHOW POST</b> . It is the responsibility of the user exit procedure to close any files or data bases opened by the user exit procedure.																																		
<b>2..3</b>	<b>Status.</b> Additional information used by the post program if an action such as <b>SYNC LOST</b> is set. Used to display informative messages.																																		
<b>Note:</b> The above fields are set to 0 when the procedure is entered.																																			
<b>4..5</b>	<b>Node number.</b> Original node number of this file.																																		
<b>6..7</b>	<b>Local node number.</b> Node number of this node.																																		
<b>8..11</b>	<b>File number.</b> For databases, bytes <b>10</b> and <b>11</b> will contain the base ID of the shadow copy of the database (if open).																																		
<b>12</b>	<b>File type.</b> Will be <b>F</b> for MPE file, <b>K</b> for KSAM file or <b>I</b> for IMAGE data base. Set to <b>U</b> if no shadow file is open.																																		
<b>13</b>	Unused.																																		
<b>14..81</b>	The updating user's process info (see <b>POSTUSERINIT</b> for details). Will only be present if certain configuration options are enabled. Will be all binary <b>0</b> if not enabled.																																		

*Continued*

Parameter	Description																																																						
<i>userarea</i>	A 1024 byte field initialized to binary 0's. This field is for the use of the user exit procedure. It retains its values between calls.																																																						
<i>header</i>	<p>Contains all non-data information about the original update. This field contains the following:</p> <table border="1"> <thead> <tr> <th>Bytes</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0..23</td> <td><b>File name.</b> This is the file name of the original, master copy of the file. The name is formatted as <b>FILE, GROUP, ACCOUNT</b> with no delimiters.</td> </tr> <tr> <td>24..25</td> <td><b>Flag word #1.</b> Currently reserved.</td> </tr> <tr> <td>26..27</td> <td><b>Date.</b> CALENDAR format of original date of the update.</td> </tr> <tr> <td>28..31</td> <td><b>Time.</b> CLOCK format of original time of the update.</td> </tr> <tr> <td>32..33</td> <td><b>Type of update.</b> Will be one of the following values:                             <table border="1"> <tbody> <tr><td>IB</td><td>IMAGE DBBEGIN</td></tr> <tr><td>IE</td><td>IMAGE DBEND</td></tr> <tr><td>IO</td><td>IMAGE DBOPEN</td></tr> <tr><td>IP</td><td>IMAGE DBPUT</td></tr> <tr><td>IU</td><td>IMAGE DBUPDATE</td></tr> <tr><td>ID</td><td>IMAGE DBDELETE</td></tr> <tr><td>FC</td><td>File close</td></tr> <tr><td>FD</td><td>File delete (RIO)</td></tr> <tr><td>FE</td><td>Set EOF</td></tr> <tr><td>FK</td><td>File killed (purged)</td></tr> <tr><td>FL</td><td>File user label update</td></tr> <tr><td>FN</td><td>File rename</td></tr> <tr><td>FO</td><td>File open</td></tr> <tr><td>FP</td><td>File put (KSAM write)</td></tr> <tr><td>FR</td><td>File remove (KSAM)</td></tr> <tr><td>FU</td><td>File update</td></tr> <tr><td>FW</td><td>File write (new record added)</td></tr> </tbody> </table> </td> </tr> <tr> <td>34..35</td> <td><b>PIN</b> of original process</td> </tr> <tr> <td>36..39</td> <td><b>Record number.</b></td> </tr> <tr> <td>40..41</td> <td><b>Data length.</b> Length of data records in 16 bit words.</td> </tr> <tr> <td>42..43</td> <td><b>IMAGE:</b> Dataset number. <b>Files:Original TCOUNT</b> parameter.</td> </tr> </tbody> </table>	Bytes	Contents	0..23	<b>File name.</b> This is the file name of the original, master copy of the file. The name is formatted as <b>FILE, GROUP, ACCOUNT</b> with no delimiters.	24..25	<b>Flag word #1.</b> Currently reserved.	26..27	<b>Date.</b> CALENDAR format of original date of the update.	28..31	<b>Time.</b> CLOCK format of original time of the update.	32..33	<b>Type of update.</b> Will be one of the following values: <table border="1"> <tbody> <tr><td>IB</td><td>IMAGE DBBEGIN</td></tr> <tr><td>IE</td><td>IMAGE DBEND</td></tr> <tr><td>IO</td><td>IMAGE DBOPEN</td></tr> <tr><td>IP</td><td>IMAGE DBPUT</td></tr> <tr><td>IU</td><td>IMAGE DBUPDATE</td></tr> <tr><td>ID</td><td>IMAGE DBDELETE</td></tr> <tr><td>FC</td><td>File close</td></tr> <tr><td>FD</td><td>File delete (RIO)</td></tr> <tr><td>FE</td><td>Set EOF</td></tr> <tr><td>FK</td><td>File killed (purged)</td></tr> <tr><td>FL</td><td>File user label update</td></tr> <tr><td>FN</td><td>File rename</td></tr> <tr><td>FO</td><td>File open</td></tr> <tr><td>FP</td><td>File put (KSAM write)</td></tr> <tr><td>FR</td><td>File remove (KSAM)</td></tr> <tr><td>FU</td><td>File update</td></tr> <tr><td>FW</td><td>File write (new record added)</td></tr> </tbody> </table>	IB	IMAGE DBBEGIN	IE	IMAGE DBEND	IO	IMAGE DBOPEN	IP	IMAGE DBPUT	IU	IMAGE DBUPDATE	ID	IMAGE DBDELETE	FC	File close	FD	File delete (RIO)	FE	Set EOF	FK	File killed (purged)	FL	File user label update	FN	File rename	FO	File open	FP	File put (KSAM write)	FR	File remove (KSAM)	FU	File update	FW	File write (new record added)	34..35	<b>PIN</b> of original process	36..39	<b>Record number.</b>	40..41	<b>Data length.</b> Length of data records in 16 bit words.	42..43	<b>IMAGE:</b> Dataset number. <b>Files:Original TCOUNT</b> parameter.
Bytes	Contents																																																						
0..23	<b>File name.</b> This is the file name of the original, master copy of the file. The name is formatted as <b>FILE, GROUP, ACCOUNT</b> with no delimiters.																																																						
24..25	<b>Flag word #1.</b> Currently reserved.																																																						
26..27	<b>Date.</b> CALENDAR format of original date of the update.																																																						
28..31	<b>Time.</b> CLOCK format of original time of the update.																																																						
32..33	<b>Type of update.</b> Will be one of the following values: <table border="1"> <tbody> <tr><td>IB</td><td>IMAGE DBBEGIN</td></tr> <tr><td>IE</td><td>IMAGE DBEND</td></tr> <tr><td>IO</td><td>IMAGE DBOPEN</td></tr> <tr><td>IP</td><td>IMAGE DBPUT</td></tr> <tr><td>IU</td><td>IMAGE DBUPDATE</td></tr> <tr><td>ID</td><td>IMAGE DBDELETE</td></tr> <tr><td>FC</td><td>File close</td></tr> <tr><td>FD</td><td>File delete (RIO)</td></tr> <tr><td>FE</td><td>Set EOF</td></tr> <tr><td>FK</td><td>File killed (purged)</td></tr> <tr><td>FL</td><td>File user label update</td></tr> <tr><td>FN</td><td>File rename</td></tr> <tr><td>FO</td><td>File open</td></tr> <tr><td>FP</td><td>File put (KSAM write)</td></tr> <tr><td>FR</td><td>File remove (KSAM)</td></tr> <tr><td>FU</td><td>File update</td></tr> <tr><td>FW</td><td>File write (new record added)</td></tr> </tbody> </table>	IB	IMAGE DBBEGIN	IE	IMAGE DBEND	IO	IMAGE DBOPEN	IP	IMAGE DBPUT	IU	IMAGE DBUPDATE	ID	IMAGE DBDELETE	FC	File close	FD	File delete (RIO)	FE	Set EOF	FK	File killed (purged)	FL	File user label update	FN	File rename	FO	File open	FP	File put (KSAM write)	FR	File remove (KSAM)	FU	File update	FW	File write (new record added)																				
IB	IMAGE DBBEGIN																																																						
IE	IMAGE DBEND																																																						
IO	IMAGE DBOPEN																																																						
IP	IMAGE DBPUT																																																						
IU	IMAGE DBUPDATE																																																						
ID	IMAGE DBDELETE																																																						
FC	File close																																																						
FD	File delete (RIO)																																																						
FE	Set EOF																																																						
FK	File killed (purged)																																																						
FL	File user label update																																																						
FN	File rename																																																						
FO	File open																																																						
FP	File put (KSAM write)																																																						
FR	File remove (KSAM)																																																						
FU	File update																																																						
FW	File write (new record added)																																																						
34..35	<b>PIN</b> of original process																																																						
36..39	<b>Record number.</b>																																																						
40..41	<b>Data length.</b> Length of data records in 16 bit words.																																																						
42..43	<b>IMAGE:</b> Dataset number. <b>Files:Original TCOUNT</b> parameter.																																																						

*Continued*

Parameter	Description
<i>old data</i>	Original data before update. Only applicable for "update" and "delete" type records. Note that this will be the ENTIRE dataset entry or file record.
<i>new data</i>	New data after update. Only applicable for "update" and "put" type records. Same format as the "old data".

### Post User Exit Notes

The first thing to keep in mind when writing post user exits is that one procedure must handle all the files which require the use of this facility. In other words, the procedure must be able to go to the appropriate section of the code based on the file name alone. A general outline of the logic of a post user exit might be:

1. Determine if this is a known file. If not, return and set the action to 3. This will prevent additional calls to the user exit for this file.
2. Assuming this is a known file, see if the file has been opened. Normally, the file number can be stored in the **userarea** field. If it is open, skip to step 4.



**Note.** In many cases, a different data base or file from the original is opened.

---

3. If the file is not open, try to open it. If it cannot be opened, set the action value to **-1** or **-3**. These values cause the posting process to either discard or queue the updates, depending on what is desired.
4. Process the update. Since the record structure is known, the procedure can now update the information as needed. If anything goes wrong, the action value may be set to **-2**. This will cause a "sync lost" to be generated.
5. Return from the procedure.

## Post User Exit COBOL Example

Some sample declarations for the call:

```

01  EXIT-COMMON-AREA.
    05  ECA-ACTION      PIC S9(4) COMP.
        88  ECA-ACT-OK          VALUE 0.
        88  ECA-ACT-DISCARD     VALUE -1.
        88  ECA-ACT-SYNC-LOST   VALUE -2.
        88  ECA-ACT-QUEUE       VALUE -3.
        88  ECA-ACT-CLOSE       VALUE -4.
        88  ECA-ACT-IGNORE      VALUE 1.
        88  ECA-ACT-POST-IT     VALUE 2.
        88  ECA-ACT-DISABLE     VALUE 3.
    05  ECA-STATUS      PIC S9(4) COMP.
    05  ECA-FILE-NODE   PIC S9(4) COMP.
    05  ECA-THIS-NODE   PIC S9(4) COMP.
    05  ECA-FILE-NUMBER PIC S9(9) COMP.
    05  ECA-FILE-TYPE   PIC X.
        88  ECA-TYPE-MPE-FILE   VALUE "F".
        88  ECA-TYPE-KSAM-FILE  VALUE "K".
        88  ECA-TYPE-IMAGE     VALUE "I".
    05  FILLER PIC X.
01  EXIT-USER-AREA     PIC X(2048).
01  EXIT-HEADER-INFO.
    05  EHI-FILE-NAME   PIC X(24).
    05  FILLER          PIC X(2).
    05  EHI-DATE        PIC S9(4) COMP.
    05  EHI-TIME        PIC S9(9) COMP.
    05  EHI-UPDATE-TYPE PIC X(2).
        88  EHI-UPD-DBUPDATE   VALUE "IU".
        88  EHI-UPD-DBPUT      VALUE "IP".
        88  EHI-UPD-DBDELETE   VALUE "ID".
        88  EHI-UPD-FWRITE     VALUE "FW".
        88  EHI-UPD-FUPDATE    VALUE "FU".
        88  EHI-UPD-KSAM-ADD   VALUE "FP".
        88  EHI-UPD-KSAM-DEL   VALUE "FR".
        88  EHI-UPD-RIO-DEL    VALUE "FD".
    05  FILLER          PIC X(2).
    05  EHI-RECORD-NUMBER PIC S9(9) COMP.
    05  EHI-DATA-LENGTH  PIC S9(4) COMP.
    05  EHI-DATA-SET     PIC S9(4) COMP.
    05  EHI-TCOUNT REDEFINES EHI-DATA-SET
                                PIC S9(4) COMP.
01  OLD-DATA.
    05  OLD-KEY-VALUE    PIC X(8).
    05  FILLER          PIC X(20).
    05  OLD-QTY         PIC S9(9) COMP.
    05  FILLER          PIC X(100).

```

## Chapter 5

---

```
01    NEW-DATA.  
      05    NEW-KEY-VALUE    PIC X(8).  
      05    FILLER          PIC X(20).  
      05    NEW-QTY        PIC S9(9) COMP.  
      05    FILLER          PIC X(100).  
  
01    OUR-FILE.  
      05    OUR-FILE-NAME   PIC X(8) VALUE "BIGDB".  
      05    OUR-GROUP      PIC X(8) VALUE "DATA".  
      05    OUR-ACCOUNT    PIC X(8) VALUE "ACCT".
```

Sample code:

```
PROCESS-USER-EXIT.
*
*   Ignore this file, it's not the one we want.
*
  IF EHI-FILE-NAME NOT = OUR-FILE THEN
    MOVE 3 TO ECA-ACTION
    GO EXIT-ROUTINE.
*
*   Is the data base we want to update open?
*
  IF ECA-FILE-NUMBER = 0 THEN
    PERFORM OPEN-THE-DATA-BASE
    IF DATA-BASE-NOT-OPEN THEN
      MOVE -3 TO ECA-ACTION
      GO EXIT-ROUTINE
    ELSE
      MOVE DATA-BASE-ID TO ECA-FILE-NUMBER
      MOVE "I"      TO ECA-FILE-TYPE.
*
*   We're only interested in the updates to the INVENTORY set
*
  IF EHI-DATA-SET NOT = USER-INVENTORY-SET THEN
    MOVE 1 TO ECA-ACTION
    GO EXIT-ROUTINE.
*
*   Look up the corresponding entry in the data base.
*
  PERFORM GET-THE-ENTRY.
*
*   No entry! Say we've lost sync.
*
  IF ENTRY-NOT-FOUND THEN
    MOVE -2 TO ECA-ACTION
    MOVE 1 TO ECA-STATUS
    GO EXIT-ROUTINE.

  COMPUTE INV-QTY = INV-QTY + (NEW-QTY - OLD-QTY).
  PERFORM UPDATE-THE-ENTRY.
  GO EXIT-ROUTINE.
```



**Note.** The database open routine would do the conversion of the dataset name **INVENTORY** to a dataset number, using **DBINFO** mode 201. This value would be saved in the **USERAREA** for subsequent use.

## Dataset Shadowing

Databases can be configured for shadowing so that only specific datasets are actually shadowed. This may be accomplished quite simply. However, be sure to read the “Dataset Shadowing Considerations” at the end of this section before doing the configuration. Configuring a database for dataset shadowing requires the use of both the **FILE** command and the **DSET** command. The **FILE** command is used to define the database and the **DSET** command is used to define the specific datasets that will either be included or excluded from shadowing.

When the **FILE** command is issued, it must be entered with either the **INCLUDE** or the **EXCLUDE** option specified. These options tell NetBase whether to include for shadowing only the datasets defined with the **DSET** command, or whether to exclude those datasets when shadowing the database. That is, the **INCLUDE** option may be used to most efficiently define just a few datasets to be shadowed. Conversely, the **EXCLUDE** option may be used to define the two or three datasets that do not need to be shadowed.

The **DSET** command is used to define the datasets that will be included for shadowing or excluded from shadowing. If the **DSET** command is used with the **RESET** option, the command will delete the dataset record from the directory. The **LISTF** command may be used to display the datasets defined for a database. If the **LISTF** command is entered with the **DSET** option specified, any datasets defined will also be displayed.

### Example

For example, to shadow only the **HEADERS** and the **DEVICES** datasets in the **BANNER** database, first enter the **FILE** command for the database on the master system:

```
D> FILE BANNER.DATA.NETBASE,1,3;INCLUDE
```

The **INCLUDE** option means that the datasets defined with the **DSET** option will be the only datasets in the database shadowed. Next, define the datasets to be shadowed:

```
D> DSET BANNER.DATA.NETBASE,HEADERS  
D> DSET BANNER.DATA.NETBASE,DEVICES
```

The datasets that are defined for a database can be listed as follows:

```
D> LISTF BANNER.DATA.NETBASE,DSET

```

-----LOCAL REFERENCE-----			-----REMOTE EQUIVALENT-----				
File	Group	Account	File	Group	Account	Node	
BANNER	DATA	NETBASE	->	--Same--		NODE-A	
				(Shadow)		SHADOW	
Datasets:	DEVICES		HEADERS				

```

O D F S P
P S L Y R L
E E A N O O
N T G C T G
- I - - - Y

```

Note that if the **FILE** definition for the IMAGE file is reentered without either the **INCLUDE** or the **EXCLUDE** option specified, the entire database will be shadowed. This has the effect of nullifying dataset shadowing for the database. Also, the **FILE** command may be reentered and the **INCLUDE/EXCLUDE** option changed. That is, if the database is defined with the **INCLUDE** option, it can be changed to **EXCLUDE** those datasets instead.

On the shadow system, the directory entries would look as follows. The shadow system does not need to know that the master system is only doing dataset shadowing.

```
D> FILE BANNER.DATA.NETBASE,1,3
D> SHADOW BANNER.DATA.NETBASE;PASS=CHANGE
```

## Dataset Shadowing Notes

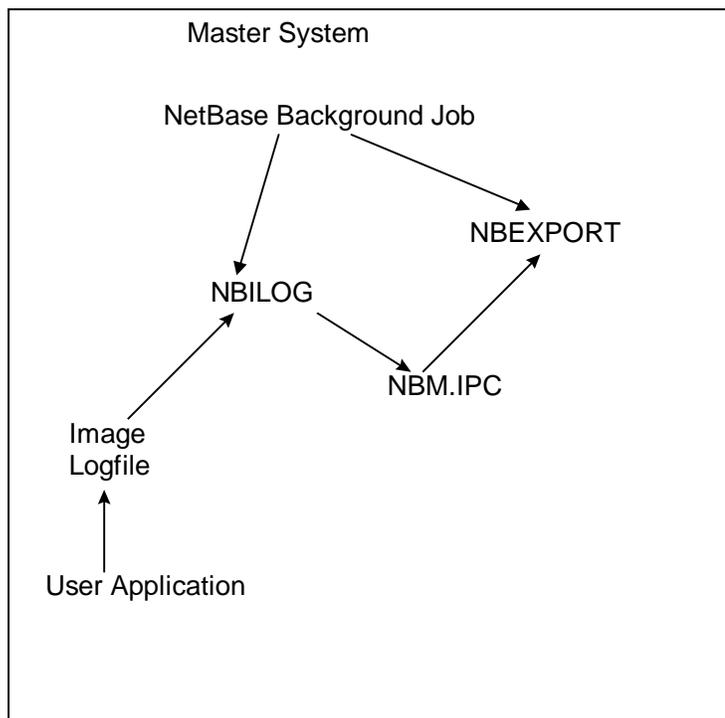
When shadowing only specific datasets, only the database root file and the datasets being shadowed need be restored to the shadow system. If any detail datasets are to be shadowed, the integrity of their associated master sets in the shadow copy must be protected:

- The manual master sets associated with shadowed detail sets must also be shadowed.
- The automatic master sets associated with shadowed detail sets must exist in the shadow copy, but do not have to be shadowed.

Datasets not being shadowed do not have to be restored to the shadow machine.

## Shadowing With TurboIMAGE Log Files

NetBase also provides the ability to shadow TurboIMAGE log files. The log files can be shadowed to another log file or they can be used to shadow changes directly into a shadow copy of the database. This method does not require the NetBase shell to be installed since it does not use the intercept facility. A separate process known as **NBILOG** is used to read the transactions from the log files and pass them to the shadow transport facility. The **NBILOG** process is controlled via **NBCTRL**. The configuration parameter **SHADOW - Use IMAGE Log Files** must be set to **Y** to enable this feature.



Use the following steps to configure a database to be shadowed using IMAGE logging. In this example, we are shadowing the **BANNER** database in **DATA.NETBASE**. The master copy resides on **SYSA** and the shadow copy resides on **SYSB**.

### Method 1: To shadow TurboIMAGE log files into a shadowed database:

1. Configure NetBase to use IMAGE logging. This is done by setting the **SHADOW - Use IMAGE Log Files** configuration parameter to **Y** on the master machine.
2. Add the database log file to NetBase's directory on the master machine:

---

```
:RUN NBDIR.PUB.NETBASE
```

```
NetBase Directory Program [x.x] (C) QUEST Software 1996
```

```
D> FILE BANLG001.DATA.NETBASE=BANNER.DATA.NETBASE,SYSA,SYSB;IMAGELOG
D> EXIT
```

3. Add the database to NetBase's directory on the shadow machine:

```
:RUN NBDIR.PUB.NETBASE
```

```
NetBase Directory Program [x.x] (C) QUEST Software 1996
```

```
D> FILE BANNER.DATA.NETBASE,SYSA,SYSB
D> SHADOW BANNER.DATA.NETBASE;PASS=dbpassword
D> EXIT
```

## Method 2: To shadow TurboIMAGE log files:

1. On the master node which is creating the log files, enter a file record for the log file into **NBDIR**. The **IMAGELOG** file option tells NetBase that the file is an IMAGE log file. If the **CHANGELOG** command is used for user logging, or if the **AUTO** parameter is specified for user logging, then the file name entered into the directory must end in **000**, for example:

```
FILE MYLOG000.GROUP.ACCT,1,2;IMAGELOG
```

This tells NetBase that we have a log file named **MYLOGnnn** in **GROUP.ACCT** residing on node 1 (the master node) and should be shadowed to node 2. NetBase changes the last three digits of the log file name automatically and assumes that the first log file will be number **001**.

2. On the shadow node (node 2 in our example), enter the same **FILE** command and also a **SHADOW** command as follows:

```
FILE MYLOG000.GROUP.ACCT,1,2;IMAGELOG
SHADOW MYLOG000.GROUP.ACCT
```



**Note.** You must build the first log file on the shadow nodes, the rest will be built automatically if they do not exist.

---

3. Verify that the first log file (**MYLOG001.GROUP.ACCT**) exists on the master node.

4. Make sure that IMAGE log file shadowing is enabled in the NetBase configuration. The **Use IMAGE Log Files** parameter is disabled by default. The next time you start NetBase, it will begin shadowing your log file or you can start the logging process with the **START ILOG** command in **NBCTRL**. You will receive messages at the console whenever a new log file is opened.

If you wish to map the log files into a different account or group, the re-mapping specification is made when entering the shadow record. For example, using the above example but mapping the shadowed file to the **BACKUP** account, perform the following procedure (Method 2a).

### Method 2a: To map the log files into a different account/group:

1. Follow step 1 as shown in Method 2 (above).
2. On the shadow node, enter the following **FILE** commands and also a **SHADOW** command for the files:

```
FILE MYLOG000.GROUP.ACCT,1,2;IMAGELOG  
FILE MYLOG000.GROUP.BACKUP,1,2;IMAGELOG  
SHADOW MYLOG000.GROUP.ACCT=MYLOG000.GROUP.BACKUP
```

Now all records coming from node 1 for the file **MYLOG000.GROUP.ACCT** are actually written to the file **MYLOG000.GROUP.BACKUP**.



**Note.** The first log file on the shadow system (**MYLOG001.GROUP.BACKUP**) must exist.

---

3. Follow steps 3 - 4 as shown in Method 2 (above).



**Note.** If you enter a **SHOW POST** command in **NBCTRL**, the file shows up under the transported name, not the actual name.

---

---

## Resetting the Logging Process

If you wish to reset your logging process back to log file number **001**, perform the following steps.

1. Stop the MPE logging process on the master system.  
**:LOG logid,STOP**
2. Shutdown the Netbase log reader process on the master system using the **STOP ILOG** command in **NBCTRL**.  
**:NBCTRL.PUB.NETBASE;INFO="STOP ILOG"**
3. Backup, purge or rename the old log files on the master system.
4. Build the new log file on the master system.
5. Use **:ALTLOG** to reset the log file name to **001**.
6. Reissue the **FILE** command in **NBDIR** on the master system to cause NetBase to reset to file **001**.  
**FILE MYLOG000,1,2;IMAGELOG**
7. Shutdown the posting process on the shadow system(s). This can be accomplished by issuing the **STOP POST** command in **NBCTRL**.  
**:NBCTRL.PUB.NETBASE;INFO="STOP POST"**
8. Backup, purge or rename the log files on the shadow system.
9. Restart the MPE logging process on the master system:  
**:LOG logid,START**
10. Restart the NetBase log reader process on the master:  
**:NBCTRL.PUB.NETBASE;INFO="START ILOG"**
11. Start the post process on the shadow system:  
**:NBCTRL.PUB.NETBASE;INFO="START POST"**

## Shadowing with Dynamic Rollback Recovery

---



**Note.** The following procedures must be followed if 1) you are shadowing a TurboIMAGE and not using TPS, and 2) if the application on the master system uses **DBXBEGIN**, **DBXEND** and **DBXUNDO**.

---

If the application on the master system uses the **DBXBEGIN**, **DBXEND** and **DBXUNDO** TurboIMAGE intrinsics, the **;DBX** option must be used to ensure the database stays in sync. The **;DBX** option is specified on the **SHADOW** statement on the shadow system. See the steps below for setting up shadowing with this option.

---



**Warning!** Do not use this parameter on databases that do not use dynamic rollback, since there is a performance hit on the shadowing side.

---

### To shadow with dynamic rollback recovery:

1. Your **SHADOW** statement on the shadow machine needs to have **DBX** added to it, for example:

```
SHADOW CHKDB.group.account;PASS=TALK;DBX
```

2. The **Enable Critical** flag must be set on the master machine. To check if this is set, do the following:

```
:RUN NBCTRL.PUB.NETBASE  
>SHOW SERVICES
```

If the following line is present, this flag is already set:

```
CRIT UPDATE: ENABLED
```

If the following line says:

```
CRIT UPDATE: DISABLED
```

then you need to issue the following command:

```
>ENABLE CRITICAL
```

3. Your **FILE** statement on the master machine needs to have **SYNC** added to it, for example:

```
D>FILE CHKDB.group.account,masternode,shadownode;SYNC
```

4. If you are setting up shadowing for the first time, perform a store/restore with the **OLDDATE** option to get the data from the master to the shadow machine. If you have already been shadowing the database, the store/restore step is not necessary.

To build the crossreference table on the shadow system, you need to perform the following step on the shadow machine:

```
:RUN NBSYNC.PUB.NETBASE,RESET
```

When the **DBX** parameter is specified for a database, **NBSYNC** will start the **DBSETUP** program to recreate the crossreference files. This takes around one minute per 400,000 sectors of detail sets. A very big database could take quite a while to resync.

## CHAPTER 6

# NetBase NFA

Network File Access (NFA) allows users access to files and databases regardless of where they physically exist in the network. In this chapter, you will find an explanation of how NetBase NFA accomplishes this using the NetBase components.

### In This Chapter

- **NFA Overview**
- **Configuring Network File Access**
- **Defining a File for NFA**
- **Using OMNIDEX and NetBase**
- **Monitoring NFA**
- **NFA Security**

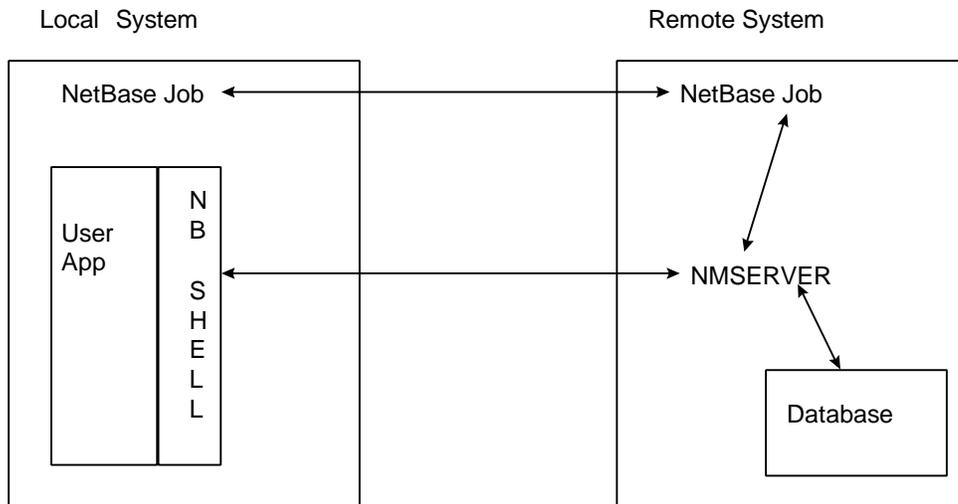
## NFA Overview

The basic operating principles of NetBase NFA are straightforward. All calls to the file and database systems are intercepted by the NetBase shell. Depending on the location of the file as specified in NetBase's directory, NetBase passes the calls either to MPE or to a server on the remote machine. The server process performs the call, and the results of the call are then returned to the user process on the local machine.

The NetBase shell is a set of procedures that intercept calls to the file system and to IMAGE. These procedures have the same names and parameter lists as their MPE counterparts. They are inserted into the load path of the application such that all calls to files or databases are resolved through the NetBase shell procedures. This gives NetBase the opportunity to preview all calls and determine whether to pass them to a remote machine.

If the call references a file or database that resides on a remote machine, NetBase passes the call to a server process on that machine. A server process is assigned to a user process the first time the user process opens a file or database on that machine. The server process continues to satisfy all file or database calls from that user to the remote machine until the user process terminates.

It is important to note that regardless of where the file resides, MPE always performs the file access. NetBase works in conjunction with the file system to enhance networking performance.



## Buffering

In order to enhance the performance of remote file access, NetBase reduces the number of network transactions by using buffering.

When an application performs a **DBFIND**, NetBase performs a **DBFIND** and a **DBGET** on the remote system, bringing back the data requested plus the data it anticipates the application to need shortly. NetBase returns the requested data to the application and stores the extra data locally. If the application subsequently needs the additional data, NetBase checks its local buffer first, returning the data in a fraction of the time required to remotely retrieve it. NetBase performs buffering for **DBFINDs**, and mode 2, 5, and 6 **DBGETs**. Buffering is an automatic feature of NetBase.

## Configuring Network File Access

The operation of NetBase, including its various features, is defined by NetBase's configuration parameters. Configuration maintenance is a subsystem of the NetBase control program, **NBCTRL**. The configuration parameters can be displayed and modified using the configuration subsystem. The system manager should become familiar with the control program, the configuration parameters, and the configuration maintenance commands in order to set up the Network File Access facility.

The parameters used for **NFA** can be listed in the configuration maintenance program as follows:

```
C> LIST NFA

NFA - Maximum Remote Users..... 64
NFA - Release Server on Last Close. N
NFA - Initial Server Allocation.... 0
NFA - Server Allocation Timeout.... 60
NFA - Server Initial Priority..... CS
NFA - Server LIB= load flag..... P
NFA - Lock Priority Boosting.....Y
NFA - Client Inactivity Timeout.... 0
NFA - Client Lock Held Timeout..... 0
NFA - Incoming users on startup.... Y
NFA - Outgoing users on startup.... Y
NFA - Client Access on startup..... N
```

The NFA parameters are set on the machine where the data is located and directly affect the behavior of the server processes created on the system. These server processes are created on behalf of remote processes seeking access to data on the machine. For example, the **Maximum Remote Users** parameter defines the maximum number of server processes that can be created to service remote processes accessing data.

For more information on the NFA parameters, turn to Chapter 3 “Using CONFIG”.

The **Server Initial Priority** parameter can be set to **SAME** to enable the server to run in the same processing queue as the process it is servicing. If the user on the local machine is in the interactive queue, then the server on the remote machine will be in the same queue. In order for this to happen, this configuration parameter must be set on the system where the server is running. By default, the server runs in the **CQ**.

## Defining a File for NFA

As with all files in NetBase, definitions for remote files are entered in NetBase's directory using the directory maintenance program, **NBDIR**. This is done on the local computer where the program runs so that NetBase knows where the remote files exist and how to access them.

This section details the steps required to define a file or database for network file access. If configuration changes are required, read the parameter documentation to determine whether processes must be stopped for the parameter change to take effect.

NetBase uses a directory to locate files in the network. The Directory Maintenance subsystem is used to define a file or database for network file access. To learn how to run the subsystem and to familiarize yourself with the **FILE** directory command, turn to Chapter 4 "Using NBDIR".

## Network File Access Options

The **FILE** directory command provides a number of options that affect the way remote files are accessed. Some of the options are specific to MPE files. Other options apply only to databases.

Option	Description
<b>ALT=node[,node]</b>	This option applies to all files. It specifies an alternate node to be used for remote access of the data if the first node is unavailable.
<b>INFO</b>	This option applies to IMAGE databases. It is only applicable for IMAGE files. It causes NetBase to look for and open a local copy of the root file to satisfy <b>DBINFO</b> calls. This option greatly reduces the networking overhead for programs like <b>QUERY</b> and 4th GL's that make extensive use of <b>DBINFO</b> .
<b>NOPEND</b>	This option applies to IMAGE databases. It disables pending locks. The default method of handling unconditional locks is for NetBase to send the <b>DBLOCK</b> and the next intrinsic across the network at the same time, instead of sending the <b>DBLOCK</b> and then sending the next intrinsic.
<b>READPASS=password</b>	This option applies to IMAGE databases. It defines a specific password for the remote server to use when opening the database for read access. The server will use this defined password instead of the password specified by the user process. The password is case sensitive.

*Continued*

Option	Description
<b>UPDPASS=</b> <i>password</i>	This option applies to IMAGE databases. It defines a specific password for the remote server to use when opening the database for update access. The server will use this defined password instead of the password specified by the user process. The password is case sensitive.

## NFA Examples

This section provides NFA syntax and examples.

### Syntax

```
FILE LOCAL FILE Reference=[remote file reference], node
```

### Example 1

In the first example, we take the simple case of using NetBase to allow local processes on **SYSA** to access the **BANNER** database on **SYSB**. To define the remote database, run **NBDIR.PUB.NETBASE** on **SYSA** and enter a **FILE** record for the database. The **FILE** record tells NetBase that **BANNER.DATA.NETBASE** is a remote file located on **SYSB**.

```
:RUN NBDIR.PUB.NETBASE
D> FILE BANNER.DATA.NETBASE,SYSB
```

### Example 2

If applications on the local machine that access this database make extensive use of **DBINFO** (**QUERY** and several other 4th GLs), the **FILE** record should be entered as shown. In addition, a copy of the database root file must exist locally. This **FILE** record tells NetBase that a copy of the database root file exists locally and will be used to satisfy calls to **DBINFO**. This root file must always match exactly on the two machines.

```
D> FILE BANNER.DATA.NETBASE,SYSB;INFO
```

### Example 3

NetBase does not require that the file on the remote machine have the same name as the file referenced by the local process. In the next example, we are going to allow local processes that reference **BANNER.DATA.NETBASE** to access **BANNER.DATA.BACKUP** on the remote system. The **FILE** record tells NetBase that **BANNER.DATA.NETBASE** is actually **BANNER.DATA.BACKUP** located on **SYSB**.

```
D> FILE BANNER.DATA.NETBASE=BANNER.DATA.BACKUP,SYSB
```

### Example 4

Entire groups and/or accounts can be defined as being located on a remote system by use of the **@** wildcard. The following illustrates defining the entire **DATA** group of the **PROD** account as being located on **SYSB**. This **FILE** record tells NetBase that the entire **DATA** group of the **PROD** account is located on **SYSB**.

```
D> FILE @.DATA.PROD,SYSB
```

### Example 5

If the primary node you specify is not available for remote access, an alternate node may be specified. If node **SYSB** is the primary node and the alternate node is **SYSC**, issue the following **FILE** command:

```
D> FILE TESTFILE.DATA.PROD,SYSB;ALT=SYSC
```

When the application opens **TESTFILE** and NetBase is unable to communicate with **SYSB**, it will try to open the file on **SYSC**.

## Using OMNIDEX and NetBase

The NetBase shell has an enhancement available for capturing and transporting OMNIDEX library calls. Using this enhancement to the shell usually results in lower network traffic and its associated overhead, for remote access of databases that utilize OMNIDEX keys. This is required if any programs will be accessing remote IMAGE databases that contain OMNIDEX indexes. With the OMNIDEX enhancement, OMNIDEX DBI calls are passed directly to the NetBase server on the remote system, where the calls are finally resolved through the OMNIDEX product.

For additional information on how OMNIDEX and NetBase work together, turn to Chapter 10 “NetBase and Third Party Products”.

## Monitoring NFA

Use the following **NBCTRL** commands for monitoring NFA. The NFA services should be configured to come up active with NetBase. For a description of each of these commands, turn to Chapter 2 “Using NBCTRL”.

### NBCTRL Commands

The following table lists the **NBCTRL** commands used to monitor NFA.

Command	Function
*> <b>SERVER</b>	Allows or aborts access to specified filesets.
*> <b>SHOW</b>	Displays NetBase's current local and remote user activity and status.
*> <b>START</b>	Starts NetBase or start a NetBase <b>NFA</b> service.
*> <b>STOP</b>	Stops NetBase or stop a NetBase <b>NFA</b> service when NetBase or the service is released by its current users.
*> <b>TRACE</b>	Displays NetBase server activity.
*> <b>ABORT</b>	Aborts a particular server.

### Examples

To display NetBase's current local and remote user activity and status:

```
SHOW LOCAL
      REMOTE
      SERVER
```

To stop/start NetBase or a NetBase **NFA** service:

```
STOP/START LOCAL [node]
              REMOTE [node]
```

To display NetBase server activity:

```
TRACE SERVER id
```

To abort a particular server:

```
ABORT SERVER id
```

To abort or allow a access to a specified fileset:

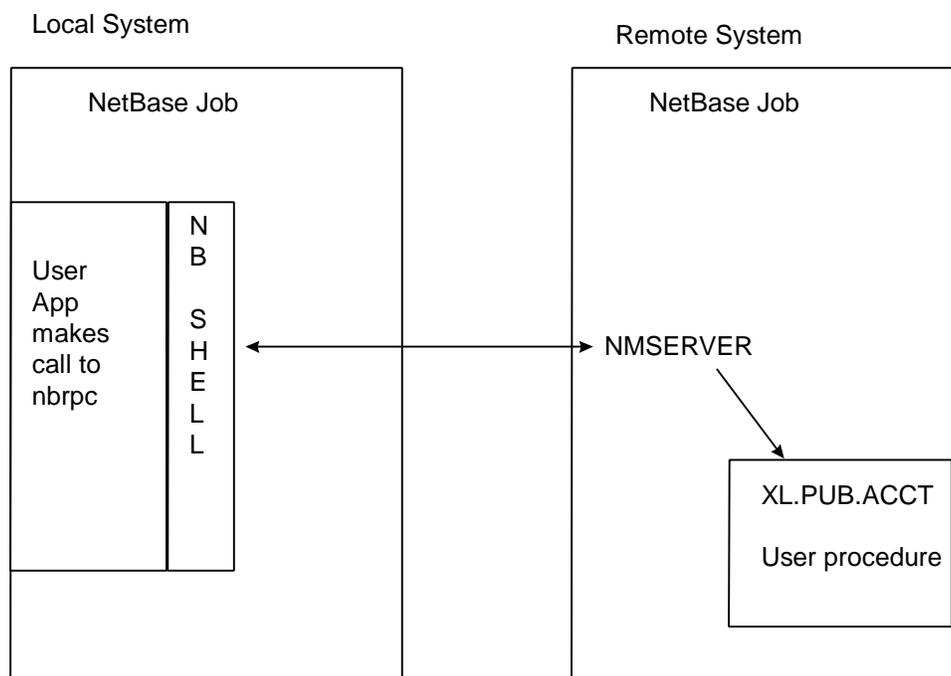
```
*> SERVER {ABORT=fileset[,fileset[,...]]}  
        {ALLOW=fileset[,fileset[,...]]}
```

## Remote Procedure Calls

The NetBase server process performs operations on a remote system on behalf of a local process. This typically involves satisfying file and IMAGE database calls for remotely located files. The server can also satisfy calls to a user written procedure located in a library on a remote system. This allows a local process to make calls to procedures located on remote systems.

When access to remote procedures is required, the local process calls a special subroutine. The procedure called by the local process is called **NBRPC**. The **NBRPC** procedure is included as part of the native mode NetBase shell.

The actual remote procedure name and the location of its library are passed to the server through the local call to **NBRPC**. The remote procedure will be dynamically loaded by the **NMSERVER** program when needed. When a process calls **NBRPC**, NetBase validates the node number and requests a remote server. The remote server then loads the remote procedure, and executes it.



See the following section for procedure calling conventions for **NBRPC** and for the remote procedure.

## NBRPC Procedure Calling Conventions

Use the following syntax and parameters for **NBRPC**.

### Syntax

**NBRPC** (*node, status, procname, xlname, sendbuff, sendlen, recvbuff, recvlen*)

### Parameters

Parameter	Description						
<b>node</b>	A 32-bit integer specifying the remote NetBase node number.						
<b>status</b>	<p>A 32-bit signed integer value passed by reference. The <b>status</b> parameter is read in two 16-bit fields:</p> <table border="0"> <thead> <tr> <th>Bits</th> <th>Value/Meaning</th> </tr> </thead> <tbody> <tr> <td><b>16:16</b></td> <td><b>status.subsys:</b> Identifies the subsystem which initiated the error or warning. The valid values are:  <b>0</b> = The error was returned by the user procedure.  <b>20034</b> = The error was returned by NetBase.  <b>104</b> = The error was returned by the load of the remote procedure.</td> </tr> <tr> <td><b>0:16</b></td> <td><b>status.info:</b> Identifies the type of error or warning.</td> </tr> </tbody> </table>	Bits	Value/Meaning	<b>16:16</b>	<b>status.subsys:</b> Identifies the subsystem which initiated the error or warning. The valid values are: <b>0</b> = The error was returned by the user procedure. <b>20034</b> = The error was returned by NetBase. <b>104</b> = The error was returned by the load of the remote procedure.	<b>0:16</b>	<b>status.info:</b> Identifies the type of error or warning.
Bits	Value/Meaning						
<b>16:16</b>	<b>status.subsys:</b> Identifies the subsystem which initiated the error or warning. The valid values are: <b>0</b> = The error was returned by the user procedure. <b>20034</b> = The error was returned by NetBase. <b>104</b> = The error was returned by the load of the remote procedure.						
<b>0:16</b>	<b>status.info:</b> Identifies the type of error or warning.						
<b>procname</b>	A 28-byte character array containing the remote procedure name, case sensitive. The first character in the array is used to delimit the name ( <b>%getopen%</b> ).						
<b>xlname</b>	A 28-byte character array containing the filename of the XL library where the remote procedure resides. The first character in the array is used to delimit the name ( <b>%XL.PUB.PROD%</b> ).						
<b>sendbuff</b>	A user-defined structure to be passed to the remote procedure.						
<b>sendlen</b>	A 32-bit signed integer by reference passing the length in bytes of <b>sendbuff</b> .						
<b>recvbuff</b>	A user-defined structure to be returned to the calling process.						

*Continued*

Parameter	Description
<i>recvlen</i>	A 32-bit signed integer by reference returning the actual length in bytes of the data to be transferred in <i>recvbuff</i> . It must be set by the calling process to the maximum receive buffer length. The procedure on the remote system may reset this variable to the actual receive length as long as it is less than or equal to the maximum length. NetBase will return the actual number of bytes returned.

## Remote User Procedure (procname) Calling Conventions

Use the following syntax and parameters for remote user procedure calling.

### Syntax

`procname (status,recvbuff,recvlen,sendbuff,sendlen)`

### Parameters

Parameter	Description
<i>status</i>	A 32-bit signed integer value passed by reference. The status variable should never exceed the length of a 16-bit integer. All NetBase errors will be returned negative leaving the positive range for errors from this procedure.
<i>recvbuff</i>	A user defined structure passed from the <b>NBRPC sendbuff</b> .
<i>recvlen</i>	A 32-bit signed integer by reference containing the length in bytes of the <b>NBRPC sendbuff</b> .
<i>sendbuff</i>	A user-defined structure to be returned to the calling process.
<i>sendlen</i>	A 32-bit signed integer by reference returning the actual length in bytes of the data to be transferred in <i>sendbuff</i> . This procedure may reset this variable to the actual send length as long as it is less than or equal to the maximum length. NetBase will return the actual number of bytes returned.

The following is sample code for **NBRPC**:

```
001000 IDENTIFICATION DIVISION.
001100*
001200 PROGRAM-ID. RPCCOB.
001300*****
001400* Sample main program for use with NBRPC          *
001500*****
001600 ENVIRONMENT DIVISION.
001700 CONFIGURATION SECTION.
001800 SOURCE-COMPUTER. HP-3000.
001900 OBJECT-COMPUTER. HP-3000.
002000*
002100 DATA DIVISION.
002200 WORKING-STORAGE SECTION.
002300*Variables for NBRPC
002400 01 NODENUM          PIC S9(9) COMP.
002500 01 STATUSVAR.
002600    05 STATSINFO      PIC S9(4) COMP.
002700    05 STATSSUB      PIC S9(4) COMP.
002800 01 PROCNAME        PIC X(28) VALUE SPACES.
002900 01 XLNAME          PIC X(28) VALUE SPACES.
003000 01 SENDBUFF      PIC X(80) VALUE SPACES.
003100 01 SENDLEN        PIC S9(9) COMP.
003200 01 RECVBUFF      PIC X(80) VALUE SPACES.
003300 01 RECVLEN        PIC S9(9) COMP.
003400*
003500 PROCEDURE DIVISION.
003600 MAIN-PROGRAM.
003700
003800* Initialize variables for NBRPC call
003900 MOVE 0 TO STATSINFO.
004000 MOVE 0 TO STATSSUB.
004100*
004200 MOVE 7 TO NODENUM.
004300*
004400 MOVE "%RPCTEST%" TO PROCNAME.
004500 MOVE "%XL.RPC.NETBASE%" TO XLNAME.
004600
004700 MOVE "TEST FOR RPC" TO SENDBUFF.
004800 MOVE 80 TO SENDLEN.
004900 MOVE 80 TO RECVLEN.
005000
005100*Call NetBase Remote Procedure
005200 CALL "NBRPC" USING NODENUM, STATUSVAR, PROCNAME, XLNAME,
005300          SENDBUFF,SENDLEN, RECVBUFF, RECVLEN.
005400
```

```
005500*Check status of call - if successful, display recv buffer.
005600 IF STATSINFO NOT= 0 THEN
005700     DISPLAY "NBRPC PROCEDURE FAILED"
005800 ELSE
005900     DISPLAY "RECVBUFF = ",RECVBUFF.
006000
006100 STOP-RUN.
:
```

```
001000$CONTROL SUBPROGRAM
001100 IDENTIFICATION DIVISION.
001200*
001300 PROGRAM-ID. rpctest.
001400*****
001500* Sample subroutine for use with NBRPC *
001600*****
001700 ENVIRONMENT DIVISION.
001800 CONFIGURATION SECTION.
001900 SOURCE-COMPUTER. HP-3000.
002000 OBJECT-COMPUTER. HP-3000.
002100*
002200 DATA DIVISION.
002300 WORKING-STORAGE SECTION.
002400 LINKAGE SECTION.
002500*
002600 01 STATUSVAR.
002700     05 STATSINFO          PIC S9(4) COMP.
002800     05 STATSSUB          PIC S9(4) COMP.
002900 01 SENDBUFF              PIC X(80).
003000 01 SENDLEN             PIC S9(9) COMP.
003100 01 RECVBUFF              PIC X(80).
003200 01 RECVLEN              PIC S9(9) COMP.
003300*
003400 PROCEDURE DIVISION USING STATUSVAR, SENDBUFF,SENDLEN,
003500                      RECVBUFF, RECVLEN.
003600 MAIN-PROGRAM.
003700
003710 CALL INTRINSIC "PRINT" USING SENDBUFF,80,0.
003800 MOVE "IN THE SUBROUTINE" TO RECVBUFF.
003900 MOVE 80 TO RECVLEN.
004000 MOVE 0 TO STATSINFO.
004100
004200 GO-BACK.
:
```

## NFA Security

When using NFA, by default all MPE file security is bypassed. If access to data needs to be restricted, **NBSECURE** must be used to define security. Security for files is defined on the system where the files reside.

**NBSECURE** is used for NetBase security maintenance. The security measures that NetBase has built into it for remote files provides the Network Administrator/Systems Manager a facility for controlling who has what type of access to files on each system. The **NBSECURE** program is the tool used to define NetBase's network file security. By using this program, groups of users can have access granted or restricted to any file or fileset.

## NBSECURE Commands

The following table lists the **NBSECURE** commands used for security maintenance.

Command	Description
<b>S&gt; ACCESS</b>	Sets the default <b>ACCESS</b> values.
<b>S&gt; ACCOUNT</b>	Sets the default <b>ACCOUNT</b> value.
<b>S&gt; ALLOW</b>	Adds/replaces security directory entries.
<b>S&gt; EXIT</b>	Quits <b>NBSECURE</b> .
<b>S&gt; GROUP</b>	Sets the default <b>GROUP</b> value.
<b>S&gt; HELP</b>	Displays a list of optional commands.
<b>S&gt; LISTSEC</b>	Lists the access given to remote users.
<b>S&gt; NODE</b>	Sets the default node.
<b>S&gt; REMOVE</b>	Removes files and/or the entire security record from the security directory.

## Running NBSECURE

**NBSECURE** can be invoked in two ways. To run from MPE, use the following command:

```
:NBSECURE.PUB.NETBASE
```

**NBSECURE** can also be entered via **NBCTRL** by entering the **SECURITY** command:

```
:NBCTRL.PUB.NETBASE  
*> SECURITY  
S>
```

Any user may run **NBSECURE**, however, certain commands do require that the user have various capabilities (**SM**, **NM**, or **OP**). Any command other than an inquiry will require special capabilities which are listed in the documentation for each command.

## Special NBSECURE Commands

Any MPE command that is executable in a break may be issued from **NBSECURE** without exiting. This is accomplished by prefixing the MPE command with a colon (:).

The **REDO** command allows the last command entered to be modified and re-executed. To repeat the previously typed command, simply type **REDO** at the **NBSECURE** prompt, modify the line using standard **EDITOR** commands, and enter an extra return to execute the command.

**NBSECURE** has a help facility. Type **HELP** to display a list of valid commands. If you type **HELP** followed by a valid command, this displays a brief description of the command's function along with the command syntax.

The following sections describe the **NBSECURE** commands.

## ACCESS

The **ACCESS** command sets the default file access mode list. The default access is originally set to **R,W,L,U** when the security subsystem is entered. Subsequent **ACCESS** commands change the default file access mode list. The default access is used by the **ALLOW** command if the access is not explicitly stated. This default is used for any commands referencing mode list. This command requires **SYSTEM MANAGER (SM)** capability.

### Syntax

```
S> ACCESS= modelist
```

### Parameters

Parameter	Description
<i>modelist</i>	<p>This parameter describes the default access modes that will be allowed on files entered through the <b>ALLOW</b> command. Valid access parameters are:</p> <p><b>R</b>      Read</p> <p><b>W</b>      Write</p> <p><b>L</b>      Lock</p> <p><b>U</b>      Update</p> <p>These parameters can be entered as follows. Commas can be used to separate.</p> <p><b>ACCESS=RWLU</b></p>

### Example

To set the default access parameters to read only access, do the following:

```
S> ACCESS=R
```

Based on the **ACCESS** list above, read access is assumed when the **ALLOW** command is issued. In the following example, all users on node one are allowed read access to all files in the **PAYABLES** account:

```
S> ALLOW @.@:1;@.@.PAYABLES
```

## ACCOUNT

The **ACCOUNT** command changes the current default account name. When the security subsystem is entered, the default account is the user's log on account. Subsequent **ACCOUNT** commands change the default account name. The default account set by **ACCOUNT** is used by the **ALLOW** and **REMOVE** commands when the account name is not explicitly stated.

### Syntax

```
S> ACCOUNT= {@  
             {accountname}}
```

### Parameters

Parameter	Description
<i>accountname</i>	The account name to be used as the default.

### Example

If you are currently logged into the **NETBASE** account, you can change the default to **PAYABLES** by entering:

```
S> ACCOUNT=PAYABLES
```

## ALLOW

This command adds records to NetBase's security directory, thus granting a remote user or a group of remote users the specified access to files. You can open up or restrict access to any set of files existing on the specified system. This command requires **SYSTEM MANAGER (SM)** capability.

The **ALLOW** command writes a security record to NetBase's security directory. Before NetBase's remote server process allows a file to be opened, it checks the security directory to see what kind of access is allowed for the specified user. If the access mode requested by the user is consistent with the type of access granted by the **ALLOW** command, access is permitted.

### Syntax

```
S> ALLOW {userset [ ;ACCESS=modelist];fileset[,fileset...]}
```

### Parameters

Parameter	Description
<b>userset</b>	<p>The list of users and/or user sets to which access is being specified. It must be in the one of the following forms. The <b>node</b> parameter can be a number or a name.</p> <p><b>user.account:node</b></p> <p><b>@.account:node</b></p> <p><b>@.@:node</b></p> <p><b>@.@:@</b></p>
<b>modelist</b>	<p>The list of types of access being granted to the specified file set. Parameters should be listed, separated by commas. Valid access parameters are:</p> <p><b>R</b>      Read</p> <p><b>W</b>      Write</p> <p><b>L</b>      Lock</p> <p><b>U</b>      Update</p>
<b>fileset</b>	<p>The list of files access is being granted. It must be in the following forms. Wildcards can be used in any or all of those parameters.</p> <p><b>file.group.account</b></p>

## Example

Access can be restricted for a specific user or group of users of another system while still allowing the remaining users of the system total access. To do this, first specify the access allowed by all users of a system:

```
S> ALLOW @.@:SYSTEMB;ACCESS=R,W,L,U;@.@.@
```

All users of **SYSTEMB** are now allowed unlimited access on our system. To verify what access has been granted, the **LISTSEC** command is used:

```
S> LISTSEC @.@:SYSTEMB
```

```
-----USERSETS-----  -----FILESETS ALLOWED-----
File      Group    Account  Node    Access
@.@:@     @        @        @       SYSTEMB (RWLU)
```

To restrict the users logged into the **PAYABLES** account on **SYSTEMB** to read access to the files in the **PAYABLES** account on our system:

```
S> ALLOW @.PAYABLES:SYSTEMB;ACCESS=R;@.@.PAYABLES
```

Any user on **SYSTEMB** who is signed on in the **PAYABLES** account can only read the files in the **PAYABLES** account on our system. All other users of the **SYSTEMB** system will be granted read, write, lock, and update capabilities on our system.

## GROUP

The **GROUP** command changes the current default group name. When the security subsystem is entered, the default group is the user's log on group. Subsequent **GROUP** commands change the default group name. The default group set by **GROUP** is used by the **ALLOW** and **REMOVE** commands when the group is not explicitly stated.

### Syntax

```
S> GROUP= {@          }
           {groupname}
```

### Parameters

Parameter	Description
<i>groupname</i>	The default group name.

### Example

If you are currently logged into the group files, you can change the default to **DATA** by entering:

```
S> GROUP=DATA
```

## LISTSEC

The **LISTSEC** command displays all file sets and access that has been granted to a particular user set. This command can also be used to show the type of access a user has to a specific file. The maximum number of entries that **LISTSEC** will display is **400**.

### Syntax

```
S> LISTSEC [userset]
```

### Parameters

Parameter	Description
<i>userset</i>	A valid user or user set. If not specified, this command will list all users or user sets. It must be in the one of the following forms. The <b>node</b> parameter can be a number or name.  <b>user.account:node</b>  <b>@.account:node</b>  <b>@.@:node</b>  <b>@.@@</b>

### Example

In the following example, **FILESETS ALLOWED** is the file, group, account and node to which access is granted to the userset. **Access** is the type of access granted to the userset for these files (**R**-read, **W**-write, **L**-lock, **U**-update).

```
S> LISTSEC MGR.NETBASE
```

```
-----USERSETS-----
MGR.NETBASE:CLIENT2
@.@@

-----FILESETS ALLOWED-----
File      Group    Account  Node    Access
@         PUB     FROG     @       (RL)
@         @       FROG     @       (RWLU)
@         @       @        DBSERVER (RWLU)
```

## NODE

This command sets the default node. The default node is either the first node on the list after the node you are on, or a node you have previously specified with the **NODE** command.

### Syntax

```
S> NODE={@ }
      {node}
```

### Parameters

Parameter	Description
<i>node</i>	The new default node. This may be expressed by name or number.

### Example

For example, assuming that you are on node **BUGS** as shown in the following configuration, **node 8 DAFFY** is used as the default node when you have not previously specified a default node with the **NODE** command. To see the list of nodes on your system:

```
:NBCTRL.PUB.NETBASE
*> CONFIG
C> L NODES
```

Node	Sys name	Line Type	Startup	Spool	Shadow
7*	BUGS	N	A	N	B
8	DAFFY	N	A	B	B
10	NORWAY	N	A	B	B

The default node is used by the **ALLOW** and **REMOVE** commands when the node is not explicitly stated.

```
S> NODE=10
```

or

```
S> NODE=NORWAY
```

## REMOVE

Removes files from NetBase's security records. The entire security record is deleted only if all files and filesets have been removed. This command requires **SYSTEM MANAGER (SM)** capability.

### Syntax

```
S> REMOVE userset [;fileset [,fileset...]]
```

### Parameters

Parameter	Description
<i>userset</i>	The userset that is to have files removed. It must be in the one of the following forms:  <b>user.account:node</b>  <b>@.account:node</b>  <b>@.@:node</b>  <b>@.:@</b>
<i>fileset</i>	The fileset for which access is to be removed.

### Example

Remove the entry that allows users from anywhere in the network access to all files on this system.

```
S> REMOVE @.:@;@.:@
```

## CHAPTER 7

# NetBase Spooling

NetBase Spooling makes sending your programs' output to any spooled printer anywhere in your HP 3000 network both simple and practical. It provides you with several transparent methods of defining the printing environment and performs the task with a minimal amount of overhead.

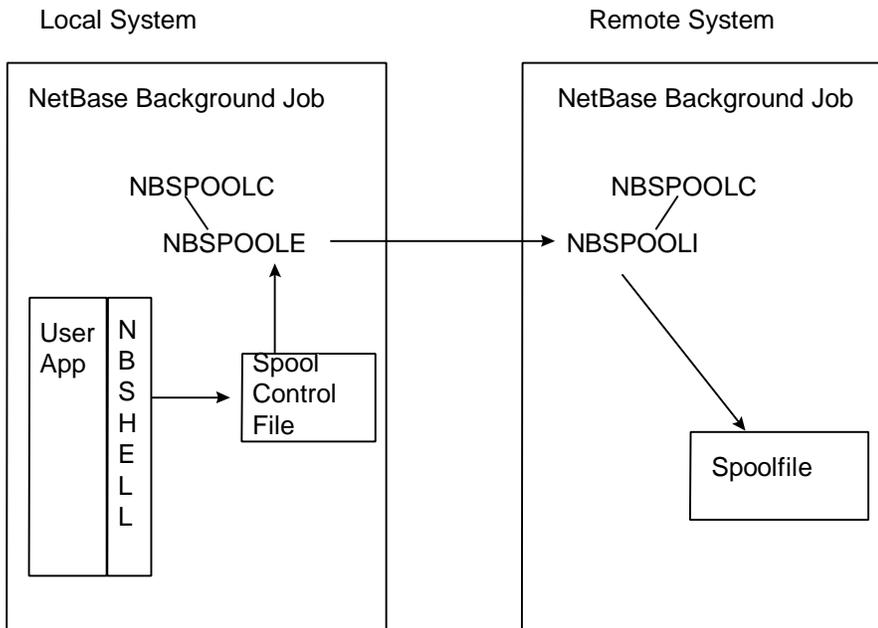
Rather than create the spool file locally and move it across the network to the destination printer after the spool file is created, NetBase spooling allows you to create the spool file on the destination system. NetBase spooling also transports multiple lines of output per network transaction so it is actually faster than MPE creating the spool file locally one line at a time. Creating a local spool file and then copying it across the network to a remote spooler doubles the amount of spooling overhead and holds up sending the report until it is complete.

## In This Chapter

- **Spooling Overview**
- **Remote Spooling**
- **Using Pseudo Devices with Remote Spooling**
- **NBSpool Program**

## Spooling Overview

Since NetBase spooling is positioned between the application and file system, it sends reports directly to the destination computer. This removes the need for a local copy and also eliminates unnecessary disk IOs. The result is a simpler, more efficient method of sharing remote spooled devices.



The above diagram shows a user application which begins to build a report destined for a specific device. The NetBase shell intercepts the creation of the spool file. The NetBase background job and its processes **NBSPOOLC** (spooling control) and **NBSPOOLE** (spooling export) transport the spool file to the remote HP3000. On the remote system, the NetBase background job and its processes - **NBSPOOLC** (spooling control) and **NBSPOOLI** (spooling import) receive the output and instruct MPE to create the spool file accordingly.

NetBase Spooling includes the utility NBSpool. NBSpool streamlines and automates the management of spool files generated by all applications, seamlessly distributing the printing function across networks and printer devices. NBSpool matches and surpasses the functionality of the MPE utility SPOOK, while using concise, simple commands that follow the SPOOK syntax.



**Note.** NetBase spooling cannot create the **\$STDLISTs** of a job on remote devices since the **\$STDLIST** is a byproduct of MPE running a job locally.

## Remote Spooling

Remote spooling is a feature of NetBase that allows you to run a program on system A, and create the resulting spool files (other than **\$STDLISTS**) on system B. **DEVICE** records define the **LDEV** numbers or device classes of remote printers. Spooled devices can be made available to the network using NetBase's directory. The directory is used to identify and locate devices within the network. This command requires **SM** and **NM** capability.

Specifying the location of a spooled device in the network is accomplished using a simple **DEVICE** command in **NBDIR**. The **DEVICE** command requires the device class or number (the name of the device as accessed by the application programs on the local system) and the device class or number and the node name or number of the computer where the device exists.

Use the following syntax for the **DEVICE** command:

```
D> DEVICE {class} [= {class}, node] [ ;ALT={class} [ , node] ] [ ;options ]
        {ldev} [ {ldev}          ] [          {ldev}          ]
```

Options for this command are:

```
COPY  [=device, outpri, copies]
TEMP
```

## Operating Principles

The basic operating principles of NetBase spooling are fairly straightforward. All opens to a spooled device are intercepted by the Shell. If a **DEVICE** directory entry exists for a local device, NetBase will set up the remote spooling. If there is no directory entry, NetBase passes the spooling request to MPE on the local system. Since the results of the call are immediately returned to the user process on the local machine, you do not have to wait for the remote spool file write.

Regardless of where the spooled device resides, MPE always performs the actual spooling process. NetBase works in conjunction with the MPE Spooler to enhance networking performance.

## Disaster Tolerance

You may want to create a local copy as well as a remote copy. If this is the case, NetBase spooling allows you to create either a permanent or temporary local copy. The **COPY** option directs NetBase to create a permanent local copy. If a network failure occurs while a process is outputting to a remote spooler that is defined with the **COPY** option, NetBase allows the process to continue using just the local spool copy. On the other hand, the **TEMP** option instructs NetBase to create the local copy as a temporary copy, and to keep it on the local system until NetBase is certain that the spool file was created successfully on the remote system. Either way, you are protected from network failures.

NetBase provides an additional feature for network disaster recovery: the ability to specify an alternate device. If the network link is not functioning between the local system and the primary destination, and you have specified an alternate destination, NetBase creates the spool file on the alternate system. With this feature, reports that keep the company in business will always print, if not at the printer of choice from the primary destination system, then from the second choice system.

NetBase Spooling relies on the network in order to create the spool files remotely. NetBase Spooling creates remote spool files while the program runs locally. If the network is down or if the NetBase job is not running on both systems when the local program runs, the spool files cannot be created on the remote system. Depending on the **NBDIR DEVICE** entry, several possible scenarios can result:

If the **NBDIR DEVICE** entry specifies an alternate node, NetBase will attempt to create the spool file on that system as the program runs locally.

```
D> DEVICE LP=LASERJET,HP995;ALT=LP,HP992
```

If the **NBDIR DEVICE** entry specifies a temporary local copy, NetBase will create a permanent one on the local system which can be moved to the remote system via an NBSpool background job, once the network is repaired.

The **TEMP** parameter is designed for this issue. When the network is up, NetBase creates a temporary copy on the local system as it creates a permanent one on the remote system. When the remote copy is completely created, the temporary version on the local system is purged. However, if the network fails during the creation or if the network is down during the creation (while the program runs), the local copy is kept.

```
D> DEVICE LP=LASERJET,HP995;COPY;TEMP
```

If the **NBDIR DEVICE** entry specifies a local copy, NetBase will create one locally, but none will be created remotely.

```
D> DEVICE LP=LASERJET,HP995;COPY
```

If the **NBDIR DEVICE** entry does not specify **COPY**, **TEMP COPY** or **ALT**, the report generating program will *fail* to create a spool file.

```
D> DEVICE LP=LASERJET,HP995
```

## Setting Up Remote Spooling

In order to use remote spooling, each system must be configured to import/export spool files. For more configuration information, turn to Chapter 3 “Using CONFIG”.

### To set up:

1. To verify that the nodes are configured:

```
: HELLO MGR.NETBASE
: RUN NBCTRL.PUB.NETBASE
*> CONFIG
C> LIST NODES
Node Sys name      Line Type      Startup      Spool      Shadow
  8*  DAFFY          N             A           N           B
  10  NORWAY         N             A           B           B
  12  DEVELOP        N             A           B           B
```

2. Add directory entries for remote devices via **NBDIR**. In our example, the local system **DAFFY** has a device class **REMLP** with spool files that need to be created on **NORWAY**.

```
:RUN NBDIR.PUB.NETBASE
D> DEVICE REMLP=LP,NORWAY;ALT=LP,DEVELOP;TEMP;COPY
D> EXIT
```

In the above entry, we are redirecting all spool files destined for device class **REMLP** to **LP** on the node named **Norway**. If the communication to **NORWAY** fails, the alternate device is **LP** on the remote system **DEVELOP**. Temporary copies of spool files are to be built locally and purged when the complete spool file has been built simultaneously on a remote device (the **TEMP** parameter).

3. Test the **DEVICE** entry. One way to do this is by using the **FCOPY** command as shown below. If the configuration and the device entry is correct, the spool file named **SPLTEST** will be created on **NORWAY** by the **FCOPY** command.

```
:FILE SPLTEST;DEV=REMLP
:FCOPY FROM=filename;to=*SPLTEST
```



**Note.** MPE's **PRINT** and **COPY** commands are not intercepted by NetBase, so they cannot be used to test remote spooling.

---

---

## Configuring Remote Spooling

Below are the additional **SPOOL** parameters associated with NetBase Spooling. The default values are usually sufficient. For a full description of each parameter, turn to Chapter 3 “Using CONFIG”.

```
C> l spool
```

```
SPOOL - Remote Transfer Timeout.... 120
SPOOL - Transfer Server Allocation. 4
SPOOL - Maximum Block (bytes)..... 8192
SPOOL - Maximum Files per Node..... 64
SPOOL - Use S' and J' on incoming.. Y
SPOOL - Change jsname to node name. N
SPOOL - Using SpoolMate..... N
SPOOL - Spooling Export Priority... DEFAULT
SPOOL - Spooling Import Priority... DEFAULT
SPOOL - Enable HP-UX Receiving..... N
SPOOL - Enable BOOTP daemon..... N
SPOOL - Enable LPD daemon..... N
SPOOL - Spooling on startup..... Y
```

## Using Pseudo Devices with Remote Spooling

In the previous example in the “Remote Spooling” section, we assumed that the local device referenced was an actual device class on the local system. With NetBase remote spooling, the local device or device class does not have to exist. Remote spooling supports the use of pseudo devices. For example, we want all the spool files created on a device with the user’s name to be created on the remote system:

```
:RUN NBDIR.PUB.NETBASE  
D> DEVICE username=LP,NORWAY
```

There is no device class configured on the local system with the user’s name, so we can use the following MPE syntax to implement a pseudo device:

```
:FILE MYREPORT;DEV=#username  
:FCOPY FROM=MYFILE;TO=*MYREPORT
```

The pound sign (#) in front of the device is used to “fake out” MPE. When MPE sees this sign, it assumes you are referencing a remote device and does not check to see if the device actually exists on the local system. This allows you to set the MPE file equation for your remote spool file. Next, since NetBase intercepts the open of the spool file, it checks to see if there is an entry in the directory for **username**. When it finds one, the spool file is automatically created on the remote system.

## A NetBase Spooling Implementation Example

Let us assume that we have four systems: **EENEY**, **MENEY**, **MINEY** and **MOE**. Our print engine is **MOE**. On our **EENEY**, **MENEY** and **MINEY** systems we would have the following directory entry to redirect spool files to **MOE**.

```
:NBDIR.PUB.NETBASE  
D> DEVICE LJ=LJ,MOE;COPY;TEMP
```

For each spool file destined for **LJ**, NetBase will create it (except **\$STDLISTS**) on **MOE**. A local copy will be created until its counterpart on **MOE** is complete. If the network fails during a spool file’s creation, the process continues and builds a copy on the local system.

Let us further assume that the local systems **EENEY**, **MENEY** or **MINEY** do not have any printers. In this case, we will need an NBSpool background job to move any **\$STDLISTs** created on the systems to **MOE**. Also, we need the NBSpool background job to pick up any spool files which were created locally while the network was down, and move them to **MOE** where they can print. (NBSpool is a spooling utility provided with NetBase spooling that allows you to manipulate spool files interactively or in a background job.) The following is an example of a background job that will complete the remote spooling implementation we have designed.

```
!JOB BACKGRND,MGR.NETBASE
!COMMENT
!COMMENT THIS JOB MOVES FILES TO THE LASERJET ON MOE
!COMMENT
!RUN NBSPOOL.PUB.NETBASE
XEQ; WAIT=60
ERROR IGNORE
MOVE DEV=LJ, TO MOE
END
EXIT
!EOJ
```

This background job checks every **60** seconds for spool files which were destined for **MOE** (by being created for the **LJ** device class) or are **\$STDLISTs** that were created locally. The background job moves any such spool files that it finds to **MOE**. If the network is down when the background job tries the move, it will try again a minute later. For more information on NBSpool background jobs, please see the *NBSpool Reference Manual*.

## NBSpool Program

NBSpool streamlines and automates the management of spool files generated by all applications, seamlessly distributing the printing function across networks and printer devices. Originally written to provide a method of moving spool files across a network, NBSpool quickly grew into a powerful, easy-to-use facility for performing a large variety of functions on spool files.

NBSpool matches and surpasses the functionality of the MPE utility SPOOK, while using concise, simple commands that follow the familiar SPOOK syntax. NBSpool makes complex spool management simple and intuitive, allowing end users to easily tap into corporate information resources without steep learning curves.



**Note.** The maximum number of spool files NBSpool can operate on is **16,380**. This applies to all NBSpool commands with the exception of the **STORE** command.

---

## NBSpool Features

This section provides information on NBSpool features. These features have been designed to allow you to easily and quickly interact with spool files. These features include:

- A flexible method of selecting subsets of spool files to alter, move, store, purge etc. Selections can be made on one or more of the following:

Spool file **DFID** or range of **DFIDs**

Job/session number or range of job/session numbers

Output priority or range of output priorities

Logical device number

Device class with wild cards

File names with wild cards

User names with wild cards

**\$STDLISTs** (include or exclude them)

Creation dates or date ranges

Spool file size

Spool files that contain a specified string or strings

Complex selections using **AND**, **OR**, and **NOT** operators

- LPR/LPD support which enables you to send and receive print jobs to and from remote hosts, whether they are on HP 9000s, Sun Sparc stations or any other UNIX environment.
- **BOOTP** support which can make your HP 3000 the **BOOTP** server for printers that require an IP address before they can be accessed by NBSpool. This eliminates the need for a UNIX system to perform the task.
- A **SHOW** command which displays a selective listing of spool files, including information not found on a **SHOWOUT** listing.
- A **SORT** command which allows sorts by any or all of the following spool file attributes:
  - Spool file name
  - Device number/class
  - User name
  - Creation date and time
  - Job/session number
  - Output priority
  - Spool file size
- A **ONCE=Y** subset which instructs NBSpool to **BANNER**, **ARCHIVE**, or **VSAVE** a file only once. The **ONCE=Y** subset causes NBSpool to operate on a spool file only once for any particular command.
- **JobRescue™** functionality, implemented as follows: A **STRING=filename** subset lets users put error strings in a file. NBSpool can **VSAVE** reports based on certain strings being in a **\$STDLIST**, such as abort messages. Also, a **FLAG={OK|WARN|ERROR| FATAL}** parameter was added to **VSAVE** which VISTA picks up and displays in the report selection screen.
- **VIEW** and **BROWSE** commands which provide a quick method to preview and optionally print or purge spool files.
- **ALTER** and **PURGE** commands which have been enhanced to operate on a subset of spool files.
- **TEXT**, **LIST**, and **KEEP** commands which provide a comprehensive method for searching, listing, and copying spool files to disk. The **TEXT** and **LIST** command can even be performed on opened spool files.
- A very fast duplication function to make exact duplicates of spool files.
- **MOVE** and **COPY** commands to quickly and efficiently move spool files across a network.

- A **TIMEOUT=** parameter for the **MOVE** and **COPY** commands which allows you to override the default time-out values.
- The **SPOOLACKFREQ** JCW available with the **MOVE** and **COPY** commands controls how many blocks of spool file data are sent before waiting for a response from a remote computer. The larger the value, the faster the transfer.
- A **STORE** and **RESTORE** capability for off-line storage. These commands are faster than **SPOOK** and use less tape. Also, multiple **STORES** can be appended to the same tape.
- A tape directory function so you can see what is on a tape, including **SPOOK** tapes.
- An **ARCHIVE** and **UNARCHIVE** command to store and optionally compress a set of spool files to disk. Also, an **ARCDIR** command to see what is in an archived file.
- A **PRINT** command for printing spooled output to a printer slaved to a terminal or PC.
- Full support of job **\$STDIN** files, that is, jobs that have been streamed but are not executing, including listing, archiving and restreaming.
- Works on both MPE/V and MPE/XL.
- NBSpool supports the 132-column capability of the newer Hewlett-Packard terminals.
- Non-spoiled devices are supported. Users can print to locally attached printers in landscape, compressed portrait, or any desired font. You can easily customize the printer definition for any printer.
- Supports **ASSOCIATED** devices.
- Can run in the background to perform repetitive tasks, such as archiving or moving spool files.
- Supports command files.
- Adds, changes or deletes environment file data.
- Creates an audit trail of activity.
- Friendly user interface gives users the tools they need to easily view their reports on-line.
- Add on modules for Novell NetWare, IBM Hosts, and UNIX systems.

## CHAPTER 8

# AutoRPM

AutoRPM is NetBase's interface to Hewlett-Packard's Remote Process Management (RPM). Remote Process Management is a feature of HP's Network Services. It provides the ability to run an interactive program on a remote computer.

### In This Chapter

- **AutoRPM Overview**
- **Defining a Program for AutoRPM**
- **Notes on Implementing AutoRPM**
- **AutoRPM Options**

## AutoRPM Overview

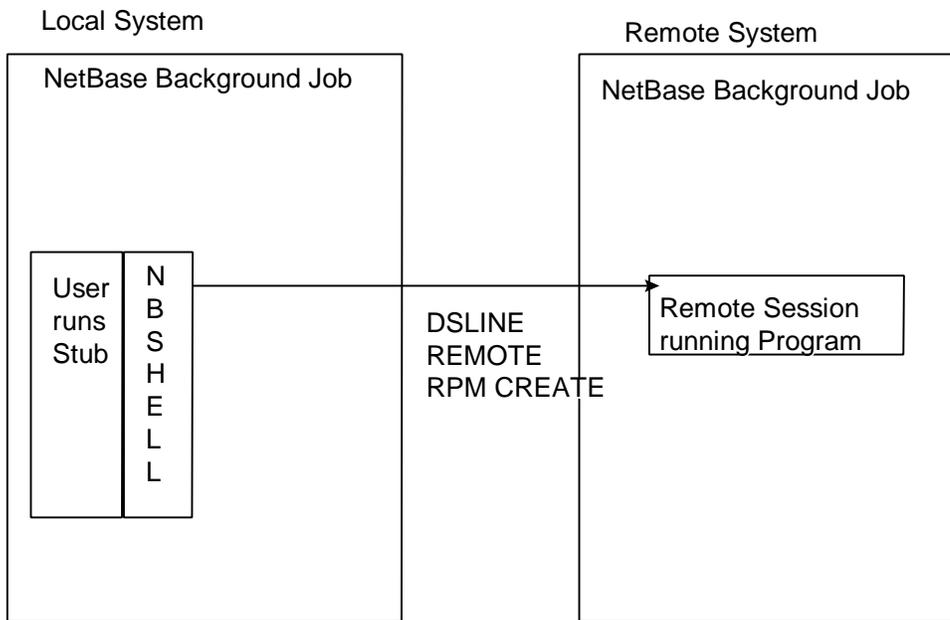
To run a program on a remote machine using HP's RPM, the user must perform a series of steps. First, the **DSL**INE command is issued to open a line to the remote computer. Then, the user must issue a **REMOTE LOGON**, and then finally a **REMOTE RUN** to run the program.

NetBase makes all this transparent to the user. By entering some information in NetBase's directory, the user can run the program, and NetBase will automatically run the program on the remote computer. Further, NetBase will transport the session and program file equations over to the remote machine so they take effect in the actual running environment. NetBase will also save the system JCW when the program terminates and transfer it back to the user's machine.



**Note.** AutoRPM requires **NS** services.

---



The above diagram show a user application which begins to run a local program. In that local program's place is a copy of our stub program, **NBNULL.LIB.NETBASE** that does an **FOPEN** call which the NetBase shell intercepts. After checking the directory for this program and determining on which system the program should be run, the NetBase shell issues the **DSL**INE, **REMOTE LOGON** and **REMOTE RUN** for the user.

## Defining a Program for AutoRPM

As with other services in NetBase, definitions for remote programs are entered in NetBase's directory using the directory maintenance program **NBDIR**. On the local computer, NetBase needs to know where the remote program exists and how to run it. It also needs to know how to logon to the remote system on behalf of the local user in order to create the remote process. Therefore, you must enter **PROGRAM** and **LOGON** entries in **NBDIR**.

### AutoRPM Example

Setting up a program for AutoRPM requires two **NBDIR** entries. A **LOGON** record tells NetBase that the local user is defined to use an AutoRPM program and provides the information needed to create a remote session for that user. A **PROGRAM** record maps a local program name to the name of a program on a remote node that should be run instead.

Setting up AutoRPM also requires the use of a “stub” program. The NetBase shell does not intercept MPE commands so when a user issues a **RUN** command, NetBase does not see it. In order for NetBase to intercept the **RUN** command, a special program has been created called **NBNUL**. This program will cause NetBase to do a lookup in its directory to find out where to actually run the remote program.

#### LOGON Syntax

```
D> LOGON localuser[:node]=logonstring
```

#### PROGRAM Syntax

```
D> PROGRAM progname [=remoteprog][,nodeid][;LIB= ]
                                     [;PARM= ]
                                     [;NOEQ ]
                                     [;NOJCW]
                                     [;XL=  ]
```

For example, to setup **HPMAIL** to run on a remote machine:

1. Run **NBDIR** on the local machine. Enter a **PROGRAM** record for **HPMAIL** specifying the node where the remote program resides (on **SYSB**):

```
PROGRAM HPMAIL.HPMAIL.SYS,SYSB;LIB=G;PARM=1
```

2. Enter a **LOGON** record for the users that will be running **HPMAIL**:

```
LOGON @.:@:SYSB=REMOTEU,MGR.MAIL,PUB
```

3. Copy the stub program **NBNUL.LIB.NETBASE** to the same file name as the local program:

```
:FCOPY FROM=NBNUL.LIB.NETBASE;TO=HPMAIL.HPMAIL.SYS;NEW
```

Now, any user running **HPMAIL** on the local machine will be running **HPMAIL** on node **SYSB** automatically.

## Notes on Implementing AutoRPM

In some cases, you may want users requesting one program locally to actually run another program remotely. NetBase allows you to do this via the renaming option in the **PROGRAM** entry. For example, if your users try to run **EDITOR.PUB.PROD** locally, and you want them to run **QUAD.LIB.NETBASE** on a remote node, you would enter the following **PROGRAM** entry:

```
D> PROGRAM EDITOR.PUB.PROD=QUAD.LIB.NETBASE,REMNODE
```

When issuing the **LOGON** entry in **NBDIR**, you should verify that the remote logon is valid. In the following example, the account **MAIL**, the user **MGR.MAIL**, and the group **PUB.MAIL** must exist on **SYSB**. Since no passwords are specified, none should be expected by the system (**SYSB**). Finally, third party security packages must not be in place to intercept a logon for **MGR.MAIL**.

```
D> LOGON @.ACCT:SYSB=REMOTEU,MGR.MAIL,PUB
```

If you do not want to send file equations over to the remote node where the program will run, you should specify the **NOEQ** option on the **PROGRAM** entry. For the same reason, if you do not need JCWs transported back to the local node, you should specify **NOJCW** on the **PROGRAM** entry. By default, all file equations are sent to the remote system and the system JCW is returned.

```
D> PROGRAM CALCR.PUB.PROD,REMNODE;NOEQ;NOJCW
```

Normally, when an AutoRPM process ends, the remote session ends. When the remote program is started via a stub program like **NBNUL**, the session will always terminate when the user exits the remote program. When the program is terminated, so is the environment.

If the remote process is created by intercepting a **CREATEPROCESS** call, the **DS** environment is created in the context of the program calling **CREATEPROCESS** (a local menu program, for example). Since that program (the menu driver) does not terminate when the user exits the remote program, the remote session will remain unless the **;SHUT** option is invoked.

```
D> PROGRAM CALCR.PUB.PROD,REMNODE;SHUT
```

- NetBase disables the break key during AutoRPM so that users cannot get to the MPE colon prompt on the remote system.
- NetBase provides the option of specifying an X2 list for the program. If you specify **;XL=MYLIST** on the program entry, you must enter an **XL** entry in **NBDIR**:

```
XL MYLIST=XL.PUB.PROD
PROGRAM CALCR.PUB.PROD,REMNODE;XL=MYLIST
```

## AutoRPM Options

The **PROGRAM** directory command provides a number of options that affect the way remote programs are run:

Option	Description
<b>LIB=</b>	The load library to be used when the process is created on the remote node. A value of <b>D</b> prevents default <b>LIB=P</b> .
<b>PARM=</b>	The parameter string to be used in the run command for the program on the remote node.
<b>XL=</b>	The name of a user defined <b>XL</b> list entered in the directory with the <b>XL</b> command. The <b>XL</b> list is used in the run command for the program on the remote node.
<b>NOEQ</b>	This option disables sending the file equates for the local process to the remote node for the remote process.
<b>NOJCW</b>	If this option is selected, NetBase will not transmit the JCW from the remote process to the local process when the remote process terminates.
<b>TRACE</b>	Tells NetBase to write the intrinsics used by the program to the NetBase trace files.
<b>;SHUT</b>	Closes the remote line when the remote program terminates.

## CHAPTER 9

# NetBase Statistics

NetBase Statistics provides a method for capturing file access and timing information, and process overhead measurements. NetBase also provides reports that clearly illustrate the captured information.

Since NetBase is positioned between the application program and the file system, collecting statistical data regarding file access is a byproduct of running applications through NetBase. The collection of statistics can be enabled or disabled at any time for any subset of data required.

NetBase has been designed to minimize the overhead incurred when statistics are enabled by making use of an extra data segment during the collection phase. The collected information is logged to disk only when the file is closed or the process terminates. Using these techniques, NetBase is able to capture accurate information without incurring additional overhead that would distort the element being measured.

### In This Chapter

- **Collecting Statistics**
- **Reporting Statistics**
- **Generating Scenarios**

## Collecting Statistics

Starting, stopping, and inquiring about the statistics process is accomplished using **NBCTRL**. With statistics enabled, NetBase's shell procedures accumulate I/O intrinsic counts and timings in an extra data segment assigned to each user process. These timings are posted to disk as the files are closed or the process terminates. The actual posting to the statistics log file is performed by NetBase's statistics logging process (**NBSTATLG**) running under **NBCOP**.

Statistic log files are created by **NBSTATLG** as they are required. The file names assigned to the log files are in the format **NBLGnnnn.DATA.NETBASE**. The data format of these files is detailed at the end of this chapter. **NBSTATLG** continues posting to each log file until the file fills up or a **STATS SWITCH** command is issued. When either of these conditions occur, **NBSTATLG** closes the current file and builds and opens the next. The log file limit is specified during configuration.

Three types of statistics records are created: IMAGE files, MPE files and Process records. A subset of these records are required by each report program. To enable collection of these records the **STATS START** command is issued from within **NBCTRL**. Specifying the **ALL**, **LOCAL**, **REMOTE**, **IMAGE** or **FILES** parameter causes the respective records to be collected.

```
:RUN NBCTRL.PUB.NETBASE
```

```
NetBase Control Program [v.u.f] Update n (C) QUEST Software 1987
```

```
*> STATS START=ALL
```

```
*>
```

The **STATS SHOW** command displays the status of the **STATISTICS** process, the current log file and the selected statistic parameters.

```
*> STATS SHOW
```

```
LOGGING STATUS:      ACTIVE
  CURRENT FILE:      NBLG0004
  RECORDS LOGGED:    1534
  PERCENT FULL:      47%

LOGGING OPTIONS:
  IMAGE              LOCAL/REMOTE
  FILES              LOCAL/REMOTE
  PROCESS            LOCAL
```

The **STATS STOP** command prevents any new processes from collecting statistics. Processes currently running will continue to have statistics collected. Specifying one of the collection parameters can be used to stop collecting a specific type of statistics.

```
*> STATS STOP  
*>
```

It should be noted that issuing a **STATS STOP** command does not stop **NBSTATLG**, nor does it close the current log file. Once **NBSTATLG** is active, it can only be killed by stopping or aborting NetBase. It must remain alive to post statistics records for any user process that were active when the **STATS STOP** was issued.

To free up the current log file for reporting, the **STATS SWITCH** will close the current log file and build the next.

```
*> STATS SWITCH
```

## Reporting Statistics

The standard NetBase reports provide the technical staff with a great deal of useful information. They aid in network design/management, network and application tuning, and they can be used to monitor computer and user response time. The following is a brief description of these reports:

Report Name	Description
<b>NBLGDUMP</b>	<b>NetBase Log Listing Utility.</b> This program formats individual statistics log records. Selection by program file and/or data file is provided. Summarization to the program/file level can be selected.
<b>NBRPT01</b>	<b>Network Statistical Analysis.</b> This report is designed to display file I/O traffic. It illustrates the source and volume of activity to each data file by all programs. The report displays all files each program accesses or all programs that access each file.
<b>NBRPT02</b>	<b>Program Activity Summary.</b> This report is designed to display file access as a percentage of total process CPU. Also included on this report is lock impede time. The report is useful in determining the benefit of off-loading CPU utilization to other computers and evaluating locking strategies.

### Running NBLGDUMP

**NBLGDUMP** can be run either from a job or a session. The program normally writes to the **\$STDLIST** device but the output can be redirected using a file equation to the file **NBLGDUMP**.

```
:FILE NBLGDUMP;DEV=LP,13
:RUN NBLGDUMP.PUB.NETBASE
```

```
NetBase Log Listing Utility [v.u.f] Update n QUEST Software 1987
```

```
Enter Log file Name: NBLG0001.DATA.NETBASE
```

```
Selection Criteria - <cr> for all
```

```
Enter PROGRAM name:
```

```
Enter FILE name:
```

```
Enter record types: (A/F/I/P)
```

```
Summarize data (Y/N):
```

## Program Input

The following table provides descriptions of the program input.

Input	Description
Enter Log file name	The file name of a valid NetBase Statistics log file. No default is allowed. It cannot be the current log file. Do a <b>STATS SWITCH</b> to run against current log file.
Enter PROGRAM name	The fully-qualified file name of the desired program subset. The default is all programs.
Enter FILE name	The fully-qualified file name of the desired data file or database subset. The default is all files.
Enter record type	Valid options are: <b>A</b> All record types <b>F</b> MPE file types <b>I</b> IMAGE file type <b>P</b> Process type
Summarize data (Y/N)	Answering <b>Y</b> to this prompt causes <b>NBLGDUMP</b> to summarize all records with like program and file names into one record before reporting. The default is no summarization.

## Program Output

**NBLGDUMP** formats three different record types: IMAGE, MPE and KSAM files. The following is an example of the output generated for MPE and KSAM files.

```

*****
USER:      #J326 TSPROC,JOB.CEI (TEST)
PROGRAM:   TSPROC.PUB.CEI
FILE:      TSPRH.DATA.CEI (TEST)
OPENED:    THU, JUL 16, 1996,  3:58 AM
CLOSED:    THU, JUL 16, 1996,  5:16 AM

          Count      -TOTAL (msec)-      -AVG (msec)-      - % of TOTAL-
          Count      Wall      CPU      Wall      CPU      Cnt Wall  CPU
FOPEN          1          62       22          62       22          0   0   0
FCHECK       1318          583      446           0         0         11   1   2
FCLOSE         1          10         8          10         8          0   0   0
FCONTROL     2635       56627     15470         21         6         22  83  64
FGETINFO         1           2         2           2         2          0   0   0
FPOINT       3953       2705     1855           1         0         33   4   8
FREAD        1318          712      655           1         0         11   1   3
FREADDIR     1317       1336     1175           1         1         11   2   5
FWRITE        1317       5825     4689           4         4         11   9  19
*****

```

The heading information is straightforward. The node that the program was run on is listed after the **USER:** information. The node that the file lives on is listed after the **FILE:** information. If the program node does not match the file node, the file is designated as a **REMOTE FILE**.

One detail line is given for each intrinsic called by the process for each file. Average timings are computed for each intrinsic. The percentages refer to the relative percentages of access broken down by intrinsic. In the example, 22% of the intrinsic calls were to **FCONTROL**, but **FCONTROL** was responsible for 83% of the total elapsed time and 64% of the CPU time spent accessing this file.

This example is of the output generated for IMAGE databases:

```

*****
USER:      #J326 TSPROC,JOB.CEI (TEST)
PROGRAM:   TSPROC.PUB.CEI
FILE:      IDEAS.DATA.CEI (TEST)
OPENED:    THU, JUL 16, 1987,  3:58 AM
CLOSED:    THU, JUL 16, 1987,  5:16 AM

          Count          -TOTAL (msec)-          -AVG (msec)-          - % of TOTAL-
          Count      Wall      CPU      Wall      CPU      Cnt Wall  CPU
DBOPEN           1         527      186      527      186         0   0   0
DBCLOSE          1317        1323      904         1         1         3   0   0
DBFIND          12933       336702    69006         26         5        25  40  32
DBGET           14705       409649    83695         28         6        29  48  38
DBLOCK           7330        11883     8053          2         1        14   1   4
DBPUT             8         1679      448        210        56         0   0   0
DBUNLOCK         7330        6895     5158          1         1        14   1   2
DBUPDATE         7376        83589    51383         11         7        14  10  23
*****

```

This display is the same as the MPE file display. The only difference is that the IMAGE intrinsics are displayed.

## Running NBRPT01

**NBRPT01** is a network statistical analysis. This report was designed to aid in the determination of program and data file location inside the network. It clearly shows the source and volume of all file related activity by programs running through NetBase. **NBRPT01** can be run either from a job or a session. The program normally outputs to the spool file **NBREPORT**, but file equations are allowed.



**Note.** A current log file *cannot* be specified. To run against a current log file, you can do a **STATS SWITCH**.

```
:RUN NBRPT01.PUB.NETBASE
```

```
NetBase Statistics Reporting [v.u.f] Update 0 Copyright QUEST Software 1991
```

```
Reduction Options:
```

- 1) Extract from log files
- 2) Use existing reduction

```
Enter option: 1
```

```
Enter first Stat log #: 1
```

```
Enter last Stat log #: 2
```

```
Enter file name to save reduction: PASS1
```

```
Report options:
```

- 1) File Access by program
- 2) Program Access by File

```
Enter option: 1
```

```
Print User Information: N
```

```
*****
```

```
REDUCTION SORT STATISTICS:
```

```
INPUT RECORDS:          500
OUTPUT RECORDS:         35
WALL TIME:              5.059(SEC)
CPU TIME:               2.120(SEC)
-----
```

```
REDUCTION SORT STATISTICS:
```

```
INPUT RECORDS:          35
OUTPUT RECORDS:         34
WALL TIME:              2.577(SEC)
CPU TIME:               .685(SEC)
-----
```

```
REPORT SORT STATISTICS:
```

```
INPUT RECORDS:          34
OUTPUT RECORDS:         34
WALL TIME:              2.558(SEC)
CPU TIME:               .648(SEC)
*****
```

## Program Input

The following table provides descriptions of the **NRPT01** reduction options.

Reduction Option	Description
<b>Enter Option</b>	The number corresponding to the desired reduction option. <b>1</b> reads through statistic log files to compile the data. <b>2</b> uses a reduction file created from a previous run of <b>NRPT01</b> or <b>NRPT02</b> .
<b>Enter first Stat log #</b>	The number of the first log file to be extracted. Applies only when option <b>1</b> is selected.
<b>Enter last Stat log #</b>	The number of the last log file to be extracted. Applies only when option <b>1</b> is selected.
<b>Enter file name to save reduction</b>	If other reports are to be run against the same data, the reduction can be reused. If a file name is entered here, the reduction data will be saved under this file name. Applies only when option <b>1</b> is selected.
<b>Enter the file name of the reduction to use</b>	The name of a reduction file created by a previous run of <b>NRPT01</b> or <b>NRPT02</b> . Applies only when option <b>2</b> is selected.

The following table provides descriptions of the **NRPT01** report options.

Report Option	Description
<b>Enter Option</b>	<p>Entering <b>1 (File access by program)</b> causes the primary sort key to be program name and all files accessed by that program to be listed there under.</p> <p>Entering <b>2 (Program access by file)</b> causes the primary sort key to be file name and all programs accessing each file to be listed thereunder.</p>
<b>Print User Information</b>	Entering <b>N</b> summarizes down to the program or file level. Entering a <b>Y</b> summarizes down to the <b>LDEV</b> , user, program or file level.

## Program Output

The following table provides definitions of the terms used in the **NBRPT01** report.

Term	Description
<b>OPENS</b>	The total number of times that this file was opened by this program.
<b>READS</b>	<p><b>Count</b> %4 The total number of read type intrinsics issued against this file by this program. Read type intrinsics are considered to be <b>DBFIND</b>, <b>DBGET</b>, <b>FREAD</b>, <b>FREADBYKEY</b>, <b>FREADLABEL</b>, <b>FREADSEEK</b>, <b>FREADC</b>, <b>FFINDBYKEY</b> and <b>FFINDN</b>.</p> <p><b>Avg Wall</b> — The average amount of elapsed time in msec that it took for each read type activity. This is computed:</p> <p style="text-align: center;"><b>Total Wall time/Count</b></p> <p><b>Avg CPU</b> — The average amount of CPU time in msec that it took for each read type activity. This is computed:</p> <p style="text-align: center;"><b>Total CPU time/Count</b></p>
<b>UPDATES</b>	<p><b>Count</b> — The total number of update type intrinsics issued against this file by this program. Update type intrinsics are considered to be <b>DBPUT</b>, <b>DBUPDATE</b>, <b>DBDELETE</b>, <b>FWRITE</b>, <b>FWRITEDIR</b>, <b>FDELETE</b>, <b>FUPDATE</b> and <b>FWRITELABEL</b>.</p> <p><b>Avg Wall</b> — The same as read, except using update type intrinsics.</p> <p><b>Avg CPU</b> — The same as read, except using update type intrinsics.</p>
<b>OTHER</b>	<p><b>Count</b> — The total number of non-read or update type intrinsics issued against this file by this program. These intrinsics are considered to be all other file intrinsics not listed above.</p> <p><b>Avg Wall</b> — The same as read, except using non-read or update type intrinsics.</p> <p><b>Avg CPU</b> — The same as read, except using non-read or update type intrinsics.</p>
<b>OVERALL</b>	<p><b>Count</b> — The total of the previous three categories.</p> <p><b>Avg Wall</b> — The average of the previous three columns.</p> <p><b>Avg CPU</b> — The average of the previous three columns.</p>

## Running NBRPT02

**NBRPT02** is a program activity summary. This report was designed to identify the amount of processing a program devotes to accessing each file. This quantity is expressed as a percentage of total CPU used. **NBRPT02** can be run either from a job or a session. The program normally outputs to the spool file **NBREPORT** but file equations are allowed.

:RUN NBRPT02.PUB.NETBASE

NetBase Statistics Reporting [v.u.f] Update n Copyright QUEST Software  
1991

Reduction Options:

- 1) Extract from log files
- 2) Use existing reduction

Enter option: 2

Enter file name of reduction to use: PASS1

Print User Information: N

\*\*\*\*\*

REDUCTION SORT STATISTICS:

INPUT RECORDS:           35  
OUTPUT RECORDS:         34  
WALL TIME:               2.831(SEC)  
CPU TIME:                .669(SEC)

-----

REPORT SORT STATISTICS:

INPUT RECORDS:           34  
OUTPUT RECORDS:         34  
WALL TIME:               2.224(SEC)  
CPU TIME:                .641(SEC)

\*\*\*\*\*

END OF PROGRAM

## Program Input

The following table provides descriptions of the **NRPT02** reduction options.

Reduction Option	Description
<b>Enter Option</b>	The number corresponding to the desired reduction option. <b>1</b> reads through statistic log files to compile the data. <b>2</b> uses a reduction file created from a previous run of <b>NRPT01</b> or <b>NRPT02</b> .
<b>Enter first Stat log #</b>	The number of the first log file to be extracted. Applies only if option <b>1</b> is selected.
<b>Enter last Stat log #</b>	The number of the last log file to be extracted. Applies only if option <b>1</b> is selected.
<b>Enter file name to save reduction</b>	If other reports are to be run against the same data, the reduction can be reused. If a file name is entered here, the reduction data will be save under this file name.
<b>Enter the file name of the reduction to use</b>	The name of a reduction file created by a previous run of <b>NRPT01</b> or <b>NRPT02</b> . Applies only if option <b>2</b> is selected.
<b>Print User Information</b>	Entering <b>N</b> summarizes down to the program or file level. Entering a <b>Y</b> summarizes down to the <b>LDEV</b> , user, program or file level.

## Program Output

The following table provides definitions of the terms used in the **NBRPT02** report.

Term	Description
<b>OPENS</b>	The total number of times that this file was opened by this program.
<b>AVG ACCESS TIMES</b>	<p><b>Wall</b> — The average amount of elapsed time in msec that it took to complete all calls against this file. This is computed:</p> <p style="text-align: center;"><b>Total Wall time/Count</b></p> <p><b>CPU</b> — The average amount of CPU time in msec that it took to process all calls against this file. This is computed:</p> <p style="text-align: center;"><b>Total CPU time/Count</b></p> <p><b>Impede</b> — The average amount of time that this process spent impeded waiting for a lock. The impede time is calculated:</p> <p style="text-align: center;"><b>(Total Elapsed lock - Total CPU lock)/Lock Count</b></p> <p><b>Overall Count</b> — The total number of calls issued against this file by this process.</p>
<b>% OF PROCESS TOTAL</b>	<p><b>Wall</b> — The percentage of wall time that this process spent accessing this file. This is calculated:</p> <p style="text-align: center;"><b>Total Wall this file / Total Process wall * 100</b></p> <p><b>CPU</b> — The percentage of CPU time that this process spent accessing this file. This is calculated:</p> <p style="text-align: center;"><b>Total CPU this file / Total Process CPU * 100</b></p> <p><b>Impede</b> — The percentage of process wall time that this process spent waiting for a lock.</p>

## Generating Scenarios

Any network strategy can be modeled using statistics gathered from the current environment, standard NetBase reports and the scenario generator **NBLGTRAN**. The statistics provide the file and program activity. The reports display the information, and the scenario generator transforms the statistics to reflect the desired configuration.

**NBLGTRAN** creates new statistic log files from existing files, editing the file as it is being processed. Based on specifications entered into a scenario file, **NBLGTRAN** consolidates, excludes and/or relocates files and programs into the new log file.

The scenario file is an unnumbered ASCII file that is created and maintained using any text editor. The format of the records contain a record type and data description. The record type indicates the action that is to be taken while the data description describes what data is to be acted upon.

Record Type	Description
CF	<b>Consolidate File.</b> This command renames a file, or a group of files specified by a file set, into a file name specified.
CP	<b>Consolidate Program.</b> This command renames a program, or a group of programs specified by file set, into the program name specified.
EF	<b>Exclude File.</b> This command removes the file, or group of files specified by file set, from the log file.
EP	<b>Exclude Program</b> This command removes a program, or a group of programs specified by file set, from the log file.
MF	<b>Move File.</b> This command changes the node number associated with a data file, or a group of files specified by file set, to a particular node.
MP	<b>Move Program.</b> This command changes the node number associated with a program, or a group of programs specified by file set, to a particular node.

## Running NBLGTRAN

**NBLGTRAN** may be run either as a job or session. Since log files may span several files, the **RANGE** alternate entry point allows several consecutive log files to combine into one transformed log file. The following sample shows the processing of one log file.

```
:RUN NBLGTRAN.PUB.NETBASE
NetBase Log file Translator [v.u.f] Update n Copyright QUEST Software 1990
SCENARIO FILE.... CEI01.TEST
File Consolidations: 4
Program Relocations: 2
File Relocations: 5
File Exclusions: 1
LOG FILE..... NBLG0000.DATA
OUTPUT FILE..... NBLG9000.DATA
TRANSLATING LOG FILE: NBLG0000.DATA.NETBASE
Records Processed: 2034
Records Updated: 1022
Records Excluded: 208
Program Names Updated: 30
File Names Updated: 130
Programs Relocated: 23
Files Relocated: 40
SCENARIO FILE.... <<RETURN to exit>>
END OF PROGRAM
```



**Note.** Since the statistics reports expect the log file names to be in the format **NBLG###.DATA.NETBASE**, the output file name should be in the same format (shown above).

---

If a range of log files is to be translated, the **RANGE** entry point should be used. This reduces the effort to process the logs and consolidates the scenario output into one log file. The following example illustrates this.

```
:RUN NBLGTRAN.PUB.NETBASE,RANGE
```

```
NetBase Log file Translator [v.u.f] (C) QUEST Software 1987
```

```
SCENARIO FILE.... CEI01.TEST
```

```
Records Processed:      2034  
Records Updated:        1022  
Records Excluded:       208
```

```
FIRST LOG FILE NUMBER: 0  
LAST LOG FILE NUMBER: 3
```

```
Log files in range:     1  
Total records logged:   5
```

```
OUTPUT FILE..... NBLG9000.DATA  
OUTPUT FILE EXISTS:   PURGE IT? Y
```

```
TRANSLATING LOG FILE:  NBLG0000.DATA.NETBASE  
TRANSLATING LOG FILE:  NBLG0001.DATA.NETBASE  
TRANSLATING LOG FILE:  NBLG0002.DATA.NETBASE  
TRANSLATING LOG FILE:  NBLG0003.DATA.NETBASE
```

```
Records Processed:      2034  
Records Updated:        1022  
Records Excluded:       208
```

```
Program Names Updated:  30  
File Names Updated:    130  
Programs Relocated:     23  
Files Relocated:        40
```

```
SCENARIO FILE....
```

```
END OF PROGRAM
```

## Scenario File Format

The scenario file command records are prefixed with a two character code indicating the action that is to take place. The delimiter used in all records is the slash (/). Specifying file and program names is accomplished using the MPE file set format complete with wild card characters. The file and program names must be fully qualified.

The order in which the command records are entered is important. **NBLGTRAN** first excludes files and programs. Next, files and programs are renamed. Finally, the node numbers are changed. Once the file or program name matches any file set, further transformations of the same type are skipped.

Keep the following items in mind while entering the scenario file:

1. Enter all of the Exclusion records first, followed by the Consolidation records, and then the Move records.
2. Since the first matching file set is used, specify the most specific file sets before the more general cases. This will ensure that the desired mapping occurs.



**Note.** Extremely long scenario files will significantly slow down the translation process.

---

### Example

The following example transforms the log file by first deleting all references to **LOG####.DATA.MFG**. All references to the program **FIXIT** are deleted. All files matching the file set **SORT####.WORK.MFG** are consolidated into **SORT.WORK.MFG**. All other files in the **WORK** group are consolidated into the **MISC** file. Finally, all files in the **DATA** group of the MFG account are moved to node 2.

```
EF/LOG####.DATA.MFG/  
EP/FIXIT.PROG.MFG/  
CF/SORT####.WORK.MFG/SORT.WORK.MFG  
CF/@.WORK.MFG/MISC.WORK.MFG  
MF/@.DATA.MFG/2
```

## Consolidating Records

You can consolidate records by renaming a program or file, or a group of programs or files, to one file. This is very useful in improving the readability of the statistic reports. If a file or program name matches the target file name, **NBLGTRAN** substitutes the new file name in lieu of the actual file name. The **CF** command changes the file name, while the **CP** command changes the program name. See the following syntax for **CF** and **CP**.

```
CF/target-file set/new-file name
CP/target-file set/new-file name
```

### Examples

To combine all files in the **WORK** group into one file called **MISC**, use the following scenario record:

```
CF/@.WORK.MFG/MISC.WORK.MFG
```

Similarly, to combine all programs running out of the **FIXIT** group into one program called **FIXIT**:

```
CP/@.FIXIT.MFG/FIXIT.PROG.MFG
```

## Removing Records

You can remove certain files of limited interest, such as menu driver files, from the log files. If a file or program name matches the target file name, **NBLGTRAN** will delete that log file record from the new log file. The **EF** command will delete the record if the file name matches, while the **EP** command deletes if the program name matches. See the following syntax for **EF** and **EP**.

```
EF/target-file set/
EP/target-file set/
```

### Examples

To exclude a file or set of files from the log file, use the exclude command. To exclude all files accessed in the **TEMP** group, use the following scenario command:

```
EF/@.TEMP.MFG/
```



**Note.** The trailing / must be entered, otherwise an invalid file name error will result.

---

Similarly, to exclude the program **FIXIT**, use the following scenario command:

```
EP/FIXIT.PROG.MFG/
```

### Moving Records

You can change the node number associated with the file or program to the node specified. The move command allows modeling of different scenarios by moving processes and files to other computers. If a file or program name matches the target file name, **NBLGTRAN** changes the associated node to that specified by new node number. The **MF** command changes the node for files, and the **MP** command changes the node for programs. See the following syntax for **MF** and **MP**.

```
MF/target-file set/new-node#  
MP/target-file set/new-node#
```

### Examples

To determine the effect of moving all files in the **DATA** group to a new node use the following scenario command:

```
MF/@.DATA.MFG/2
```

Similarly, you can determine the effect of moving programs using the same command. In this case, all processes associated with the **ACCT** account are moved to node 3.

```
MP/@.@.ACCT/3
```

## CHAPTER 10

# NetBase and Third Party Products

This chapter documents the relationship of NetBase to other third party products that may be installed on your system.

### In This Chapter

- **ADAGER**
- **DBGENERAL**
- **OMNIDEX**
- **NetBase MPEX Interface**

If you have any questions regarding NetBase and third party products, please contact Technical Support at 714-720-1434.

## NetBase ADAGER Interface

This interface is used in shadowing environments and allows for ADAGER database changes that are performed on the master system to be automatically sent to the shadow system and be performed on the shadow copy of the data. In order to use the NetBase ADAGER Interface, a special command file, **ADAGER.PUB.NETBASE**, must be used to create an ADAGER job to perform the structural changes.

All commands input into this are put into a job stream. When the job stream executes, it runs the program **ADAGER.PUB.REGO**. Once ADAGER has successfully completed the changes on the master system, **NBADAGER** writes all of the ADAGER commands into the shadow transport file, **NBM.IPC.NETBASE**, and sends them to the shadow system.

When the posting process, **NBPOST**, receives the ADAGER commands, it automatically puts the database into a queue state. **NBPOST** then invokes ADAGER on the shadow system to make the same changes as were made on the master. A temporary file with the name **Annnnnn** (where **n** is a random digit) is created in the group with the root file on the shadow system. This file contains the ADAGER commands. It will be automatically purged upon completion of the ADAGER changes. Once the ADAGER changes complete successfully, **NBPOST** does a roll forward on the database.

### Requirements

- ADAGER must be installed on both the master and shadow systems
- Version of ADAGER equal to or greater than 941201
- NetBase version 9.6
- NetBase directory program entries for **NBADAGER.PUB.NETBASE** and **@.PUB.REGO**

### Syntax

```
:NBDIR.PUB.NETBASE
D> PROGRAM NBADAGER.PUB.NETBASE;NONB
D> PROGRAM @.PUB.REGO;NONB
D> EXIT
```

### To use ADAGER:

1. Execute the command file, **ADAGER.PUB.NETBASE**. This will run **ADAGER.PUB.REGO**, which will prompt for the name of the job stream, the database name, and changes to be made. When this step is completed, the job stream will have been created.
2. Stream the job created in Step 1. This will execute the ADAGER changes and once it has completed successfully, **NBADAGER** will send the changes to the shadow system.

### Notes

- Online changes made with ADAGER will *not* be shadowed.
- The command file, **ADAGER.PUB.NETBASE**, must be used.
- Only one database can be modified per execution of the command file.

## NetBase DBGENERAL Interface

This interface is used in shadowing environments and allows for DBGENERAL database changes that are performed on the master system to be automatically sent to the shadow system and be performed on the shadow copy of the data.

When DBGENERAL is run, the input is captured and applied to the master system. Once DBGENERAL has successfully completed the changes on the master system, NBGENERAL writes all of the DBGENERAL commands into the shadow transport file, **NBM.IPC.NETBASE**, and sends them to the shadow system.

When the posting process, **NBPOST**, receives the DBGENERAL commands, it automatically puts the database into a queue state. **NBPOST** then invokes DBGENERAL on the shadow system to make the same changes as were made on the master. Once the DBGENERAL changes complete successfully, **NBPOST** does a roll forward on the database.

### Requirements

- DBGENERAL must be installed on both the master and shadow systems
- DBGENERAL Version 7.01.02 or later
- NetBase 9.7

### Syntax

```
:NBDIR.PUB.NETBASE  
D> PROGRAM NBGENRL.PUB.NETBASE;NONB  
D> PROGRAM @.PUB.BRADMARK;NONB  
D> EXIT
```

### To use DBGENERAL:

1. Run **DBGENRL.PUB.BRADMARK** on the master machine and input changes.
2. NetBase will automatically perform the same changes on the shadow machine

## NetBase OMNIDEX Interface

This section documents how NetBase and OMNIDEX 3.x work for both NFA and for shadowing.

### NFA Requirements

If using OMNIDEX 3.x or later and NetBase for remote file access, the NetBase OMNIDEX interface must be installed.

### Installation Requirements

- `NBINST.PUB.NETBASE;INFO="INSTALL OMNIDEX"`
- `:NBDIR.PUB.NETBASE`  
`D> XL OMNIDEX=XLOMNIDX.PUB.SYS`  
`D> EXIT`
- If the application program references a library (XL) that contains the OMNIDEX procedures, either the library can no longer be referenced or the OMNIDEX modules must be removed from the library.

## How the Interface Works

The following steps show how the NetBase OMNIDEX Interface works when accessing a remote database using the OMNIDEX **dbiopen** procedure.

1. On the local system, the application program calls **dbiopen**.
2. The local system **dbiopen** goes to the **dbiopen** in **XL.PUB.SYS** (this is the NetBase **dbiopen**). NetBase then takes the **dbiopen** to the remote system
3. On the remote system, NMServer calls **dbiopen** in **XLOMNIDX.PUB.SYS** (library in the **NBDIR** entry).
4. The remote system **dbiopen** then calls **dbopen** in **XL.PUB.SYS** and opens the database.

## Special Installation Steps for Shadowing

- If using OMNIDEX 3.x and NetBase in a shadowing environment, there are no special installation steps to be performed by default.
- If the customer does not own NFA and is using TPI, then nothing needs to be done on the shadow system as far as NetBase is concerned. If the customer wants the shadow copy to be indexed, then they would set it up via **DBUTIL**, and OMNIDEX and TPI will automatically maintain the indexes.
- If the customer owns NFA and is using TPI or non-TPI, and they are going to use NFA to access the master copy, then the **INSTALL OMNIDEX** must be done on the master and shadow system. This still does not effect shadowing, it is strictly for NFA.
- If the customer is using non-TPI and they want the indexes to be maintained on the shadow system, then they must prep **NBPOST** with the **Call Conversion** library - this is job **NBPOST3X.JOB.NETBASE**.

## NetBase MPEX Interface

This section documents the NetBase/MPEX integration command files, intended to allow **VESOFT's** MPEX to be safely used in a NetBase environment.

The following MPEX commands are intercepted by the NetBase/MPEX interface:

```
%ALTFILE  
%BUILD  
%COPY  
%PURGE  
%RENAME  
%SAVE  
%YESPURGE
```

When any of the above commands are used, the NetBase/MPEX interface will intercept and execute the command, instead of the normal MPEX command file. The NetBase/MPEX interface is designed to emulate normal MPEX operation wherever possible. The request is passed to NetBase only in cases where the source file (or, when applicable, the target file) is either a NetBase shadowed or a NetBase remote file.

### Requirements

- NetBase Version 9.7 or later
- MPEX: 25n60119 (**NM MAIN**)
- The following line must be added to **MPEXMGR.PUB.VESOFT** file:  

```
XEQ STARTUP.NETBASE.VESOFT
```
- The following NetBase Directory entry must be made:  

```
:RUN NBDIR.PUB.NETBASE  
D> PROGRAM MAIN.PUB.VESOFT;LIB=D  
D> EXIT
```

### Notes

For detail information regarding the NetBase MPEX Interface, see the file **README.MPEX.NETBASE**

## CHAPTER 11

# System Operations and Maintenance

With NetBase installed on your system, many of your operational procedures must change. In this chapter, you will find recommended and required changes with steps to follow.

### In This Chapter

- **System Startup**
- **System Shutdown**
- **Partial Backup**
- **Full Backup**
- **HP Patches**
- **FOS Upgrades**
- **NetBase Updates**
- **Removing NetBase**

## System Startup

The system startup procedures need to change to include the starting of NetBase prior to any jobs or users logging on. The following changes are recommended.

Action	Enter	Explanation
1. Start the network	<pre>:NETCONTROL START;NET=LAN1* :NETCONTROL START;NET=LOOP :NSCONTROL START</pre>	<p>NetBase requires the network to be running prior to its starting.</p> <p>* Substitute in your actual network name</p>
2. Enable NetBase	<pre>:RUN NBVERS.PUB.NETBASE,ENABLE</pre>	<p>To enable NetBase to intercept compatibility mode (CM) activities.</p>
3. Start NetBase	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"</pre>	<p>To start the NetBase background job, starting NetBase transport services.</p>
4. Normal startup procedures		<p>Any other processing that is done in the <b>SYSSTART</b> file.</p>

## System Shutdown

The system shutdown procedures need to change to include the stopping of NetBase prior to shutting down the system. The following changes are recommended.

Action	Enter	Explanation
1. Stop NetBase	<pre>:RUN NBCTRL . PUB . NETBASE ; INFO = " STOP NETBASE "  :PAUSE <i>nnn</i>*</pre>	<p>To stop the NetBase job gracefully, allowing all processes - shadowing, NFA, AutoRPM... to cease without adversely affecting the data.</p> <p>(The pause is recommended if this is in a background job, to allow time for the shutdown to complete, prior to stopping the network.)</p> <p><i>nnn</i>* - The length of the pause depends on the amount of time it takes NetBase to shutdown on your system.</p>
2. Stop the network	<pre>:NSCONTROL STOP  :NETCONTROL STOP ; NET = LAN1 **  :NETCONTROL STOP ; NET = LOOP</pre>	<p>Shutdown the network, now that NetBase has stopped gracefully.</p> <p>** You may have multiple LANs or a different network name. Substitute your network information here.</p>
3. Shutdown the system	however you choose	



**Note.** Shutting down NetBase must be complete prior to stopping the network. Shutting down NetBase must be done via **NBCTRL** only. Aborting the NetBase background job may cause synchronization errors when shadowing. Aborting the NetBase job can adversely affect programs running remotely, users' accessing data remotely, and remote spooling. *Do not abort the NetBase background job.*

## Partial Backup

If you are shadowing, you can change your partial backups to take place on the shadow system. If you have remote users accessing local data to be stored, your backup procedure will change as well. Below is the process we recommend for partial backups.

Action	Enter	Explanation
<b>Flush the pipeline (optional)</b>	<pre>:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM=s hadownode</pre> <p>(on the master node)</p>	<p>This step is optional. If performed, it will force all updates in the shadowing pipeline to post to the shadow files, giving you a sync point at which time both the master and shadow data matches.</p> <p>This step does not interfere with production on the master system.</p>
<b>Stop post</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP POST"</pre> <p>(on the shadow node)</p>	<p>This step is required if post was not stopped via <b>NBFLUSH</b>. (This frees the files to be stored.)</p>
<b>Stop NFA access</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP REMOTE"</pre> <p>(on the shadow node)</p>	<p>This step is required if you have remote users accessing data on this system which you want to backup.</p>
<b>Perform backup</b>	<p>however you do it (on the shadow node)</p>	
<b>Resume posting</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="START POST"</pre> <p>(on the shadow node)</p>	<p>This tells NetBase to catch up, posting any transactions that took place on the master while the backup was being performed on the shadow.</p>
<b>Resume NFA access</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="START REMOTE"</pre> <p>(on the shadow node)</p>	<p>This instructs NetBase to allow remote users to access local data.</p>

## Full Backup

For a full backup to include NetBase, you will need to stop the NetBase job.

Action	Enter	Explanation
<b>Flush the pipeline (optional)</b>	<pre>:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM= shadownode</pre> <p>(on the master node)</p>	<p>This step applies to shadowing environments only, and it is optional for those. If performed, it will force all updates in the shadowing pipeline to post to the shadow files, giving you a sync point at which time both the master and shadow data matches.</p> <p>This step does not interfere with production on the master system.</p>
<b>Stop NetBase</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"</pre>	<p>This step gracefully stops all NetBase activities, including NFA, Shadowing, AutoRPM and Remote spooling.</p>
<b>Stop the network</b>	<pre>:NSCONTROL STOP :NETCONTROL STOP;NET=LAN1* :NETCONTROL STOP;NET=LOOP</pre>	<p>Shutdown the network, now that NetBase has stopped gracefully.</p>
<b>Perform backup</b>	<p>however you do it</p>	
<b>Restart the network</b>	<pre>:NETCONTROL START;NET=LAN1* :NETCONTROL START;NET=LOOP :NSCONTROL START</pre>	
<b>Restart NetBase</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"</pre>	

\* You may have multiple LANs or a different network name. Substitute your network information.

## HP Patches

Having NetBase installed into your system SL and XL affects HP patch procedures. If you are applying a Power Patch or an Image patch, the patch will expect the Image intrinsics to be where HP put them, as opposed to where NetBase renamed them, when installing the shell. To handle this:

Action	Enter	Explanation
<b>Flush the pipeline (optional)</b>	<pre>:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM= shadownode</pre> <p>(on the master node)</p>	This step applies to shadowing environments only, and it is optional for those. If performed, it will force all updates in the shadowing pipeline to post to the shadow files, giving you a sync point at which time both the master and shadow data matches.
<b>Stop NetBase</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"</pre>	This step gracefully stops all NetBase activities, including NFA, shadowing, AutoRPM and remote spooling.
<b>Disable CM NetBase and UDC</b>	<pre>:RUN NBVERS.PUB.NETBASE,DISABLE :SETCATALOG NMCMDUDC.PUB.NETBASE;SYSTEM; ;DELETE</pre>	This toggles off the redirection of CM processes to RTMSL and unloads our UDC.
<b>Remove the shell</b>	<pre>:HELLO MANAGER.SYS,PUB :RUN NBINST.PUB.NETBASE;INFO="REMOVE"</pre> <p>(reboot the system)</p>	This places the IMAGE intrinsics back where they are expected by HP patches.
<b>Apply the HP patch</b>	per HP's instructions (except do not reset NetBase system UDC)	
<b>Reinstall NetBase shell</b>	<pre>:HELLO MANAGER.SYS,PUB :RUN NBINST.PUB.NETBASE;INFO="INSTALL" (reboot the system) :SETCATALOG NMCMDUDC.PUB.NETBASE;SYSTEM ;APPEND</pre>	This puts the NetBase shell in place to intercept IMAGE intrinsics, and re-instates the NetBase UDCs.

## FOS Upgrades

When you replace the operating system, you must re-install NetBase.

Action	Enter	Explanation
<b>Flush the pipeline (optional)</b>	<pre>:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM= shadownode</pre> <p>(on the master node)</p>	This step applies to shadowing environments only, and it is optional for those. If performed, it will force all updates in the shadowing pipeline to post to the shadow files, giving you a sync point at which time both the master and shadow data matches.
<b>Stop NetBase</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"</pre>	This step gracefully stops all NetBase activities, including NFA, shadowing, AutoRPM and remote spooling.
<b>Disable NetBase and the NMCMDUDC</b>	<pre>:RUN NBVERS.PUB.NETBASE,DISABLE :SETCATALOG NMCMDUDC.PUB.NETBASE;SYSTEM ;DELETE</pre>	This toggles off the redirection of CM processes to RTMSL.
<b>Install the new FOS</b>	per HP's instructions	
<b>Reinstall NetBase shell</b>	<pre>:HELLO MANAGER.SYS,PUB :RUN NBINST.PUB.NETBASE;INFO="INSTALL"</pre> <p>(reboot the system)</p> <pre>:SETCATALOG NMCMDUDC.PUB.NETBASE; SYSTEM;APPEND</pre>	This puts the NetBase shell in place to intercept Image intrinsics.

## NetBase Updates

To update your version of NetBase, you will need to follow the instructions with the tape. In general, you will:

Action	Enter	Explanation
<b>Flush the pipeline</b>	<pre>:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM=S hadownode</pre> <p>(on the master node)</p>	This step is <i>required</i> for shadowing environments. It will force all updates in the shadowing pipeline to post to the shadow files, eliminating records from queue files which may change size or structure with the update.
<b>Stop NetBase</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"</pre>	This step gracefully stops all NetBase activities.
<b>Disable NetBase and the UDC</b>	<pre>:RUN NBVERS.PUB.NETBASE,DISABLE :SETCATALOG NMCMDUDC.PUB.NETBASE ;SYSTEM;DELETE</pre>	This toggles off the redirection of CM processes to RTMSL.
<b>Disable NetBase UDCs</b>	<pre>:SETCATALOG NBUDC.PUB.NETBASE;DELETE</pre>	If you have set NetBase UDCs, you will need to reset them so that they may be updated/replaced.
<b>Follow the installation instructions included with the tape.</b>		

## Removing NetBase

To remove NetBase completely from the system, follow the instructions below:

Action	Enter	Explanation
<b>Flush the pipeline</b>	<pre>:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM= shadownode</pre> <p>(on the master node)</p>	This step is <i>required</i> for shadowing environments. It will force all updates in the shadowing pipeline to post to the shadow files, eliminating records from queue files which may change size or structure with the update.
<b>Stop NetBase</b>	<pre>:RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"</pre>	This step gracefully stops all NetBase activities.
<b>Disable NetBase and the UDC</b>	<pre>:RUN NBVERS.PUB.NETBASE,DISABLE :SETCATALOG NMCMDUDC.PUB.NETBASE ;SYSTEM;DELETE</pre>	This toggles off the redirection of CM processes to RTMSL.
<b>Disable NetBase UDCs</b>	<pre>:SETCATALOG NBUDC.PUB.NETBASE;DELETE :HELLO MANAGER.SYS,PUB</pre>	If you have set NetBase UDCs, you will need to reset them so that they may be updated/replaced.
<b>Remove the NetBase Shell</b>	<pre>:RUN NBINST.PUB.NETBASE;INFO="REMOVE"</pre> <p>Reboot the system</p>	Remove the NetBase Shell from <b>PUB.SYS</b> .
<b>Purge the NetBase Account</b>	<pre>:PURGEACCT NETBASE</pre>	Remove the NetBase account from the system.

## CHAPTER 12

# Disaster Recovery Procedures for NetBase Shadowing Environments

This chapter serves as a guideline for NetBase failover procedures. It covers three scenarios in a shadowing environment. It addresses both planned and unplanned failovers in each scenario. This document does not include site specific procedures that may need to be performed.

### In This Chapter

- **Case One:** Master system fails and the shadow system becomes the master system with shadowing in the reverse direction.
- **Case Two:** Master system fails and the shadow system is available for **READ** only access.
- **Case Three:** Master system fails and the shadow system is set up to operate in a “standalone” environment. No shadowing.
- **Creating Alternate Directories with NBDIR.PUB.NETBASE**

## Case One

The Case One scenario assumes that the master system fails and the shadow system becomes the master system, with shadowing in the reverse direction.

### Planned Failover

The following guidelines can be used for a planned failover. An example of a planned failover would be hardware maintenance.

#### On the master system:

1. Quiet the system (except for the NetBase background job).

2. Flush all transactions to the shadow system:

```
RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM=ShadowNode#
```

3. Change the directory to reflect that this will now be the “shadow” system:

```
RUN NBCTRL.PUB.NETBASE;INFO="CHGDIR newdir"
```

4. Stop and start the NetBase Background job:

```
RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"  
RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"
```

#### On the shadow system:

5. Quiet the system (except for the NetBase background job).

6. Check the directory for any out of sync conditions.:

```
RUN NBSYNC.PUB.NETBASE,VERIFY;INFO="@.@@"
```

If any files are out of sync, they should be resynced before continuing.

7. Change the directory to reflect that this will now be the “master” system:

```
RUN NBCTRL.PUB.NETBASE;INFO="CHGDIR newdir"
```

8. Stop and start the NetBase background job:

```
RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"  
RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"
```

9. Let users on to the new "master" system.

When it is time to switch back to the original master/shadow configuration, the above steps for a planned failover would be followed again.

## Unplanned Failover

The following guidelines can be used for an unplanned failover. This scenario assumes that the master will become the shadow system and the shadow will become the master. This scenario also assumes that the master system is “dead” or unavailable.

### On the shadow system:

1. Quiet the system (except for the NetBase background job).
2. Check to make sure all shadow transactions have been posted:  

```
RUN NBCTRL.PUB.NETBASE;INFO="SHOW POST"
```
3. The backlog value should be **0** and the state of **POST** should be **IDLE**.
4. Change the directory that will reflect this is the “master” system:  

```
RUN NBCTRL.PUB.NETBASE;INFO="CHGDIR newdir"
```
5. Start an online backup of the data. This is done since the condition of the original master is not known.
6. Stop/start the NetBase background job.
7. Allow users and processing to start on the new “master”.

At this point, the shadow system is now the “master” system.

When the old master system is available, it must be brought up to think that it is a shadow system. Follow the steps below for restarting the failed system as a “shadow” system.

**On the OLD master system:**

1. *Do not* start the NetBase background job at system startup.
2. Check the **NBM.IPC** file and make sure it is empty. If it is not empty, it must be purged and rebuilt:

```
PURGE NBM.IPC.NETBASE
BUILD NBM.IPC.NETBASE;REC=8187;MSG;DISC=31000,32
```

3. Delete any transactions in the export queue files:

```
PURGE EQ#####.QUEUE
```

4. Change to a directory that says this is the shadow system:

```
RUN NBCTRL.PUB.NETBASE;INFO='CHGDIR newdir'
```

5. Set the **SHADOW - Post on Startup.....** flag to **N**.

```
RUN NBCTRL.PUB.NETBASE
*> CONFIG
C> M SHADOW_____ SHADOW - Post On Startup..... N
```

6. Start the NetBase background job.:

```
RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"
```

This will allow the transactions to be sent over but not posted.

7. Start the restore that was made at the time of the failover.

8. When the restore is complete, start the posting process:

```
RUN NBCTRL.PUB.NETBASE;INFO="START POST"
```

When the posting process has completed all transactions and it is time to switch back to the original master/shadow configuration, you can perform the steps in a planned failover.

## Case Two

The Case Two scenario assumes that the master system fails and the shadow system is available for **READ** only access.

By default, shadow data is available for **READ** only access. If applications will be run on the shadow system during the failover that want to access the data in a **MODE 1**, then the **;LOCAL** option must be added to the directory entries to force the application into **READ** mode only.

## Case Three

The Case Three scenario assumes that the master system fails and the shadow system is set up to operate in a “standalone” environment. No shadowing will be done during this failover.

### Planned Failover

The following guidelines can be used for a planned failover.

#### On the master system:

1. Quiet the system (except for the NetBase background job).

2. Flush all transactions to the shadow system.

```
RUN NBFLUSH.PUB.NETBASE, STOPWAIT; PARM=ShadowNode#
```

3. Stop the NetBase background job:

```
RUN NBCTRL.PUB.NETBASE; INFO="STOP NETBASE"
```

#### On the shadow system:

4. Quiet the system (except for the NetBase background job).

5. Check the directory for any out of sync conditions:

```
RUN NBSYNC.PUB.NETBASE, VERIFY; INFO="@.@@"
```

If any files are out of sync, they should be resynced before continuing.

6. Change the directory to reflect that this is a “standalone” system:

```
RUN NBCTRL.PUB.NETBASE; INFO="CHGDIR newdir"
```

7. Stop and start the NetBase background job:

```
RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"  
RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"
```

8. Let users on to the new “standalone” system.

## Unplanned Failover

The following guidelines can be used for a unplanned failover. This scenario assumes that the shadow system will no longer be a shadow system and it will operate in a “standalone” environment.

### On the shadow system:

1. Quiet the system (except for the NetBase background job).
2. Check to make sure all shadow transactions have been posted:  

```
RUN NBCTRL.PUB.NETBASE;INFO="SHOW POST"
```
3. The backlog value should be **0** and the state of **POST** should be **IDLE**.
4. Change the directory that will reflect this is a “standalone” system.  

```
RUN NBCTRL.PUB.NETBASE;INFO="CHGDIR newdir"
```
5. Stop/start the NetBase background job.
6. Allow users and processing to start on the new “standalone” system.

At this point, the shadow system is now operating in a “standalone” environment.

Whether a planned or unplanned failover is used in this scenario, at some point it will be necessary to get back to the original configuration. When it is time to go back to the original configuration of master and shadow, there are several different ways that this can be done.

One method would be to make a store tape of the standalone system, and then restore it on to the original master and switch back to the original directories. However, this process could take a long time for the store and restores to complete. This method would involve a lot of “downtime” to users.

Another method would be to set the “standalone” system up as the master system and bring the old master up as a shadow system. This method would minimize the amount of downtime to the users. After the systems are brought up, then a planned failover could be performed to get the systems back to their original master/shadow configuration.

Regardless of which method is chosen, the following steps must be performed on the old master before any NetBase processing can begin.

### On the OLD master system:

1. Check the **NBM.IPC** file and make sure it is empty. If it is not empty, it must be purged and rebuilt:

```
PURGE NBM.IPC.NETBASE  
BUILD NBM.IPC.NETBASE;REC=8187;MSG;DISC=3100,32
```

2. Delete any transactions in the export queue files:

```
PURGE EQ#####.QUEUE.NETBASE
```

## Creating Alternate Directories with NBDIR.PUB.NETBASE

In order to implement multiple scenarios with NetBase, it requires the use of alternate directories. These directories should be prebuilt whenever possible. Below are the steps outlining how to take a current NetBase directory that defines shadowing for a master system and create a directory for the master system that now defines it as the shadow system.

Unload the current NetBase directory using **NBDIR**. This will create an ASCII file that can be manipulated with any editor on the HP3000.

```
:RUN NBDIR.PUB.NETBASE
UNLOAD filename
```

The **filename** is the name of the ASCII file which can now be edited to make the necessary changes. Here is a sample of the ASCII file:

```
:PRINT filename
DIR NBD.DATA.NETBASE <<This is the directory name>>
FILE BANNER.DATA.NETBASE,DAFFY,BUGS <<File entry for Shadowing>>
PROGRAM @.PUB.BRADMARK;NONB
PROGRAM NBPOST.NB.NETBASE;NONB
PROGRAM @.NFSPROG.NETBASE;NONB
PROGRAM NBADAGER.PUB.NETBASE;NONB
PROGRAM NBGENRL.PUB.NETBASE;NONB
PROGRAM NBINST.PUB.NETBASE;NONB
PROGRAM @.PUB.REGO;NONB
PROGRAM VTSERVER.NET.SYS;NONB
PROGRAM CI.PUB.SYS;NONB
PROGRAM DBRESTOR.PUB.SYS;NONB
PROGRAM DBSCHEMA.PUB.SYS;NONB
PROGRAM DBSTORE.PUB.SYS;NONB
PROGRAM DBUTIL.PUB.SYS;NONB
PROGRAM @.@.VESOFT;NONB
EXIT
```

The next step is to edit the above file and change the necessary directory entries for the system to think it is the shadow system. In the above example, two lines need to be changed:

```
DIR NBD.DATA.NETBASE
File Banner.Data.NetBase,Daffy,Bugs
```

The **DIR** command needs to be changed to a **BUILD** command so that we will create a new directory. We do not want to modify the existing directory. Change the **FILE** statement to reflect that **BUGS** will be the master and **DAFFY** will be the shadow. After making the changes the new file looks like:

```
:PRINT filename
BUILD DRDIR.DATA.NETBASE <<This is the new directory name>>
FILE BANNER.DATA.NETBASE,BUGS,DAFFY <<Changed FILE entry>>
PROGRAM @.PUB.BRADMARK;NONB
PROGRAM NBPOST.NB.NETBASE;NONB
PROGRAM @.NFSPROG.NETBASE;NONB
PROGRAM NBADAGER.PUB.NETBASE;NONB
PROGRAM NBGENRL.PUB.NETBASE;NONB
PROGRAM NBINST.PUB.NETBASE;NONB
PROGRAM @.PUB.REGO;NONB
PROGRAM VTSERVER.NET.SYS;NONB
PROGRAM CI.PUB.SYS;NONB
PROGRAM DBRESTOR.PUB.SYS;NONB
PROGRAM DBSCHEMA.PUB.SYS;NONB
PROGRAM DBSTORE.PUB.SYS;NONB
PROGRAM DBUTIL.PUB.SYS;NONB
PROGRAM @.@.VESOFT;NONB
EXIT
```

Now that the ASCII file contains the correct entries, use **NBDIR** to create this alternate directory.

```
:RUN NBDIR.PUB.NETBASE;STDIN=filename
```

At this point, **NBDIR** will read through the file and create a directory called **DRDIR.DATA.NETBASE**. This directory can now be used at the time of a failover. Remember that creating a directory using **NBDIR** does not make it the “active” directory on the system. The **CHGDIR** command in **NBCTRL** must be used for that.

## CHAPTER 13

# NetBase Utilities

A number of utilities are provided with NetBase. Some utilities, such as NBVERS, are used to support the NetBase installation and address specific operational needs. Others are generic utilities that have been Quest contributions to the INTEREX CSL.

### In This Chapter

- NBVERS
- NBLOG
- NBTRACE
- QUARK

Two additional utilities are described in this chapter. *These two utilities must be purchased separately.* The utilities listed above come with the NetBase product.

- QUARK+
- NBCOPY

## NBVERS

The **NBVERS** utility is used to check the versions of the various NetBase programs and segments that have been installed on the system. When NetBase is globally installed, it is used to control the enabling and disabling of the interception of compatibility mode programs.

The syntax for running **NBVERS** is as follows:

```
:RUN NBVERS.PUB.NETBASE[ ,entrypoint][ ;INFO="filename"]
where:
entrypoint =      PROG
                   SHELL
                   ENABLE
                   DISABLE
                   STATE
```

If **NBVERS** is run without an entry point, it checks both the program and the segment versions that are installed on the system. For NetBase programs, **NBVERS** verifies that each program can load and that their versions are reasonably matched. For the NetBase Shell, **NBVERS** verifies the versions of the shell in **PUB.SYS**. If the versions do not match, **NBVERS** will terminate with an error.

The alternate entry points and the values for the **INFO** parameter operate as follows:

Entry Point	Description
<b>PROG</b>	Displays the version of each installed NetBase component.
<b>SHELL</b>	Displays the version of both the NM and CM shells installed.
<b>ENABLE</b>	Begins interception of all subsequent CM processes when NetBase is globally installed.
<b>DISABLE</b>	Terminates the interception of all subsequent CM processes when NetBase is globally installed.
<b>STATE</b>	Displays whether NetBase is currently intercepting CM processes ( <b>ENABLED</b> ) or not ( <b>DISABLED</b> ) when the shell is globally installed.

For example, to check the versions of the NetBase programs:

```
:RUN NBVERS.PUB.NETBASE,PROG
```

```
NetBase Version Display [x.x] Update n (C) QUEST Software 1996
```

Program File:	NBCOP.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBCTRL.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBDIR.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBCOM.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSERVER.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NMSERVER.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSPOOL.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSPOOLC.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSPOOLE.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSPOOLI.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSPOOLT.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSPOOLR.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSTATCP.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSTATLG.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBEXPORT.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBIMPORT.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBPOST.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBILOGC.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBILOG.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSYNC.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBFLUSH.PUB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSQLCTL.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSQLEXP.NB.NETBASE	Version:	[v.u.f]	Upd nn
Program File:	NBSQLIMP.NB.NETBASE	Version:	[v.u.f]	Upd nn

## Enabling/Disabling NetBase via NBVERS

**NBVERS** may also be used to enable and disable NetBase for CM. NetBase automatically intercepts all native mode file and database intrinsics via the routines in **XL.PUB.SYS** with a global install. *In order to intercept all compatibility mode intrinsics, you must run **NBVERS.PUB.NETBASE,ENABLE**. The **ENABLE** entry point redirects the CM intrinsics to run through **RTMSL.PUB.SYS**, where the NetBase routines are located.*

In a shadowing environment, running **NBVERS,ENABLE** prior to any users running CM programs is critical and should be part of the system startup process. If CM programs are not being intercepted, their updates to databases will not be shadowed, and files will go out of sync. In an NFA environment, **NBVERS** is similarly important if you have multiple copies of a database and you want to direct certain programs to a remote copy versus a local one.

To intercept all subsequent CM programs (ones that are not currently running):

**:RUN NBVERS.PUB.NETBASE,ENABLE**

To dynamically disable NetBase from new CM processes enter:

**:RUN NBVERS.PUB.NETBASE,DISABLE**

To check to see if NetBase is enabled or disabled:

**:RUN NBVERS.PUB.NETBASE,STATE**



**Note.** System Manager (SM) capability is required to run **NBVERS** for the listing of program versions (**RUN NBVERS** or **RUN NBVERS,PROG**). Operator (OP) capability is required to run **NBVERS,ENABLE** or **NBVERS,DISABLE**.

---

## NBLOG

**NBLOG** displays the status and error messages NetBase reported to the **\$STDLIST** and the console for the time period specified. You can also use **NBLOG** with an alternate entry point to display a history of all changes made in the **NBDIR**, **NBCTRL**, **CONFIG** and **NBVERS** programs for the time period specified.

When the NetBase background job starts, the **NBLOG.DATA.NETBASE** file is copied to **NBLOGBK**, and the previous **NBLOGBK** file is purged. **NBLOG.DATA.NETBASE** contains the history of the NetBase job for the current session, since the background job was most recently started. **NBLOG.PUB.NETBASE** reads the log file **NBLOGBK** to display the desired results. **NBLOGBK** contains the history of the runs of the previous NetBase job.

To review error and status messages that were displayed on the console:

```
:RUN NBLOG.PUB.NETBASE  
NetBase Log Dump Facility [0.9.7] (C) QUEST Software 1987  
Enter Start Date <cr> for all: 08/15/96  
08/15/96 0:21:29.8 NBP009/NETBASE Shadow Posting for BOB STARTED  
08/15/96 0:21:30.0 NBP009/Post log 1 started  
08/15/96 0:23:44.8 NBN/COMMUNICATION READY TO NODE 9: BOB  
08/15/96 0:23:49.7 NBEXP/TRANSFERRING 16 SHADOW RECORDS TO BOB  
08/15/96 0:23:51.5 NBEXP/TRANSFER OF SHADOW RECORDS TO BOB COMPLETE  
08/15/96 0:23:52.1 NBEXP/SHADOW EXPORTING TO BOB ACTIVE  
08/15/96 0:24:02.7 NBI009/SHADOW IMPORTING FROM BOB ACTIVE  
08/15/96 0:24:05.1 NBSE009/SPOOLING TO BOB READY  
:
```

To see all changes to **NBDIR**, **NBCTRL**, **CONFIG** and **NBVERS**:

```

:RUN NBLOG.PUB.NETBASE,LOD
NetBase Log Dump Facility [x.x] (C) QUEST Software 1987
Enter Start Date <cr> for all: 08/15/94
08/15/96 10:26:35.3 Ctl #S1134 PHIL,MGR.NETBASE Pin 66 Dev 35
STOP NETBASE

08/15/96 10:30:46.5 Ver #S1134 RICK,MGR.NETBASE Pin 65 Dev 35
NetBase Access DISABLED

08/15/96 10:53:11.3 Cfg #S1134 BIFF,MGR.NETBASE Pin 115 Dev 35
M 2

08/15/96 10:53:18.2 Cfg #S1134 BIFF,MGR.NETBASE Pin 115 Dev 35
SHADOW - Direction (N/E/I/B)..... [B] N

08/15/96 10:53:38.6 Ctl #S1134 RICK,MGR.NETBASE Pin 110 Dev 35
START NETBASE

08/15/96 16:23:27.8 Dir #S1147 RICK,MGR.NETBASE Pin 124 Dev 35
FILE @.REDIRECT,BOB,DAFFY

:

```

**NBLOG** allows you to specify the program for which you want to see a history, as shown below.

```

:RUN NBLOG.PUB.NETBASE,LOD;PARM=n

```

The parm entry allows you to specify **NBCTRL**, **CONFIG**, **NBDIR** or **NBVERS**.

PARM=	
1	NBDIR
2	NBCTRL
3	CONFIG
4	NBVERS

To see the changes in **NBCTRL** only:

```
:RUN NBLOG.PUB.NETBASE,LOD;PARM=2
NetBase Log Dump Facility [0.9.6] (C) QUEST Software 1987
Enter Start Date <cr> for all: 08/15/94
08/15/96 10:21:28.8 Ctl #S1134 RICK,MGR.NETBASE      Pin 69   Dev  35
    START POST 9
08/15/96 10:26:35.3 Ctl #S1134 RICK,MGR.NETBASE      Pin 66   Dev  35
    STOP POST
08/15/96 10:36:50.6 Ctl #S1134 RICK,MGR.NETBASE      Pin 136  Dev  35
    START POST 9
08/15/96 10:47:37.2 Ctl #S1134 RICK,MGR.NETBASE      Pin 135  Dev  35
    ABORT NETBASE
08/15/96 10:53:38.6 Ctl #S1134 RICK,MGR.NETBASE      Pin 110  Dev  35
    START NETBASE
08/15/96 10:54:24.8 Ctl #S1134 RICK,MGR.NETBASE      Pin 108  Dev  35
    START POST BOB
08/15/96 10:55:31.9 Ctl #S1134 RICK,MGR.NETBASE      Pin 92   Dev  35
    START POST BOB
08/15/96 11:12:36.7 Ctl #S1142 ROBIN,MGR.NETBASE     Pin 75   Dev  33
    START CLIENT
08/15/96 16:31:29.7 Ctl #S1147 RICK,MGR.NETBASE      Pin 128  Dev  35
    ROLL FORWARD TEST.REDIRECT
08/15/96 16:31:43.5 Ctl #S1147 RICK,MGR.NETBASE      Pin 55   Dev  35
    STOP POST
END OF PROGRAM
```

## NBTRACE

**NBTRACE.PUB.NETBASE** records an application's activity from NetBase's viewpoint. With **NBTRACE** you can see which calls are being intercepted by the NetBase shell, which ones are successful, and which ones fail. **NBTRACE** can help identify the exact file not found by identifying which **FOPEN** failed and what it was attempting to open.

**NBTRACE** is enabled by setting a JCW for the session. It logs its results to **NBAT###.IPC.NETBASE** files, where **###** is the process number. Most file system and all **IMAGE** intrinsics intercepted by NetBase are recorded. If you perform a trace on a CM program, and none of its calls are in the resulting trace, NetBase may not be enabled. To verify, run **NBVERS.PUB.NETBASE,STATE**.

### To use NBTRACE:

1. To start the trace, set a JCW:

```
:SETJCW NBDISABLE=WARN
```

2. Run your application.

3. Reset the JCW as follows:

```
:SETJCW NBDISABLE=0
```

4. Run **NBTRACE** to convert the results into a readable format. The results are in a message file, so you may want to redirect the results to a spool file in order to review the information more than once.

```
:FILE QUEST;DEV=LP,1  
:RUN NBTRACE.PUB.NETBASE,USER;PARM=1 >*QUEST
```

The trace files are **NBAT###.IPC.NETBASE** (five files). If the program that is being traced creates a son process, then **NBAT0001** contains the father process and **NBAT0002** contains the trace for the son process. The **;PARM=#** indicates the trace file to view.

5. Use NBSpool to view your new spool file which contains the trace results.

```
:RUN NBSPOOL.PUB.NETBASE
O> T #*
O> L |FOPEN| (or |DBOPEN|)
```

This command (either **FOPEN** or **DBOPEN**) lists all the lines where a file open occurs. The following is an example of a trace of an inactive user process using the **DBOPEN** option.

```
TRACE OF INACTIVE USER PROCESS STARTED
WED, JAN 26, 1994 Prog: QUERY.PUB.SYS User: MGR.PUB.NETBASE
14:53:46.6 1001 BANNER.DATA.NETBASE DBOPEN = 0 1 #0
Field: 1 2 3 4 5 6 7
```

### Fields Definitions

The following table lists definitions of the fields.

Definitions
<b>1 = Time stamp</b>
<b>2 = File number or Base id</b>
<b>3 = File name or Database name</b>
<b>4 = Intrinsic called</b>
<b>5 = Status</b>
<b>6 = Database Open Mode</b>
<b>7 = Dataset # (if applicable)</b>

## QUARK

This program is an archiving program similar to the ones available for PCs. It takes a set of files and stores them in a compressed format in one file. A suggested use for **QUARK** is for copying files between boxes.

```
:RUN QUARK.LIB.NETBASE;INFO="cmd archive [fileset ...]"
```

where cmd is:

```
  A:  Add files to archive (replace if present)
 AP:  Add files to archive, purge original
  X:  Extract files from archive
 XL:  Extract files, put in local group
 XG:  Extract files, original groups in login account
 XP:  Extract files, purge existing copies
 XLP: Extract files, put in local group and purge existing copies
  D:  Delete files from archive
  R:  Replace files in archive
  L:  List archive contents
```

File equations will be used if they are fully qualified.

---



**Note.** **QUARK** does not support KSAM files, files greater than 60,000 sectors, or indirect files.

---

### Examples

This example adds the file **NEWVERSN.PUB.PROD**:

```
:RUN QUARK.LIB;INRO:"A purgeme NEWVERSN.PUB.PROD"
```

This example adds the file **NEWVERSN.PUB.PROD** to the quark file named **PURGEME**. This creates the quark file, if it did not exist.

```
:HELLO MGR.NETBASE,LIB
:RUN QUARK;INFO="A PURGEME NEWVERSN.PUB.PROD"
```

This example extracts the files contained in the quark file **PURGEME** locally. **NEWVERSN.PUB.PROD** is extracted as **NEWVERSN.TEST.BACKUP**.

```
:HELLO MGR.BACKUP,TEST
:COPY PURGEME.LIB.NETBASE,PURGEME
:RUN QUARK.LIB.NETBASE;INFO="XL PURGEME"
```

## Additional Utilities

NetBase now additionally offers the **QUARK+** and **NBCOPY** utilities.



**Note.** The **NBCOPY** and **QUARK+** utilities must be purchased separately.

---

### QUARK+

**QUARK+** is an enhanced version of the **QUARK** utility. It works like **QUARK**, but has additional functionality. The following example adds all files in the **JOB** group to the **PURGEME.PUB.PROD** quark file.

```
:HELLO MGR.PROD, JOB
:COPY PURGEME.LIB.NETBASE, PURGEME.PUB
:RUN QUARK.LIB.NETBASE;INFO="A PURGEME.PUB @.JOB"
```

This example copies the quark file to another system, and extracts just the files that qualify as **@.JOB.PROD** into the group **JOB** of the local account **OPER**. By specifying the **P** in **XP**, you can instruct **QUARK** to purge any existing copies and replace them with ones from the quark file.

```
:REMOTE :SYSB
SYSB: HELLO MGR.OPER, PUB
SYSB: DSCOPY PURGEME.PUB.PROD:SYSA
SYSB: QUARK.LIB.NETBASE;INFO="XP -GJOB PURGEME @.JOB.PROD"
```



**Note.** When **QUARK** fails, it sets a JCW named **CJCW** to 1.

---

### QUARK+ Features

The following section lists **QUARK+** features.

- **QUARK+** handles very large files (>60000 sectors), unlike **QUARK** which often gets "crc" errors when unquarking big files.
- Handles every kind of file (except **SPOOL** files).
- Has both **QUARK** and **QUARK9** integrated into one program.
- Stores and restores accounting structure, passwords, capabilities etc.
- Runs only on 4.0 and later releases of MPE/iX.

- Supports **fileset-fileset-fileset-...** like **STORE/RESTORE**.
- Stores lockwords and **:ALTSEC** security information. Note: **QUARK+** will not prompt for a lockword if you have **SM** or **AM** capability and it is in your account.
- Allows redirection to various groups and accounts, like **GROUP=** and **ACCT=**.
- Can unquark across accounts. If you unquark a database, a **NONB** record is required for **QUARK.PUB.NETBASE**.
- Locks and compresses files open for lock access.
- Supports files created by the older **QUARK**.

## Syntax

```
:RUN QUARK.PUB.NETBASE;INFO="cmd [opts] archive [fileset ...]"
```

where **cmd** is:

```
A:      Add files to archive (replace if present)
AD:     Add files and store accounting structure
AL:     Add files and lock if file is in use
AO:     Add files without changing last access time
AP:     Add files to archive, purge original
AF:     Use faster compression method
AQ:     Use quickest compression method.
AC:     Enable CRC calculation (or :SETVAR QUARKCRC "Y")
X:      Extract files from archive
XO:     Preserves the creation/access/modification time stamps for a file
XD:     Extract files and accounting structure
XL:     Extract files, put in local group
XG:     Extract files, original groups in login account
XP:     Extract files, purge existing copies
XV:     Extract files and accounting structure on volumes
```

The following table lists what **opts** can be specified.

Parameter	Description
<b>-Ggname</b>	Extract to group <b>gname</b>
<b>-Aaname</b>	Extract to account <b>aname</b>
<b>-Cuname</b>	Make creator name <b>uname</b>
<b>-Vvname</b>	Make volume set <b>vname</b> (requires XV)

Also:

Parameter	Description
<b>D</b>	Delete files from archive
<b>R</b>	Replace files in archive
<b>L</b>	List archive contents

## Notes

- The **AQ** option is 8-15 times faster than standard compression, but does not compress more than 88% of a file.
- The **XO** feature requires version A.03.00 of QUARK+. Older archive files are compatible with this version, however, the time stamps will not work unless the archive was created with version A.03.00 or later.
- A maximum of 20 filesets may be specified.
- Use **-filesets** to exclude files.
- A file set of **^filename** specifies an indirect file.
- Compression speeds vary based upon the processor used and the type of data being compressed.

## Example

The following example shows how to use **QUARK+** to compress an IMAGE database and then uncompress it preserving the original timestamps on the file label.

```
RUN QUARK.PUB;info="A SAMPLE BANNER@.DATA.NETBASE"
```

```
Quest Archiver+ A.03.05b (C) QUEST Software 1994
```

```
Licensed to: Quest Software
```

```
Adding BANNER.DATA.NETBASE... 69.3%, 118 msec
Adding BANNER01.DATA.NETBASE... 92.5%, 25 msec
Adding BANNER02.DATA.NETBASE... 91.5%, 44 msec
Adding BANNER03.DATA.NETBASE... 79.3%, 24 msec
Adding BANNER04.DATA.NETBASE... 99.4%, 1014 msec
Adding BANNER05.DATA.NETBASE... 93.0%, 21 msec
Adding BANNER06.DATA.NETBASE... 96.5%, 32 msec
Adding BANNER07.DATA.NETBASE... 95.8%, 71 msec
```

## Chapter 13

---

```
Adding BANNER08.DATA.NETBASE... 89.3%, 31 msec
Adding BANNER09.DATA.NETBASE... 99.7%, 121 msec
    10 files, 0 failed, 97.9% saved
```

END OF PROGRAM

```
:LISTF SAMPLE,2
```

```
ACCOUNT= NETBASE      GROUP= PUB
```

FILENAME	CODE	-----LOGICAL RECORD-----				----SPACE----			
		SIZE	TYP	EOF	LIMIT R/B	SECTORS	#X	MX	
SAMPLE	707	128W	FB	30	30	1	32	1	1

The XLO option tells **QUARK+** to unarchive the files into the **Local** group and to preserve the **Olddate** information:

```
:RUN QUARK.PUB;INFO="XLO SAMPLE.PUB"
```

Quest Archiver+ A.03.05b (C) QUEST Software 1994

Licensed to: Quest Software

```
Extracting BANNER.DATA.NETBASE as BANNER.TEST.NETBASE...
Extracting BANNER01.DATA.NETBASE as BANNER01.TEST.NETBASE...
Extracting BANNER02.DATA.NETBASE as BANNER02.TEST.NETBASE...
Extracting BANNER03.DATA.NETBASE as BANNER03.TEST.NETBASE...
Extracting BANNER04.DATA.NETBASE as BANNER04.TEST.NETBASE...
Extracting BANNER05.DATA.NETBASE as BANNER05.TEST.NETBASE...
Extracting BANNER06.DATA.NETBASE as BANNER06.TEST.NETBASE...
Extracting BANNER07.DATA.NETBASE as BANNER07.TEST.NETBASE...
Extracting BANNER08.DATA.NETBASE as BANNER08.TEST.NETBASE...
Extracting BANNER09.DATA.NETBASE as BANNER09.TEST.NETBASE...
    10 files
```

END OF PROGRAM

---

## NBCOPY

**NBCOPY** enables you to copy files easily, making exact duplicates on the same or remote systems without the need for remote sessions. Since **NBCOPY** has been optimized to perform this one function, it copies files many times faster than HP's **FCOPY** or **DSCOPY**. The **NBCOPY** utility is flexible in the types of files it handles, and it offers several options for modifying the new copy during the copying process.



**Note.** **NBCOPY** is **COPYN** with extensions to allow copying between systems using the NetBase NFA facility. **NBCOPY** is used to copy files between systems when you do not have HP's **NS** product (**:DSCOPY**).

---

### NBCOPY Features

- Making exact copies of disk files (with or without the user labels). KSAM files can be duplicated.
- Extracting data from a KSAM file at very high speed while removing deleted records (similar to the **NOKSAM** parameter of **FCOPY**).
- Reloading KSAM files at anywhere from 30-100% faster than **FCOPY**.
- Copying files, data sets, or databases across a network using NetBase. This function does not require remote logons and is much faster than **DSCOPY**.
- Converting **EBCDIC** to ASCII (and vice versa) with high speed and low overhead during any copy operation.
- Reblocking files. You can change wasteful blocking factors for files in a fraction of the time. In fact, **NBCOPY** will even reblock records. For example, you can copy one large record into two records half as big, or many little records into one large one.
- Changing the limit of an MPE or KSAM file with one command. The program handles duplicating, copying, purging and renaming.
- Copying to and from tape. When writing to tape, you can reblock the data to records of up to 8,192 words.
- Extracting entries from an IMAGE data set at a rate equivalent of up to 4,000 **DBGETS** per second. The extraction is always at least 5 times faster than serial **DBGETS**.
- Loading entries from a file or a data set into another data set. This allows copying a data set in one base to the same set in another database.
- Changing the blocking factor for a file with one command. If desired, the optimal blocking factor will be calculated.
- Loading records into a KSAM file or IMAGE master set and ignoring any duplicate key errors.
- A **REDO** feature to allow syntax errors to be easily corrected.

## NBCOPY Syntax

The syntax for **NBCOPY** closely approximates the **FCOPY** syntax. As a result, it can be easily inserted into job streams and UDCs where it can replace **FCOPY** commands. The following table lists the **NBCOPY** parameters and what they do.

### Parameters

Parameter	Description
<b>FROM=</b>	<p>Specifies the file to copy from (input file). If this parameter is not specified at all, the default <b>FROM</b> file is <b>*QCINPUT</b>. Note that this back-references a file equation, not a file name. If no file name is specified, <b>\$STDIN</b> is assumed.</p> <p>If the <b>FROM</b> file is a KSAM file, it will be copied with <b>NOKSAM</b> access unless the <b>TO</b> file is a KSAM file as well. Deleted records are not copied.</p> <p>If you are extracting from a data set, the database name is specified with the <b>FROM=</b> parameter. The syntax for specifying a database is:</p> <p><b>FROM=</b><i>basename</i> [ , <i>password</i> ] , <i>mode</i></p> <p>Where <i>basename</i> is the root file name, <i>password</i> is a valid password for the base, and <i>mode</i> is one of the read modes of IMAGE, namely 5, 6, 7 or 8. If no password is specified, you must be the creator of the database. If you omit the password and you are not the creator, <b>NBCOPY</b> will ask for the password.</p> <p>If you are copying between machines, NetBase must be running on both machines. You may specify the nodes either by number or name. The syntax for specifying nodes is:</p> <p><b>FROM=</b><i>filename</i> [ : <i>node</i> ]</p>
<b>FROMSET=</b>	<p>Specifies the data set from which records will be read. Of course a database must be specified with the <b>FROM=</b> parameter if you specify a <b>FROMSET</b>.</p>

*Continued*

Parameter	Description
<b>TO=</b>	<p>Specifies the file to which to copy (output file). If omitted, this parameter defaults to <b>*QCOUTPUT</b>. If specified without a file name, <b>\$STDLIST</b> is assumed.</p> <p>If the specified file exists, it is erased unless a file equation is issued specifying <b>ACC=OUTKEEP</b>. Data sets are never erased.</p> <p>If the <b>TO</b> file does not exist, it will be built as a copy of the <b>FROM</b> file, with the same limit, extents and user labels. If the <b>FROM</b> file is a KSAM file, the new <b>TO</b> file will not be a KSAM file unless the <b>KEYFILE=</b> parameter is also specified (see below).</p>
<b>TO= (continued)</b>	<p>When the <b>TO</b> file is a data set, you must specify the database with the <b>TO=</b> parameter. The format is the same as the <b>FROM=</b> database specification:</p> <p><b>TO=basename[ ,password] ,mode</b></p> <p>The mode must be 1, 3 or 4. If you specify mode 1, the <b>TOSET</b> will be locked while all entries in an input block are being put into the set. If the input blocking factor is large, the data set could be locked for quite a long time while the puts are processed.</p> <p>Mode 3 is by far the fastest mode to use, but it does have a potential problem. Using the exclusive write mode causes <b>NBCOPY</b> to turn off posting for the data set with the <b>DBCONTROL</b> procedure. If the system should fail while the entries are being put, the database will be flagged as bad. You must weigh the benefits of really high speed against the odds of the system crashing when deciding the mode.</p> <p>If you are copying to another machine, NetBase must be running on both machines. The node may be specified either by name or number. The syntax for specifying a node is:</p> <p><b>TO=database[ :node] [ ,password] ,mode</b></p> <p>When specifying a node, the database (or file) is sent to the same group and account on the receiving machine if none are specified. To specify a different location, use this syntax:</p> <p><b>FROM=filename.group.acct;TO=filename.diffgroup.diffacct:node</b></p>
<b>TOSET=</b>	<p>Specifies the data set into which records will be added (with <b>DBPUT</b>). A <b>TO</b> database must be specified with the <b>TO=</b> parameter, prior to this parameter.</p>

Continued

Parameter	Description
<b>BF=</b>	Specifies the blocking factor of the <b>TO</b> file. If no value is specified, a blocking factor will be computed. If no <b>TO</b> file is specified, the <b>FROM</b> file will be rebuilt with the specified blocking factor. See the <b>LIMIT</b> parameter for more information on this feature.
<b>COPYBASE</b>	Copies an entire database. Database specifications do not need passwords and modes. <b>FROM=database[ :node ] ; TO=newdatabase[ :node ] ; COPYBASE</b>
<b>EBCDICIN</b>	Translates all records in the <b>FROM</b> file to ASCII before writing them to the <b>TO</b> file. This option only works if your computer has the COBOLII firmware, which includes every HP machine made after the Series III.
<b>EBCDICOUT</b>	Translates all records to <b>EBCDIC</b> before writing them to the <b>TO</b> file. Again, this only works if your machine has the COBOLII firmware.
<b>IGNERR</b>	Ignores any file system or IMAGE errors, primarily KSAM write and <b>DBPUT</b> errors, when writing to the <b>TO</b> file or <b>TOSET</b> . This prevents a <b>COPY</b> from stopping if a duplicate key error is detected. The number of ignored records is displayed at the end of the copy.
<b>KEY=byte</b>	Like the <b>KEY=</b> parameter of <b>FCOPY</b> , this causes <b>NBCOPY</b> to read <b>FROM</b> file records in key sequence. Unfortunately, the copy will run much slower since the key file must be consulted for key sequence. The <i>byte</i> specification identifies the byte offset of the key to be used for the sorted read.
<b>KEYFILE=</b>	Specifies the name of the keyfile to be built when copying from a KSAM file to a new KSAM file. This parameter is invalid unless the <b>FROM</b> file is KSAM. If no file name is specified, the keyfile name used will be the <b>TO</b> file name with a <b>K</b> appended. If the <b>TO</b> file name is 8 characters, the last character will be changed to a <b>K</b> . If the last character is already a <b>K</b> , specify a keyfile name.

*Continued*

Parameter	Description
<b>LIMIT=</b>	<p>Specifies the file limit of the <b>TO</b> file. This assumes the <b>TO</b> file is a new file. If the <b>TO</b> file is not specified, the <b>FROM</b> file is copied to a temporary file of the same name, purged, and the temporary file is saved as the new <b>FROM</b> file. The net effect is to change the limit of the <b>FROM</b> file. For example, to change the limit of the file <b>NAMES.DATA</b> to 5000, you would enter:</p> <p><u>&gt;FROM=NAMES . DATA ;LIMIT=5000</u></p> <p>If the limit you specify is less than the current EOF, the copy will stop when the file fills up, and some records will be lost. However, the original file will be left untouched since it only is purged after a successful copy. The temporary file will contain the partially completed copy.</p>
<b>NEW</b>	<p>Specifies that the <b>TO</b> file cannot already exist. If the <b>TO</b> file exists, the copy will not proceed. This protects existing <b>TO</b> files from being obliterated.</p>
<b>NOLABELS</b>	<p>Prevents user labels from being copied to the <b>TO</b> file.</p>
<b>OLD</b>	<p>Specifies that the <b>TO</b> file must exist. This prevents <b>NBCOPY</b> from creating a new <b>TO</b> file.</p>
<b>REBLOCK</b>	<p>If the <b>FROM</b> file and <b>TO</b> file do not have the same record size, the copy will not occur unless the <b>REBLOCK</b> option is specified. This is discussed in the limitations section.</p>
<b>RELEASE</b>	<p>When the current copy operation completes, the <b>TO</b> file limit is set to the current EOF.</p>
<b>SKIPEOF=</b>	<p>This has the same format as the <b>FCOPY</b> parameter of the same name. It is useful for skipping past files on a tape. The full syntax is:</p> <p><b>SKIPEOF=[ from-eofs ] [ , to-eofs ]</b></p> <p>If this parameter is specified for a non-tape file, it is ignored.</p>
<b>START=</b>	<p>Specifies the first record to be copied from the <b>FROM</b> file. The first record of a file is considered to be record 0.</p>
<b>STATS</b>	<p>Displays a block of status information after the copy is complete. This includes the number of I/Os, records transferred, and records transferred per second.</p>

When you specify any of the above parameters, a copy operation will begin. If you have set up file equations for both **QCINPUT** and **QCOUTPUT**, you can just specify **>STATS** to begin a copy. As the copy is performed, you may hit **Ctrl-Y** to see how far along the copy is.

## Running NBCOPY

To start **NBCOPY**, run the program, **NBCOPY.PUB.NETBASE**. You will see the standard utility prompt, the greater than sign (>). One or more copy commands can be entered. Each command will be processed in turn unless **NBCOPY** is run from a job and an error is detected. In this case, all other commands are flushed.

At the prompt, you must enter either a copy command, **HELP**, **REDO**, **QUIT** or **EXIT**. The **HELP** command displays a quick reference chart for command syntax. The **REDO** command repeats the last command entered, allowing you to edit it with standard line-editor commands before executing it.

If **NBCOPY** is run programmatically, it will suspend rather than terminate when **EXIT** is entered. This allows an application program to quickly activate it when it is needed again. To cause it to terminate, use the **QUIT** command. Commands may be continued on subsequent lines if the **&** character is specified at the end of a command line.

Like **FCOPY**, **NBCOPY** can use the **INFO=** parameter of the run command. Therefore, you can create a UDC for copies, such as:

```
NBCOPY FROMF,TOF
RUN NBCOPY.PUB.NETBASE;INFO="FROM=!FROMF; TO=!TOF; STATS"
*****
EXPAND FILE,LIMIT
RUN NBCOPY.PUB.NETBASE;INFO="FROM=!FILE; LIMIT=!LIMIT; STATS"
*****
DUMPSET BASE,SET,MODE=5,FILE
RUN NBCOPY.PUB.NETBASE;&
INFO="FROM=!BASE,,!MODE; FROMSET=!SET; TO=!FILE; STATS"
*****
```

If the **INFO=** parameter is supplied, command input will be suppressed, so only one command can be executed. If an error occurs, the JCW will be set to **FATAL** plus the number of the error that occurred.



**Note.** If NetBase is not installed, or if its processes are not running, the following message appears when you try to run **NBCOPY**:

```
ERROR 126: This program must be run through NetBase.
```

To correct it, verify that NetBase is installed and running. If you are copying from one machine to another, and NetBase is not running on both machines, you may receive the following message:

```
ERROR: REQUESTED NODE IS DOWN.
```

To correct it, start NetBase on the machine for which you specified a node name or number.

---

## Limitations

In order to copy files as fast as it does, **NBCOPY** does not really pay too much attention to records. In most copy operations, **NBCOPY** copies blocks of records, not records. As a result, **NBCOPY** does not really work as well when copying to or from undefined record length files. In other words, if you simply want to dump a file to your terminal, or key in data to a disk file, use **FCOPY** instead.

Variable files can be copied, but not to a fixed length file. You will get unusual results if you do. Appending one file to another does not work unless the last block of the **TO** file is completely full. If it is not, you will find a few blank or zero records filling the (formerly) last block of the **TO** file.

**FCOPY** is also a must if you want to copy a file with small records to one with larger records. With **FCOPY**, records are copied one by one, so each record is filled with blanks or binary zeroes when it is written to the new file. In the same situation, **NBCOPY** would reblock as many small records as necessary to make one larger record. For example, if you copy from a file with 100 word records to a file with 150 word records, it would take three input records, concatenate them, and write out two output records. It would look something like this:

```
: FILE NEWFILE;REC=150,,F,ASCII
> FROM=OLDFILE;TO=NEWFILE;REBLOCK
```

Record 1	Record 2	Record 3
100	100	100

would become:

Record 1	Record 2
150	150

Use **FCOPY**, unless you want this result when increasing record sizes. **NBCOPY** cannot copy **NM** **KSAM** files.

## Examples

Copy the file **HISTORY.DATA** to the file **BACKUP.DATA**. In this example, **BACKUP.DATA** does not necessarily have to exist. If it does, it will be erased (EOF set to 0) before the copy begins.

```
: RUN NBCOPY.PUB.NETBASE
NBCOPY [x.x] (C) QUEST Software 1996
> FROM=HISTORY.DATA;TO=BACKUP.DATA
  {Statistics are printed at the end of the copy}
> EXIT
:
```

Copy the KSAM file **COPYLIB.SOURCE** to the file **COPYX**. If **COPYX** does not exist, it will be built exactly like the data file for **COPYLIB.SOURCE**, but it will not be KSAM.

```
> FROM=COPYLIB.SOURCE;TO=COPYX
```



**Note.** **COPYX** records will be in the chronological sequence of the file **COPYLIB.SOURCE**. Any deleted records in **COPYLIB.SOURCE** will be ignored.

---

Copy the KSAM file **COPYLIB.SOURCE** to the new KSAM file **NBNEW.SOURCE**. The new keyfile name will be **NBNEWK.SOURCE**.

```
> FROM=COPYLIB.SOURCE;TO=NBNEW.SOURCE;KEYFILE= NBNEWK
```

Copy the file **BADBLOCK** to the new file **OKBLOCK**. The blocking factor on **OKBLOCK** is being changed to 16.

```
> FROM=BADBLOCK;TO=OKBLOCK;BF=16
```

Copy the file **TOIBM.PUB.CONV** to the file **TOIBM.DATA** and convert the data to **EBCDIC**. In addition, release any unused space in **TOIBM.DATA** when done.

```
> FROM=TOIBM.PUB.CONV;TO=TOIBM.DATA;EBCDICOUT;RELEASE
```

Change the limit on the file, **TOOSMALL.RBM**, from whatever it is now to 10,000.

```
> FROM=TOOSMALL.RBM;LIMIT=10000
```

Extract every entry in the data set **CUSTOMERS** from the database **AR** and put it into the new file **CUSTFILE**.

```
> FROM=AR, , 5;FROMSET=CUSTOMERS;TO=CUSTFILE
```

Put records in the file **CUSTFILE** into the data set **OLD-CUSTS** of the **HIST** database:

```
> FROM=CUSTFILE;TO=HIST, , 1;TOSET=OLD-CUSTS
```

To reblock and expand the file **BIGFILE**.

```
> FROM=BIGFILE;LIMIT=100000;BF=
```

To expand the KSAM file **NAMELOOK** and ignore any duplicate keys.

```
>FROM=NAMELOOK;LIMIT=50000;IGNERR
```

Occasionally, a KSAM file is damaged internally, resulting in duplicate keys even though duplicate keys are not allowed. **NBCOPY** can correct this problem.

```
> FROM=NAMELOOK
```

Copy the file, **USERFILE** on **DAFFY** (nodename) to the **USERS** data set in the **BIGBASE** database on **SILVER** (nodename).

```
> FROM=USERFILE:DAFFY;TO=BIGBASE:SILVER,PRESIDENT,3;TOSET=USERS
```

Copy the file, **STOREME**, to tape and reblock to minimize wasted space.

```
: FILE TAPE;DEV=TAPE  
> FROM=STOREME.DATA.BACKACCT;TO=*TAPE;REBLOCK
```

Copy the database, **CUSTBASE** on **SystemA** to the **HISTORY** database on **SystemB**.

```
> FROM=CUSTBASE:SYSTEMA;TO=HISTORY:SYSTEMB;COPYBASE
```

## A P P E N D I X A

# Installation/Update Procedures For NetBase

## Overview

Typically, the installation and configuration of each system or the updating of software can be accomplished in less than 30 minutes. In general, the following steps comprise the installation/update process:

1. **Restore the NetBase software from tape using NETACCT job.**
2. **Configure operational parameters and define all nodes in the network.** Note that this step is required for initial implementation only. Updates will not overwrite this information.
3. **Install the NetBase shell routines and restart the system.**
4. **Validate the NetBase installation and modify system startup procedures.**

## Before Installing NetBase

Before installing NetBase, determine which of the following scenarios describes your system, and then take the action listed for that scenario.

- If you are updating from a NetBase 9.6 global installation, proceed to the “Updating From NetBase 9.6 Global Install” section below.
- If you are updating from a version of NetBase that is prior to 9.6, or converting from an account installation, turn to the “Installation Appendix” section at the end of this chapter before performing the installation/update.
- If you are installing NetBase for the first time, proceed to **Step 1** on the following page.

If you have any questions or if you are unsure about any of the steps, please contact Technical Support at (714)-720-1434.

## Updating From NetBase 9.6 Global Install

If you are updating from a previous version of NetBase 9.6 that is globally installed, proceed to the following steps *before* you update.

\_\_\_\_\_ **A.** Stop all sessions and jobs except for NetBase.

\_\_\_\_\_ **B.** If shadowing, flush shadow transactions:

```
:RUN NBFLUSH.PUB.NETBASE,STOPWAIT;PARM=shadownode#
```

\_\_\_\_\_ **C.** Stop the NetBase background job:

```
:RUN NBCTRL.PUB.NETBASE;INFO="STOP NETBASE"
```

\_\_\_\_\_ **D.** Disable the CM intercepts for NetBase:

```
:RUN NBVERS.PUB.NETBASE,DISABLE
```

\_\_\_\_\_ **E.** If the **NMCMUDC** is set, delete it:

```
:SETCATALOG NMCMUDC.PUB.NETBASE;SYSTEM;DELETE
```

\_\_\_\_\_ **F.** Back up the NetBase account to tape.

## Installation Steps

Once you have read the “Before Installing NetBase” section, take steps 1 through 4 to install NetBase. Notice that the “Before Starting” section in **Step 1** describes two system scenarios with actions to take.

### Step 1. Restore NETACCT job and STREAM NETACCT

To create NetBase's environment and install/update the software, you will restore a job into **PUB.SYS**. This job is then streamed to create/update the NetBase account and restore the required files.

#### Before Starting

Before doing **Step 1**, determine if any of the following scenarios describes your system, and if so, take the action for that scenario.

- If you use VESOFT's **STREAMX** utility, do *one* of the following:
  - Disable **STREAMX** and use the MPE **STREAM** command
  - Modify the installation job and change each **STREAM** command to **STREAMX**
- If you use SECURITY/3000, enter this command *prior* to streaming the installation job:  
**:NEWACCT NETBASE,MGR;PASS=QUEST**
- If you have a lockword on **QUERY.PUB.SYS**, remove it prior to streaming **NETACCT**.

#### To do Step 1:

- \_\_\_\_\_ 1.1. Sign on to **MANAGER.SYS** and restore the job stream file:



**Note.** Characters in << >> marks are comments and are not entered or displayed. User input is underlined and bold.

---

```
:HELLO MANAGER.SYS,PUB  
:RESTORE ;NETACCT.JOB.@"LOCAL  
  
<<reply to the tape request>>
```

- \_\_\_\_\_ 1.2 If necessary, add any required passwords to the **NETACCT.PUB.SYS** file. This job may stream additional jobs depending on your environment, so the job limit must allow for at least two NetBase jobs. To stream the job:

```
:STREAM NETACCT
```

```
<<reply to the tape request>>
```

This job will request the tape and print a message to the console when complete. If the job encounters any errors, a message telling what action is necessary will be displayed to the console.



**Note.** If you are using private volumes, we require that you install Netbase on the system volume set, and not a user volume set. If however, you are concerned about the amount of free disc space on the system volume set, then you can move the **LOG** and **QUEUE** groups onto a user volume set.

---

## Step 2. Configure Network Nodes for NetBase

---



**Note.** If you are updating to a new version of NetBase, skip ahead to **Step 3**. Updating will not overwrite your current configuration files.

---

In order for NetBase to communicate to the other systems on the network, all systems must be defined to each other. Additionally, the operating parameters and limits of each system must be defined. This is accomplished using the **CONFIG** subsystem in **NBCTRL**. All configuration must be performed from the NetBase account. Some of the operating parameters should be configured before running NetBase for the first time. Others may or may not need to be configured, depending on the system. To get started, you will need to do the following:

- \_\_\_\_\_ A. Assign a *node number* for each system being configured. Node numbers can be assigned arbitrarily, however, once a number is assigned to a particular node, that number *must* be used consistently for that node throughout the configuration for *all* nodes.
- \_\_\_\_\_ B. Assign a **System Name**— it can be any name up to 8 characters long. This name will be used in NetBase displays, reports, and commands. It is the internal NetBase name.

## Example

We have two systems we want to configure for NetBase. We will assign node numbers **3** and **4** and use their network node names for the system name:

<u>Node #</u>	<u>System Name</u>
<b>3</b>	<b>FOX</b>
<b>4</b>	<b>VIXEN</b>

After the node numbers and names have been assigned, proceed to step 2.1 where a **modify quick** and **add node** commands must be performed.

### To do Step 2:

\_\_\_\_\_ 2.1 Modify the parameters by entering the **CONFIG** subsystem in **NBCTRL** as follows:

```
:HELLO MGR.NETBASE
:RUN NBCTRL.PUB.NETBASE

NetBase Control Program [v.u.f] Update n (C) QUEST Software
1987

*> CONFIG
C> MODIFY QUICK

GLOBAL - Local Node Number..... [1]      3
GLOBAL - Maximum Local Users..... [0]     128
NFA - Maximum Remote Users..... [0]      256
NFA - Initial Server Allocation... [0]     64
NFA - Server Initial Priority..... [CS]
SPOOL - Use S' and J' on incoming.. [Y]
DIR - Allow File Equates..... [N]
DIR - Allow Wildcards..... [Y]
SHADOW - Flush Shadow File Posts... [N]
PASSWORD - User password.....
PASSWORD - Group password.....
PASSWORD - Account password.....
```

The password parameters apply to the job that **NBCTRL** streams to start NetBase. The default passwords are correct for the account, group and user as they are created by **NETACCT.JOB**. If the default passwords are changed, the password parameters (shown below) must be modified.

In this example, we are setting the **Local Node Number = 3**. This node will be configured as **Node 3** on all systems. The values that appear in brackets [ ] are the defaults for each of the parameters.

\_\_\_\_\_ 2.2 Enter each system in the network into the configuration using the **ADD [Node#]** command. This includes the local system being configured.

```

:HELLO MGR.NETBASE
:RUN NBCTRL.PUB.NETBASE

NetBase Control Program [v.u.f] Update n (C) QUEST Software
1987

*> CONFIG
C> ADD 3

System Name..... FOX
Startup Access (S/I/O/A/W)..... [A]
Node Name..... FOXTROT.QUEST.COM
Node Connection Timeout..... [180]
SHADOW - Direction (N/E/I/B)..... [N]
SHADOW - Export queue size..... [3172]
SHADOW - Import queue size..... [3172]
SHADOW - Post queue size..... [3172]
SHADOW - Initial Post Priority..... [DEFAULT]
SHADOW - Post Inactivity Timeout... [0]
SHADOW - Enable Post Logging..... [N]
SHADOW - Log all Post Transactions. [N]
SHADOW - Save Process Info on Post. [N]
SHADOW - Abort Post on Errors..... [N]
SHADOW - Buffered Exporting..... [Y]
SHADOW - Transfer mode only..... [N]
SHADOW - Transfer Single Queues... [N]
SHADOW - Transfer ACK Frequency... [0]
SHADOW - Post Directory Filename... [DEFAULT]

```

In the above example, node **FOXTROT** is the local node and has now been assigned node number 3. It must continue to be defined as node 3 in the node configuration on each system in the network.

```

C> ADD 4

System Name..... VIXEN
Startup Access (S/I/O/A/W)..... [A]
Node Name..... VIXEN.QUEST.COM
Node Connection Timeout..... [180]
SPOOL - Direction (N/E/I/B).. .... [N] B
SPOOL - Allow incoming job streams. [N]
SPOOL - Enable Export Flow Control. [N]
SHADOW - Direction (N/E/I/B)..... [N] B
SHADOW - Export queue size..... [3172]
SHADOW - Import queue size..... [3172]
SHADOW - Post queue size..... [3172]
SHADOW - Initial Post Priority..... [DEFAULT]

```

```

SHADOW - Post Inactivity Timeout... [0]
SHADOW - Enable Post Logging..... [N]
SHADOW - Log all Post Transactions. [N]
SHADOW - Save Process Info on Post. [N]
SHADOW - Abort Post on Errors..... [N]
SHADOW - Buffered Exporting..... [Y]
SHADOW - Transfer mode only..... [N]
SHADOW - Transfer Single Queues... [N]
SHADOW - Transfer ACK Frequency... [0]
SHADOW - Post Directory Filename... [DEFAULT]

```

This procedure is repeated for each system in the network to be accessed by NetBase. The **LIST** command can be used to display the configured nodes.

\_\_\_\_\_ 2.3 Check the configuration using the **LIST NODES** command in **CONFIG**.

```
C> LIST NODES
```

Node	Sys name	Line Type	Startup	Spool	Shadow
3*	FOX	N	A	N	N
4	VIXEN	N	A	B	B

Currently 2 Nodes are defined

The \* designates the local node.

## Step 3: Install the NetBase Shell

---



**Note.** After the installation of the shell has been completed, the system must be shutdown and restarted.

---

Global installation of NetBase requires that NetBase modules be installed into the system libraries located in **PUB.SYS**. These libraries are **SL.PUB.SYS**, **RTMSL.PUB.SYS** and **XL.PUB.SYS**. Before performing any installations to a system library, you should perform a complete backup of all important data. In addition, you should have a current SLT prior to the install. The system should be quiet, this means no active jobs or sessions, during the installation.

Global installation is accomplished via **NBINST.PUB.NETBASE**, an auto-installation program. This installation program requires the user to be **MANAGER.SYS**. **NBINST** is run with an **INFO** string depending on the product(s) being installed.

The following table lists the various installation options and when each is used.

Install Option	Usage
<b>INSTALL</b>	Most installations.
<b>INSTALL OMNIDEX</b>	Required if using OMNIDEX 3.0 and NetBase NFA. If using this option, there is an additional configuration step. Turn to Chapter 10 “NetBase and Third Party Products” for more information on the OMNIDEX/NetBase interface.
<b>INSTALL NOIMAGE</b>	Used when installing NBSpool Plus <i>only</i> (without NetBase).
<b>INSTALL NFS</b>	Used when installing NFS Client <i>only</i> (without NetBase).

\_\_\_\_\_ 3.1 Run the **NBINST** installation program:

```

:HELLO MANAGER.SYS, PUB
:SETJCW NBDISABLE=FATAL
:RUN NBINST.PUB.NETBASE;INFO="INSTALL"

NetBase/NFS Auto-Install Program [x.x] Update n (C) QUEST Software 1996

Ready to install NetBase? Y

PHASE 1 - CHECK CURRENT INSTALLATION
PHASE 2 - CHECK SYSTEM SL STRUCTURE
PHASE 3 - SCAN SYSTEM SL
PHASE 4 - CREATE NETBASE SL
PHASE 5 - INSTALL NETBASE SYSTEM SEGMENTS
PHASE 6 - UPDATE SYSTEM SL
PHASE 7 - OCT THE NETBASE SL
PHASE 8 -OCT THE NETBASE SYSTEM SEGMENTS
PHASE 9 -INSTALL NATIVE MODE NETBASE

Now shutdown and restart the system

END OF PROGRAM

```

The above example shows a typical **NBINST** display. The actual number of steps or the text of each step may vary from installation to installation.

\_\_\_\_\_ 3.2 After the successful completion of the NetBase installation program, the system must be shut down and restarted.

## Step 4. Validate Installation and Modify System Startup Procedures

Once the NetBase Shell has been installed, NetBase will begin to intercept all file system and IMAGE intrinsic calls. This happens automatically for Native Mode (**NM**) programs. However, for the NetBase Shell to be able to intercept Compatibility Mode (**CM**) programs, NetBase must be “enabled”. The “enabling” of NetBase for **CM** programs is done through the utility, **NBVERS.PUB.NETBASE**. This “enabling” must be done each time the system is restarted and should become part of the standard system startup procedures.

### To Enable NetBase for CM programs:

#### \_\_\_\_\_ 4.1 Logon as **MGR.NETBASE** or **MANAGER.SYS**:

```

:RUN NBVERS.PUB.NETBASE,ENABLE
NetBase Version Display [x.x] Update 3 (C) QUEST Software 1996

Program File:   NBCOP.NB.NETBASE           Version:   [x.x] Upd nn
Program File:   NBCTRL.PUB.NETBASE         Version:   [x.x] Upd nn
Program File:   NBDIR.PUB.NETBASE          Version:   [x.x] Upd nn
Program File:   NBINST.NB.NETBASE          Version:   [x.x] Upd nn
Program File:   NBCOM.PUB.NETBASE          Version:   [x.x] Upd nn
Program File:   NMSERVER.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBSPOOL.PUB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBSPOOLC.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBSPOOLE.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBSPOOLI.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBSPOOLT.PUB.NETBASE       Version:   [x.x] Upd nn
Program File:   NBSTATCP.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBSTATLG.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBEXPORT.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBIMPORT.NB.NETBASE        Version:   [x.x] Upd nn
Program File:   NBPOST.NB.NETBASE          Version:   [x.x] Upd nn
Program File:   NBILOGC.NB.NETBASE         Version:   [x.x] Upd nn
Program File:   NBILOG.NB.NETBASE          Version:   [x.x] Upd nn
Program File:   NBSYNC.PUB.NETBASE         Version:   [x.x] Upd 00
Program File:   NBFLUSH.PUB.NETBASE        Version:   [x.x] Upd 00

Program File:   NFS.NET.SYS                Allocated
Program File:   CMSTORE.PUB.SYS            Allocated
Program File:   RPMDAD.NET.SYS             Allocated

```

```
NetBase access has been ENABLED
```

The NetBase utility **NBVERS** can be used to verify the version of the NetBase shell procedures installed in the system SL (and XL).

\_\_\_\_\_ 4.2 Verify the installation of the NetBase Shell

```
:RUN NBVERS.PUB.NETBASE,SHELL

NetBase Version Display [x.x] Update n (C) QUEST Software 1996

SL Procedure: NBVERSION'SSL Version: [x.x] Upd nn
SL Procedure: NBVERSION'SSL1 Version: [x.x] Upd nn
SL Procedure: NBVERSION'PSL Version: [x.x] Upd nn
SL Procedure: NBVERSION'PSL1 Version: [x.x] Upd nn
XL Procedure: NMVersionXLI Version: [x.x] Upd nn
XL Procedure: NMVersionXLF Version: [x.x] Upd nn
XL Procedure: NMVersion XLP Version: [x.x] Upd nn
XL Procedure: NMVersionSYS Version: [x.x] Upd nn
```

\_\_\_\_\_ 4.3 Modify system startup procedures:

Now that NetBase is globally installed on the system, the system startup procedures must be modified to include NetBase. The NetBase steps in the system startup must be performed each time the system is restarted and needs to be performed in the order listed below:

1. Start the network. Since NetBase makes use of the network, the NetBase background job cannot run unless the network has already been started. Starting the network includes the **NETCONTROL** commands.
2. Enable NetBase for **CM** programs. By default only **NM** programs are automatically intercepted by the NetBase shell, therefore everytime the system is restarted, the NetBase shell must be enabled for **CM** programs.

```
:RUN NBVERS.PUB.NETBASE,ENABLE
```

3. Start the NetBase background job:

```
:RUN NBCTRL.PUB.NETBASE;INFO="START NETBASE"
```

- 4.4 Set the **NMCMUDUC** to enable the capture of certain MPE commands such as **BUILD**, **PURGE**, **RENAME** etc. This is required if shadowing non-IMAGE files.



**Note.** If the Quest **NMCMUDUC** was disabled for this update, you must set it again. Enter the following command:

```
:SETCATALOG NMCMUDUC.PUB.NETBASE;SYSTEM;APPEND
```

---

## Implementation Notes

This section provides notes on system startup procedures and global installation.

### System Startup Procedures

Once NetBase is installed and running, you should take steps to insure that it is always running when you expect it to be. We recommend modifying your **SYSSTART** file to automatically enable NetBase and launch your NetBase services when your system is brought up.

Sample startup procedure:

1. Start the network
2. Enable NetBase for CM programs
3. Start the NetBase background job

### Global Installation

Now that NetBase is globally installed on the system, all programs will load through the NetBase procedures. However, some programs/software need to be ignored by the NetBase shell. This is accomplished in one of two ways. If it is a **CM** program, then the file **ALLOCATE.DATA.NETBASE** can be used to place individual filenames of programs to **Allocated** prior to enabling NetBase for **CM** programs. The other way makes use of the NetBase directory and the **PROGRAM** entry. Utility software such as **ADAGER**, **DBGENRL**, **FASTBACK** are examples of software that should be **NONBed** - ignored by the NetBase shell.

For **CM** programs that should not run through the shell, include the file name in the **ALLOCATE.DATA.NETBASE** file. When the **NBVERS,ENABLE** command is issued, any programs specified in **ALLOCATE.DATA** will be loaded through **SL.PUB.SYS** before NetBase is enabled for **CM** programs.

For **NM** programs, use the **PROGRAM** command of **NBDIR** with the **;NONB** option to disable a specific program or set of programs. Following is an example of a **PROGRAM** entry:

```
:RUN NBDIR.PUB.NETBASE
```

```
NetBase Directory Program [x.x] Update n (C) QUEST Software 1996
```

```
D> PROGRAM @.@.ORBIT;NONB
```

```
D> EXIT
```

The default NetBase directory already contains a number of **NONB** statements. They are automatically added any time a directory is built. To see what **PROGRAM** statements are already in the NetBase directory, use the **LISTP @.@.@** command in **NBDIR**. Turn to Chapter 4 “NBDIR” for more information on the **PROGRAM** command.

There is also a file called **SLEXCLUD.DATA.NETBASE** that can be used to prevent **NBINST** from adding certain segments into the **RTMSL.PUB.SYS** file. For example, the **FORMATION** segments can be prevented from being placed into **RTMSL** by editing the file **SLEXCLUD** and adding an entry for **PSP@**.

```
:PRINT SLEXCLUD.DATA.NETBASE
NETBASE@
VESOFT@
CORONA2
TIMAGE
PSP@
```

## Installation Appendix

This Appendix covers the following subjects:

- Converting from an account installation to global installation. With the 9.7 release of NetBase, account installations are no longer supported. Before updating to 9.7 NetBase, the account installation must be removed.
- Updating from a global installation of NetBase that is previous to the 9.6 Release.

If you have any questions, or if you are unsure about any of the steps, please contact Technical Support at (714)-720-1434.

### Converting From an Account Installation to a Global Installation

The Account installation must be removed prior to installing the 9.7 Release. Perform the steps below for removing the Account installation prior to updating.

\_\_\_\_\_ 1. Logon as **MGR.NETBASE**.

\_\_\_\_\_ 2. Use **NBCTRL** to remove the NetBase segments from account libraries:

```
:RUN NBCTRL.PUB.NETBASE
*> CONFIG
C> REMOVE @
```

Verify that the NetBase segments have been removed from all account libraries:

```
C> SHOW
```

\_\_\_\_\_ 3. Proceed to **Step 1** in the “Installation Steps” section.

## Updating From Global Installation Prior to NetBase 9.6

Any release prior to 9.6 is unsupported and the NetBase Shell must be removed before updating to the 9.7 Release.

- \_\_\_\_\_ 1. Logon as **MANAGER.SYS,PUB**.
- \_\_\_\_\_ 2. Remove the NetBase Shell:  
  
          **:RUN NBVERS . PUB . NETBASE , DISABLE**  
  
          **:RUN NBINST . PUB . NETBASE ; INFO="REMOVE"**
- \_\_\_\_\_ 3. Reboot the system.
- \_\_\_\_\_ 4. Proceed to **Step 1** of the "Installation Steps" section.

## A P P E N D I X B

# Troubleshooting NetBase

In this chapter, you will find questions with answers and tips from Quest technical support. Following these, you will find NetBase error messages listed with explanations and actions to take.

### In This Chapter

- **Technical Support Tips**
- **NetBase Error Messages**

## Technical Support Tips

This section lists commonly asked questions on the following topics, with answers and tips from Quest technical support:

- **SHOW POST**
- **versions**
- **NFA**
- **NBSYNC,VERIFY**
- **MPE builds, purges and renames.**

### SHOW POST

#### QUESTION:

*What does it mean when I see ??? in the **TYPE** column of the **SHOW POST Node#** command, and what should I do?*

#### ANSWER:

The ??? in the **Type** column of a **SHOW POST Node#** command indicates that **NBPOST** (the posting process) encountered a problem opening the shadow copy of a file. For example:

```
*> SHOW POST 7
```

File	Type	Update	Time Stamp of Last Record Processed	S Q D Y U I N E S
PAYDB.DATA.PAYROLL	???	0	06/01/92 10:39 am	- Q -

This display indicates that **NBPOST** was unable to open the shadow file. The posting process knows there are transactions for this file, but for some reason is unable to open the file. Also notice that the **QUE** flag has been set to **Q**, which indicates the transactions for this file are being queued. When you see this display, you can examine the **\$STDLIST** of the current NetBase background job to find out what error was encountered by the posting process (error message is also displayed to the console):

```
:RUN NBSPOOL.PUB.NETBASE
O> T $,MGR.NETBASE,S=O
O> L LAST-10/LAST
```

```
57 153 12:54:05 NBP07/WARNING 23: UNABLE TO OPEN DATABASE PAYDB.DATA.PAYROLL
58 153 12:54:06 NBP07/BAD PASSWORD - GRANTS ACCESS TO NOTHING (DBERR -21)
59 153 12:54:06 NBP07/STARTING POST QUEUE FOR PAYDB.DATA.PAYROLL
```

The error message in the background job indicates that a bad password has been defined and that queuing for this file has started. The password for a shadowed database is specified with the **SHADOW** command in **NBDIR**.

Check the database passwords using **DBUTIL** and reissue the **SHADOW** statement with the correct password:

```
:RUN NBDIR.PUB.NETBASE  
D> SHADOW PAYDB.DATA.PAYROLL;PASS=WRITER
```

After reissuing the **SHADOW** statement, the next thing we need to do is get **NBPOST** to post the transactions that have been queuing. This is accomplished by issuing a **ROLL FORWARD** command within **NBCTRL**. The **ROLL FORWARD** command tells the posting process to begin posting the queued transactions for a file that has been queuing:

```
:RUN NBCTRL.PUB.NETBASE  
*> ROLL FORWARD PAYDB.DATA.PAYROLL
```

At this point we should be able to issue a **SHOW POST Node#** command again and we should see **IMAGE** under the **TYPE** column and the **QUE** flag should be blank.

## NetBase Versions

### QUESTION:

*How do I tell what version of NetBase is installed on my system?*

### ANSWER:

The **NBVERS** utility can be used to determine the version of the NetBase Shell and the NetBase programs.

#### To find the NetBase Shell version:

```
:RUN NBVERS.PUB.NETBASE,SHELL
```

#### To find the version of NetBase programs:

```
:RUN NBVERS.PUB.NETBASE,PROG
```

### Sample

```
:run nbvers.pub,prog
```

```
NetBase Version Display [0.9.7] Update 0 (C) QUEST Software 1996
```

Program File:	NBCOP.NB.NETBASE	Version:	[0.9.7]	Upd 09
Program File:	NBCTRL.PUB.NETBASE	Version:	[0.9.7]	Upd 09
Program File:	NBDIR.PUB.NETBASE	Version:	[0.9.7]	Upd 04
Program File:	NBINST.PUB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NBCOM.NB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NMSERVER.NB.NETBASE	Version:	[0.9.7a]	Upd 00
Program File:	NBSPOOL.PUB.NETBASE	Version:	[0.9.7]	Upd 02
Program File:	NBSPOOLC.NB.NETBASE	Version:	[0.9.7]	Upd 01
Program File:	NBSPOOLE.NB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NBSPOOLI.NB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NBSPOOLT.PUB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NBSPOOLR.PUB.NETBASE	Version:	[0.9.7]	Upd 02
Program File:	NBSTATCP.NB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NBSTATLG.NB.NETBASE	Version:	[0.9.7]	Upd 00
Program File:	NBEXPORT.NB.NETBASE	Version:	[0.9.7]	Upd 04
Program File:	NBEXPCOM.NB.NETBASE	Version:	[0.9.7]	Upd 02
Program File:	NBEXPRDR.NB.NETBASE	Version:	[0.9.7]	Upd 06
Program File:	NBIMPORT.NB.NETBASE	Version:	[0.9.7]	Upd 03
Program File:	NBPOST.NB.NETBASE	Version:	[0.9.7]	Upd 05
Program File:	NBILOGC.NB.NETBASE	Version:	[0.9.7]	Upd 00

## Troubleshooting NetBase

---

Program File:	NBILOG.NB.NETBASE	Version:	[0.9.7] Upd 00
Program File:	NBSYNC.PUB.NETBASE	Version:	[0.9.7] Upd 00
Program File:	NBFLUSH.PUB.NETBASE	Version:	[0.9.7] Upd 00
Program File:	NBSQLCTL.NB.NETBASE	Version:	NOT INSTALLED
Program File:	NBSQLEXP.NB.NETBASE	Version:	NOT INSTALLED
Program File:	NBSQLIMP.NB.NETBASE	Version:	NOT INSTALLED

END OF PROGRAM

BUGS:

BUGS:RUN NBVERS.PUB.NETBASE,SHELL

NetBase Version Display [0.9.7] Update 0 (C) QUEST Software 1996

SL Procedure:	NBVERSION'SSL	Version:	[0.9.7] Upd 00
SL Procedure:	NBVERSION'SSL1	Version:	[0.9.7] Upd 00
SL Procedure:	NBVERSION'PSL	Version:	[0.9.7] Upd 00
SL Procedure:	NBVERSION'PSL1	Version:	[0.9.7] Upd 00
XL Procedure:	NMVersionXLI	Version:	[0.9.7] Upd 02
XL Procedure:	NMVersionXLF	Version:	[0.9.7] Upd 02
XL Procedure:	NMVersionXLP	Version:	[0.9.7] Upd 02
XL Procedure:	NMVersionSYS	Version:	[0.9.7] Upd 02

END OF PROGRAM



The **SHOW REMOTE** command shows all users coming into this system for their file access. It also displays the **PIN** number of the associated server in the **SPIN** column. By matching the pin number from the **DBUTIL** command with the server pin (**SPIN**) number from the show remote command, you can identify the user process that is accessing the database. In this example, server pin **92** is associated to job number **#S200** with a logon of **USER.PROD**. This user is running Query to access the database.

The **SHOW SERVER** command displays essentially the same information as the **SHOW REMOTE** command. However, it also provides the names and modes of the opened files.

```
*>SHOW SERVER 1
```

NODE	JOBNUM	SRVR	UPIN	JOBNAME	PROGRAM NAME
SYSA	#S200	1	136	USER.PROD	QUERY.PUB.SYS
65	PAYROLL.DATA.PROD				OPEN MODE 1

By using the **SHOW REMOTE** and **SHOW SERVER** commands, you have been able to identify the user and program holding the lock on the database.

## NBSYNC, VERIFY

### QUESTION:

*Is there a program I can run to check the status of my shadowed files in NBDIR?*

### ANSWER:

Yes, a **VERIFY** entry point **NBSYNC** has been added to allow jobs and UDCs to check to see if a file is in the **DISCARD**, **SYNC** or **QUEUE** state. Run **NBSYNC** with the **VERIFY** entry point, specifying the file (or fileset) in the **INFO** string. A JCW named **VERJWCW** will have the following values:

```
:RUN NBSYNC.PUB.NETBASE,VERIFY;INFO="PAYDB.FILE.PAYROLL"  
:SHOWJCW VERJWCW
```

The following table lists the **VERJWCW** values.

Value	Description
<b>WARN</b>	A file entry exists, but the file does not exist on the system.
<b>0</b>	Everything is okay
<b>1</b>	File is in the <b>QUEUE</b> state.
<b>2</b>	File is in the <b>SYNC</b> error state.
<b>4</b>	File is in the <b>DISCARD</b> state.

You can have a combination of these values. For example, if a file does not exist on the system and a database is out of sync, the value for **VERJWCW** will be **WARN2**. Another example is if a file is in queue state and another file is out of sync, the **VERJWCW** value will be **3**.

## MPE Builds, Purges and Renames

### QUESTION:

*How can I use NetBase to intercept MPE commands such as **BUILD**, **PURGE**, **SAVE**, **COPY** and **RENAME**?*

### ANSWER:

You can invoke a UDC from a sample file provided by Quest Software. Essentially, each UDC runs **NMCMC.PUB.NETBASE** which passes your request to **HPCICOMMAND**. From this point, NetBase can intercept the command. We recommend the following sample UDCs:

```
build
anyparm parm1
xeq nmcmd.pub.netbase;info="build !parm1";parm=!HPMSGFENCE
if nmcmd = 1
    escape !cierror
endif
***
purge
anyparm parm1
xeq nmcmd.pub.netbase;info="purge !parm1";parm=!HPMSGFENCE
if nmcmd = 1
    escape !cierror
endif
***
rename
anyparm parm1
xeq nmcmd.pub.netbase;info="rename !parm1";parm=!HPMSGFENCE
if nmcmd = 1
    escape !cierror
endif
***
save
anyparm parm1
xeq nmcmd.pub.netbase;info="save !parm1";parm=!HPMSGFENCE
if nmcmd = 1
    escape !cierror
endif
***
copy
anyparm parm1
xeq nmcmd.pub.netbase;info="copy !parm1";parm=!HPMSGFENCE
if nmcmd = 1
    escape !cierror
endif
***
```

Use the **MPE SETCATALOG** command to make these UDCs available (system-wide) by appending the filename to the list of UDC files you use.

## NetBase Error Messages

This section includes the following types of error messages:

- **File system**
- **Directory file (NBD)**
- **Network configuration file (NBI)**
- **Network control file (NBN)**
- **Shadowing queue file (NBM)**
- **Statistics logging file (NBVL)**
- **Spooling files (NBSEC and NBSED)**

Also included are errors that are detected by:

- **AutoRPM facility**
- **NetBase control program (NBCOP)**
- **Spooling processes**
- **NBSERVER**
- **NMSERVER**
- **Shell**

Each error message has an explanation and a suggested action to take.

## File System Errors

The following NetBase errors are returned to the user's program through the **FCHECK** intrinsic and the **CONDITION WORD** of all **IMAGE** calls. If the user's program calls **FERRMSG**, **PRINTFILEINFO**, **DBERROR** or **DBEXPLAIN** the new messages will automatically be displayed.

ERR#	Error Message	Explanation/Action
1001	LOCAL NETBASE SERVICES ARE DOWN	NetBase is not running on the local node.
1002	NETBASE VERSION MISMATCH: REINSTALL NETBASE SEGMENTS	The versions of the NetBase segments do not match. NetBase must be reinstalled in the appropriate SLs. Run <b>NBVERS</b> to determine which segments are in error.
1003	REQUESTED NODE IS DOWN	NetBase services on the requested node are down. Start NetBase on the requested node.
1004	COMMUNICATION WITH NODE HAS BEEN TERMINATED	Communication failure caused by a system or network failure. Only an <b>FCLOSE</b> or a <b>DBCLOSE</b> can be issued for this file.
1005	OUTGOING USER TABLE IS FULL	Too many users are trying to access remote files. The user table may be modified with the <b>GLOBAL - Maximum Local Users</b> configuration parameter.
1006	VERSION MISMATCH BETWEEN NETBASE SEGMENTS AND PROCESSES	The versions of the NetBase segments do not match. NetBase must be reinstalled in the appropriate SLs.
1007	ACCESS TO NODE DENIED BY SYSTEM MANAGER	Access to the remote node has been stopped using NetBase control.
1008	OUTGOING USER LIMIT HAS BEEN EXCEEDED	Too many users are trying to access remote files. The user table may be modified with the <b>GLOBAL - Maximum Local Users</b> configuration parameter.
1009	ACCESS DISABLED DUE TO SHADOWING SYNC ERROR	Access to the local file has been disabled because it is no longer in sync with its master.

*Continued*

Appendix B

ERR#	Error Message	Explanation/Action
1010	INVALID OVERRIDE FLAGS SPECIFIED BY JCW	Flags are set in the <b>NBDISABLE</b> JCW that are not used by NetBase. Check the JCW for invalid values.
1011	NO SERVERS AVAILABLE ON REMOTE NODE	The number of users accessing files or databases on the remote node exceeds the maximum. The servers may be increased by modifying the <b>NFA - Maximum Remote Users</b> configuration parameter on the remote node.
1012	NETBASE ACCESS DENIED BY REMOTE NODE	Access to the local node by remote users has been stopped using NetBase control.
1013	INVALID OVERRIDE FLAGS SPECIFIED IN DIRECTORY	NetBase internal error. If this error occurs contact Quest Software.
1014	NO RESPONSE FROM NETWORK CONTROL PROCESS	See the console for error information. The remote system may have gone down. Note: an unusually high system load may cause this problem.
1015	INVALID BYTE COUNT FOR NFA, READ OR WRITE	The number of bytes to be written or read exceeds 28,672.
1016	NO RESPONSE FROM REMOTE SERVER PROCESS	See the console for error information. The remote system may have gone down. Note: an unusually high system load may cause this problem.
1018	NETWORK TRANSPORT NOT ACTIVE; MUST BE STARTED WITH :NETCONTROL	Network transport must be active for NetBase to run. Start the transport with the <b>:NETCONTROL</b> command and restart NetBase.
1019	REMOTE ACCESS NOT SUPPORTED WITH SPECIFIED ACCESS OPTIONS	A user process is attempting to open a file with the <b>nowait</b> I/O access option.
1023	UNABLE TO OBTAIN FILE INFORMATION DURING OPEN	NetBase internal error. If this error occurs contact Quest Software.
1025	FILE CANNOT BE SHADOWED	Shadowing is invalid for non-disc files, files opened for mapped access, and POSIX files.

ERR#	Error Message	Explanation/Action
1026	SHADOWING INVALID WITH SPECIFIED ACCESS OPTIONS	Shadowing is invalid for files opened with nowait I/O file access.
1027	USER LACKS CAPABILITY TO SHADOW FILE	A user accessing a shadowed file must have both read and write access to the file.
1028	INSUFFICIENT STACK SPACE AVAILABLE	Insufficient stack space for IMAGE call. Increase the maximum stack value on the user program.
1029	SHADOW PATH DOES NOT HAVE ACCESS TO ENTIRE DATA SET	The password used to open the database does not allow access to all items in the data sets. Use another password.
1030	REMOTE SPOOLING IS NOT ENABLED	Remote spooling is disabled. Enable spooling using the <b>ENABLE</b> command in <b>NBCTRL</b> .
1031	SPECIFIED NODE IS NOT CONFIGURED FOR SPOOLING	A user has attempted to create a spool file on a remote node that is not configured for spooling. Check NetBase's configuration.
1032	REMOTE SPOOLING TO SPECIFIED NODE IS DOWN	A user has attempted to create a spool file on a remote node that is down.
1033	NETBASE SPOOL FILE TABLE FOR THIS NODE IS FULL	The number of open spool files originating from a remote node exceeds the maximum. The file table may be increased the <b>SPOOL - Maximum Files per Node</b> configuration parameter.
1035	INVALID READ PASSWORD CONFIGURED FOR SHADOWED DATA BASE	The read password specified on the directory <b>FILE</b> command for this database is not valid.
1036	READ PASSWORD MUST BE CONFIGURED FOR SHADOWED DATA BASE	The read password must be specified on the directory <b>FILE</b> command for this database.
1037	CANNOT OPEN SHADOW MULTIPLEXER	The NetBase configuration is corrupt and may require a restore.
1040	WRITE LENGTH INVALID FOR SHADOWING	The number of bytes being written exceeds 32,767.

Appendix B

ERR#	Error Message	Explanation/Action
1041	GMULTI ACCESS WITHOUT LOCKING INVALID FOR SHADOWING	The shadow file is being opened with <b>GMULTI</b> access but not allowing locking. This can be solved by adding <b>;SYNC</b> to the file statement for the file in question.
1042	UPDATE ON SHADOW FILE INVALID WITHOUT LOCK	The shadow file is not locked and the number of writers that have it open exceeds one.
1045	TPS CALLED WITHOUT NETBASE'S IMAGE INTERCEPT	Check to make sure NetBase is enabled. Run <b>NBVERS,STATE</b> . If disabled, enable it with <b>RUN NBVERS, ENABLE</b> .
1050	GETDSEG FAILED	An attempt to get an extra data segment failed.
1051	ALTDSEG FAILED	An attempt to get an extra data segment failed.
1052	FREEDSEG FAILED	NetBase internal error. If this error occurs contact Quest Software.
1054	GETDSEG FAILED FOR SHADOWING	An attempt to get an extra data segment failed.
1055	FREEDSEG FAILED FOR SHADOWING	NetBase internal error. If this error occurs contact Quest Software.
1056	UNABLE TO LOCK GLOBAL RIN	NetBase internal error. If this error occurs contact Quest Software.
1060	IPCRECV INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1061	IPCSEND INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1062	IPCSHUTDOWN INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1063	IPCCREATE INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.

ERR#	Error Message	Explanation/Action
1064	IPCNAME INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1065	IPCREVCVN INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1066	IPCCONTROL INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1070	UNABLE TO OBTAIN FILE LABEL INFORMATION FOR SHADOWED FILE	NetBase internal error. If this error occurs contact Quest Software.
1071	UNABLE TO OPEN SHADOWED FILE	NetBase internal error. If this error occurs contact Quest Software.
1072	UNABLE TO READ SHADOWED FILE	NetBase internal error. If this error occurs contact Quest Software.
1073	UNABLE TO CLOSE SHADOWED FILE	NetBase internal error. If this error occurs contact Quest Software.
1074	FFILEINFO FAILED ON SHADOWED FILE	NetBase internal error. If this error occurs contact Quest Software.
1075	FREADLABEL FAILED IN SHADOWED FILE	A new, shadowed file is being copied to the shadow machine(s). It has labels, but <b>FREADLABEL</b> to copy the labels failed.

## Directory File (NBD) Related Errors

The following NetBase errors are related to the directory file (NBD).

ERR#	Error Message	Explanation/Action
1100	CANNOT OPEN NETWORK DIRECTORY	The directory file is invalid or does not exist. Check that the active directory exists and is valid. Also check for file equations.
1101	CANNOT READ NETWORK DIRECTORY	The directory may be corrupt. Rebuild using the <b>LIMIT</b> command in <b>NBDIR</b> .
1102	INVALID NETWORK DIRECTORY	The directory file is invalid or has been corrupted. Check that the active directory is valid.
1103	INVALID NODE IN DIRECTORY	An entry in the directory has an invalid node number. Make sure the directory entries match the configuration.
1104	CANNOT CLOSE NETWORK DIRECTORY	NetBase internal error. If this error occurs contact Quest Software.

## Network Configuration File (NBI) Related Errors

The following NetBase errors are related to the network configuration file (NBI).

ERR#	Error Message	Explanation/Action
1120	CANNOT OPEN NETWORK CONFIGURATION FILE	The NetBase configuration is corrupt and may require a restore. Contact Technical Support.
1121	CANNOT READ NETWORK CONFIGURATION FILE	The NetBase configuration is corrupt and may require a restore. Contact Technical Support.
1122	INVALID CONFIGURATION FILE HEADER	The NetBase configuration is corrupt and may require a restore.
1123	INVALID LOCAL NODE NUMBER IN CONFIGURATION	The NetBase configuration is corrupt and may require a restore.
1124	CANNOT CLOSE NETWORK CONFIGURATION FILE	The NetBase configuration is corrupt and may require a restore.
1125	OBSOLETE NETWORK CONFIGURATION FILE	Run <b>NBCTRL</b> to convert the configuration file. Re-install NetBase if necessary.
1126	INCOMPATIBLE NETWORK CONFIGURATION FILE	Run <b>NBCTRL</b> to convert the configuration file. Re-install NetBase if necessary.



**Note.** Contact Technical Support before doing anything for any of the above errors.

## Network Control File (NBN) Related Errors

The following NetBase errors are related to the network control file (NBN).

ERR#	Error Message	Explanation/Action
1141	CANNOT WRITE TO NETWORK CONTROL IPC FILE	NetBase internal error. If this error occurs contact Quest Software.
1142	CANNOT CLOSE NETWORK CONTROL IPC FILE	NetBase internal error. If this error occurs contact Quest Software.
1143	FFILEINFO ON NETWORK CONTROL IPC FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.

## Shadowing Queue file (NBM) Related Errors

The following NetBase errors are related to the shadowing queue file (NBM).

ERR#	Error Message	Explanation/Action
1170	CANNOT OPEN SHADOW EXPORT FILE	The NetBase configuration is corrupt and may require a restore. Contact Quest.
1171	FCONTROL ON SHADOW EXPORT FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.
1172	CANNOT CLOSE SHADOW EXPORT FILE	NetBase internal error. If this error occurs contact Quest Software.
1173	FSETMODE ON SHADOW EXPORT FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.
1174	CANNOT OPEN NBM SEQUENCE FILE	NetBase internal error. If this error occurs contact Quest Software.
1175	CANNOT OPEN SEQUENCE SYNC FILE	NetBase internal error. If this error occurs contact Quest Software.
1176	FLOCK ON SEQUENCE SYNC FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.
1177	FFILEINFO ON SEQUENCE FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.
1178	SEQUENCE FILE INITIALIZATION FAILED	NetBase internal error. If this error occurs contact Quest Software.
1179	CANNOT CLOSE SEQUENCE SYNC FILE	NetBase internal error. If this error occurs contact Quest Software.

## Statistics Logging File (NBVL) Related Errors

The following NetBase errors are related to the statistics logging file (NBVL).

ERR#	Error Message	Explanation/Action
1180	CANNOT OPEN STATISTICS LOGGING FILE	The NetBase configuration is corrupt and may require a restore.
1181	FCONTROL ON STATISTICS LOGGING FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.
1182	FFILEINFO ON STATISTICS LOGGING FILE FAILED	NetBase internal error. If this error occurs contact Quest Software.
1183	CANNOT WRITE TO STATISTICS LOGGING FILE	NetBase internal error. If this error occurs contact Quest Software.

## Spooling Files (NBSEC and NBSED) Related Errors

The following NetBase errors are related to the spooling files (NBSEC, NBSED).

ERR#	Error Message	Explanation/Action
1190	CANNOT OPEN SPOOLING CONTROL FILE	The NetBase configuration is corrupt and may require a restore.
1191	CANNOT SET TIMEOUT ON SPOOLING CONTROL FILE	NetBase internal error. If this error occurs contact Quest Software.
1192	CANNOT WRITE TO SPOOLING CONTROL FILE	NetBase internal error. If this error occurs contact Quest Software.
1193	FFILEINFO FAILED ON SPOOLING CONTROL FILE	NetBase internal error. If this error occurs contact Quest Software.
1194	CANNOT CLOSE SPOOLING CONTROL FILE	NetBase internal error. If this error occurs contact Quest Software.
1195	CANNOT OPEN SPOOLING DATA FILE	NetBase internal error. If this error occurs contact Quest Software.
1196	CANNOT SET TIMEOUT ON SPOOLING DATA FILE	NetBase internal error. If this error occurs contact Quest Software.
1197	CANNOT WRITE TO SPOOLING DATA FILE	NetBase internal error. If this error occurs contact Quest Software.
1198	CANNOT CLOSE SPOOLING DATA FILE	NetBase internal error. If this error occurs contact Quest Software.

## AutoRPM Facility Detected Errors

The following NetBase errors are detected by the AutoRPM facility.

ERR#	Error Message	Explanation/Action
1301	NOT ENOUGH STACK SPACE TO EXECUTE REMOTE PROCESS	Insufficient stack space is available to create the remote process. Increase the <b>MAXDATA</b> for the local program.
1302	CANNOT EXECUTE REMOTE NETBASE PROCESS	NetBase internal error. If this error occurs contact Quest Software.
1303	CANNOT EXECUTE LOCAL NETBASE PROCESS	NetBase internal error. If this error occurs contact Quest Software.
1304	REMOTE PROCESS HANDLING NOT INSTALLED ON LOCAL COMPUTER	Remote process handling is not installed on the local computer.
1305	REMOTE PROCESS HANDLING NOT INSTALLED ON REMOTE COMPUTER	Remote process handling is not installed on the remote computer.
1306	UNABLE TO CREATE REMOTE PROCESS	NetBase was not able to create the process on the remote computer. Check the <b>PROGRAM</b> directory entry and make sure that the process exists on the remote machine.
1307	NO RESPONSE FROM REMOTE COMPUTER	This problem may occur if the machine is heavily loaded or may be due to an internal error.
1308	NO LOGON INFORMATION SPECIFIED WITH DSLINE COMMAND	Check the <b>LOGON</b> directory entry for the user.
1309	REMOTE HELLO FAILED	This error may be due to an incorrect logon record in the directory or to an internal error.
1310	UNABLE TO CREATE TEMPORARY ENVIRONMENT	NetBase internal error. If this error occurs contact Quest Software.
1311	UNABLE TO CREATE LOCAL STUB PROCESS	NetBase internal error. If this error occurs contact QUEST Software.

*Continued*

ERR#	Error Message	Explanation/Action
1312	SENDMAIL INTRINSIC FAILED	NetBase internal error. If this error occurs contact QUEST Software.
1313	RECEIVEMAIL INTRINSIC FAILED	NetBase internal error. If this error occurs contact Quest Software.
1314	UNSUCCESSFUL EXECUTION OF LOCAL STUB PROCESS	NetBase internal error. If this error occurs contact Quest Software.
1315	USER NOT ALLOWED ACCESS TO SPECIFIED NODE	Check the status of the specified node. Start remote access to that node if necessary.
1350	CANNOT PURGE TEMP ENVIRONMENT FILE	NetBase internal error. If this error occurs contact Quest Software.
1351	CANNOT CREATE TEMP ENVIRONMENT FILE	NetBase internal error. If this error occurs contact Quest Software.
1352	CANNOT SAVE TEMP ENVIRONMENT FILE	NetBase internal error. If this error occurs contact Quest Software.
1353	CANNOT OPEN TEMP ENVIRONMENT FILE	NetBase internal error. If this error occurs contact Quest Software.
1354	CANNOT READ TEMP ENVIRONMENT FILE	NetBase internal error. If this error occurs contact Quest Software.
1360	COMMAND INTRINSIC FAILED	Internal error. If this error occurs contact Quest Software.
1361	INITOPT INTRINSIC FAILED	Internal error. If this error occurs contact Quest Software.
1362	ADDOPT INTRINSIC FAILED	Internal error. If this error occurs contact Quest Software.
1363	NSINFO INTRINSIC FAILED	Internal error. If this error occurs contact Quest Software.
1370	IPCCREATE INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1371	IPCNAME INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.

## Appendix B

---

ERR#	Error Message	Explanation/Action
1372	IPCCONTROL INTRINSIC FAILED ON CALL SOCKET	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1373	IPCRCVCN INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1374	IPCCONTROL INTRINSIC FAILED ON VC SOCKET	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1375	IPCSEND INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.
1376	IPCRCV INTRINSIC FAILED	This error indicates that a problem exists in the network transport. See the associated error message for the cause of the error.

## NetBase Control Program (NBCOP) Detected Errors

The following NetBase errors are detected by the NetBase control program (**NBCOP**).

ERR#	Error Message	Explanation/Action
2002	INVALID LOCAL NODE SPECIFIED IN REQUEST	NetBase internal error. If this error occurs contact Quest Software.
2003	INVALID REMOTE NODE SPECIFIED IN REQUEST	The <b>PROGRAM</b> entry in the directory specifies an invalid node number. Make sure the directory entry matches the configuration.
2006	CANNOT CREATE SERVER PROCESS	NetBase internal error. If this error occurs contact Quest Software.
2007	CANNOT COMMUNICATE WITH REMOTE NODE	See the system console for more information about this problem.
2008	STATISTICS ARE NOT ENABLED	NetBase statistics are disabled. Statistics may be enabled using the <b>ENABLE</b> command in <b>NBCTRL</b> .
2100	STATISTICS HAVE BEEN STOPPED	NetBase statistics have been stopped.
2103	DUPLICATE STATISTICS XDS ALLOCATION REQUESTED	NetBase internal error. If this error occurs contact Quest Software.
2104	GETDSEG FAILED FOR STATISTICS XDS	An attempt to get an extra data segment failed.
2105	ALTDSEG FAILED FOR STATISTICS XDS	An attempt to get an extra data segment failed.

## Spooling Processes Detected Errors

The following NetBase errors are detected by the spooling processes.

ERR#	Error Message	Explanation/Action
3001	UNABLE TO CREATED SPOOLING DATA SEGMENT	An attempt to get an extra data segment failed.
3002	INTERNAL ERROR IN SPOOLING EXPORT PROCESS	NetBase internal error. If this error occurs contact Quest Software.
3003	WRITE ERROR ON REMOTE SPOOL FILE	NetBase internal error. If this error occurs contact Quest Software.
3004	REMOTE SPOOL FILE LIMIT EXCEEDED	The number of spool files that may be open on the remote node has been exceeded. The file limit may be increased by modifying the <b>SPOOL - Maximum Files per Node</b> configuration parameter on the remote node.
3005	SPOOL FILE SEGMENT SUPPLY EXHAUSTED	NetBase internal error. If this error occurs contact Quest Software.
3006	SPOOLING TO NODE STOPPED BY LOCAL SYSTEM MANAGEMENT	NetBase spooling to the node has been stopped.
3007	SPOOLING TO NODE STOPPED BY REMOTE SYSTEM MANAGEMENT	NetBase spooling to the node has been stopped.

## NBSERVER Detected Errors

The following NetBase errors are detected by **NBSERVER**.

ERR#	Error Message	Explanation/Action
5001	FILE RECORD SIZE EXCEEDS NETBASE MAXIMUM	The requested remote file I/O is longer than the maximum expected by NetBase.
5002	TOO MANY REMOTE DATA BASES OPEN	The number of remote databases opened by one user is greater than 512.
5003	UNKNOWN REQUEST DETECTED BY REMOTE NODE	NetBase internal error. If this error occurs contact QUEST Software.
5004	FFILEINFO FAILED DURING FOPEN PROCESSING	NetBase internal error. If this error occurs contact QUEST Software.
5020	REQUESTED READ SIZE EXCEEDS NETBASE MAXIMUM	The number of bytes that a user is reading from a remote file exceeds the NetBase's maximum buffer size.

## NMSERVER Detected Errors

The following NetBase errors are detected by **NMSERVER**.

ERR#	Error Message	Explanation/Action
5001	FILE RECORD SIZE EXCEEDS NETBASE MAXIMUM	The requested remote file I/O is longer than the maximum expected by NetBase.
5002	TOO MANY REMOTE DATA BASES OPEN	The number of remote databases opened by one user is greater than 512.
5003	UNKNOWN REQUEST DETECTED BY REMOTE NODE	NetBase internal error. If this error occurs contact Quest Software.
5004	FFILEINFO FAILED DURING FOPEN PROCESSING	NetBase internal error. If this error occurs contact Quest Software.
5020	REQUESTED READ SIZE EXCEEDS NETBASE MAXIMUM	The number of bytes that a user is reading from a remote file exceeds the NetBase's maximum buffer size.
5021	PREVIOUS LOCK FAILED: BASE IS NOT LOCKED	Error is returned when pending locks is enabled on the remote side of the lock failed.
5022	AIFFILEGPUT FAILED SETTING CREATOR ON USER FILE	Attempt to set the creator on a new file failed.
5023	UNABLE TO LOAD AIF CALL	Attempt to load <b>AIFKSMCREATE</b> or <b>AIFKSMWRITE</b> failed. Check the operating system (it may not be consistent with this operation).
5024	AIFKSMCREATE FAILED, CANNOT CREATE KSAMXL FILE	Attempt to call <b>AIFKSMCREATE</b> to create new <b>KSAMXL</b> file failed. Check the system log.
5025	AIFKSMWRITE FAILED, CANNOT WRITE TO KSAMXL FILE	Attempt to call <b>AIFKSMWRITE</b> to create new <b>KSAMXL</b> file failed. Check the system log.
5026	AIFCLOSE FAILED, CANNOT SAVE KSAMXL FILE	Attempt to call <b>AIFCLOSE</b> to close new <b>KSAMXL</b> file failed. Check the system log.

*Continued*

ERR#	Error Message	Explanation/Action
5800	DATA RETURNED FROM USER PROCEDURE EXCEEDS RECEIVE LENGTH	Remote procedure call returned a length greater than specified by remote NBRPC call.
5801	INVALID LENGTH FOR CACHING INFO, TOO MANY ITEMS	The number of items in the cached data set is too great. The cache info would exceed the maximum send length.

## Shell Detected Errors

The following NetBase errors are detected by the shell.

ERR#	Error Message	Explanation/Action
8002	SHADOWING INVALID WITH SPECIFIED HPFOPEN OPTIONS	Shadowing is invalid when used in conjunction with <b>HPFOPEN</b> options 18, 21 (mapped files), 45, 50 or 52.
8003	REMOTE ACCESS INVALID WITH SPECIFIED HPFOPEN OPTIONS	Remote access is invalid when used in conjunction with <b>HPFOPEN</b> options 18, 21 (mapped files), 45, 50, 52 or 54.
8004	CANNOT OPEN LOCAL FILE FOR REMOTE FILE NUMBER	NetBase internal error. If this error occurs contact Quest Software.
8005	NATIVE MODE AUTO RPM NOT ENABLED	AutoRPM is not yet enabled for the native mode shell.
8006	SHADOWING INVALID WITH FILE DISPOSITION SET	A file disposition of <b>DEL</b> , <b>TEMP</b> , <b>SAVE</b> or <b>SPSAVE</b> has been specified for a file defined for shadowing. These options are not valid for shadowed files.
8009	DBCONTROL INVALID MODE, LOCAL WHEN ASKED NOT SET	<b>DBCONTROL</b> mode to change to local access used without dbcontrol mode 701 to allow this.
8010	FUPDATE CALLED AFTER FREADC, INVALID WITH THIS OS	<b>FUPDATE</b> called after <b>FREADC</b> on a shadowed file not supported before 4.0.
8011	CANNOT RENAME THE SHADOW COPY OF A FILE	<b>FRENAME</b> called on the shadow copy of a file.
8012	CANNOT COPY A KSAMXL FILE WITH THIS OS	Cannot create a new, remote KSAMXL file, the operating system must be 4.0 or greater.
8013	CANNOT LOAD AIFKSMREAD FOR COPY OF KSAMXL FILE	Cannot create a new, remote KSAMXL file. The load of the <b>AIFKSMREAD</b> failed.
8014	CANNOT CREATE KSAM FILE, NOT SUPPORTED BY SERVER	Server version on remote node does not support creation of KSAMXL files.

*Continued*

ERR#	Error Message	Explanation/Action
8015	RENAME OF SHADOWED FILE TO REMOTE FILE INVALID	Cannot rename a shadowed file to a shadow copy or remote file.
8016	MULTIPLE TRANSACTION - EXCEEDS MAXIMUM BASES	For <b>DBXBEGIN</b> mode 3 or 4, the number of bases in the list is greater than 23.
8017	MULTIPLE TRANSACTION - MULTIPLE TRANSACTION ALREADY STARTED	A <b>DBXBEGIN</b> mode 3 or 4 was called while multiple transactions were in process.
8018	MULTIPLE TRANSACTION - CANNOT MIX REMOTE AND LOCAL BASES	For <b>DBXEND</b> mode 3 or 4, the base list contains at least one remote and one local base.
8019	MULTIPLE TRANSACTION - CANNOT REMOTE TO MORE THAN ONE NODE	For <b>DBXBEGIN</b> mode 3 or 4, the list of bases are all remote, but not remote to the same node.
8020	MULTIPLE TRANSACTION - BASE ARRAY FOR END MUST MATCH BEGIN	For <b>DBXEND</b> mode 3 or 4, the base list does not match the base list passed in to <b>dbxbegin</b> .
8021	MULTIPLE TRANSACTION - NOT SUPPORTED BY SERVER	For <b>DBXBEGIN</b> mode 3 or 4, the list of bases are remote to a node that does not support multiple transactions.
8201	NODE PASSED TO NBRPC IS INVALID	The node number specified in the call to <b>NBRPC</b> is an invalid or undefined NetBase node.
8202	NODE PASSED TO NBRPC IS THE LOCAL NODE	The node number specified in the call to <b>NBRPC</b> defines the local node.
8203	BOUNDS VIOLATION ON PROCEDURE NAME	The procedure name passed to <b>NBRPC</b> has a bounds violation. Check and correct call to <b>NBRPC</b> .
8204	INVALID PROCEDURE NAME DELIMITER	The terminating delimiter could not be found or is invalid or the procedure name is too large. Check and correct call to <b>NBRPC</b> .
8205	BOUNDS VIOLATION ON XL NAME	The XL name passed to <b>NBRPC</b> has a bounds violation. Check and correct call to <b>NBRPC</b> .
8206	INVALID XL NAME DELIMITER	The terminating delimiter could not be found or is invalid or the XL name is too large. Check and correct call to <b>NBRPC</b> .

Appendix B

ERR#	Error Message	Explanation/Action
8207	SEND LENGTH PASSED TO NBRPC IS INVALID	The send length passed to <b>NBRPC</b> is less than <b>0</b> or greater than <b>8192</b> . Check and correct call to <b>NBRPC</b> .
8208	BOUNDS VIOLATION ON SEND BUFFER	The send buffer passed to <b>NBRPC</b> has a bounds violation. Check and correct call to <b>NBRPC</b> .
8209	RECEIVE LENGTH PASSED TO NBRPC IS INVALID	The receive length passed to <b>NBRPC</b> is less than <b>0</b> or greater than <b>8192</b> . Check and correct call to <b>NBRPC</b> .
8210	BOUNDS VIOLATION ON RECEIVE BUFFER	The receive buffer passed to <b>NBRPC</b> has a bounds violation. Check and correct call to <b>NBRPC</b> .
8211	NETBASE IS DISABLED, CANNOT SEND REMOTE PROCEDURE	The remote NetBase process is not available or local access has been disabled.
8212	ERROR LOCALIZING STRINGS IN NBCREATEPROCESS	One or more parameters passed by the local program to the <b>CREATEPROCESS</b> intrinsic is in error. Check all parameters and correct.
8213	LIBRARY NOT SPECIFIED FOR OMNIDEX CALLS	The OMNIDEX shell requires that the XL containing the OMNIDEX software be defined with the directory XL command.

# A P P E N D I X C

## Index

### A

ABORT command, NBCTRL	
Examples.....	2-9
Parameters	
CLIENT .....	2-8
EXPORT .....	2-8
IMPORT <i>node-id</i> .....	2-8
NETBASE.....	2-8
NODE <i>node-id</i> .....	2-8
SERVER <i>server-id</i> .....	2-8
SPOOL UNIX .....	2-8
SPOOL IN <i>node-id</i> .....	2-8
SPOOL OUT <i>node-id</i> .....	2-8
SPOOLING .....	2-8
SQL .....	2-8
Syntax .....	2-7
ABORT parameter, SERVER (NBCTRL) .....	2-21
Abort Post on Errors parameter, CONFIG .....	3-28
Aborting NetBase .....	2-7, 2-10
ACCESS command, NBSECURE .....	6-18
ACCOUNT command, NBDIR	
Examples.....	4-9
Parameters.....	4-9
Syntax .....	4-9
ACCOUNT command, NBSECURE .....	6-19
Account Password parameter, CONFIG.....	3-23
Acct parameter, LISTF (NBDIR).....	4-32
ACTIVE state, SHOW STATUS (NBCTRL).....	2-27
ACTIVE status, SHOW POST (NBCTRL).....	2-28
ADAGER	
Requirements .....	10-2
Structural changes/shadow databases .....	5-30, 10-2
ADD command, CONFIG	
Examples.....	3-31
Parameters.....	3-30
Syntax .....	3-30

ALL parameter	
LIST (CONFIG).....	3-33
LISTF (NBDIR) .....	4-30
LISTS (NBDIR) .....	4-37
MODIFY (CONFIG).....	3-37
PRINT (CONFIG).....	3-40
STATS (NBCTRL) .....	2-40
STOP (NBCTRL).....	2-44
ALLBASE DBEs, shadowing .....	xiv
ALLOW command, NBSECURE.....	6-20
Allow File Equates parameter, CONFIG.....	3-15
Allow incoming job streams parameter, CONFIG .....	3-25
ALLOW parameter, SERVER (NBCTRL) .....	2-21
Allow Wildcards parameter, CONFIG .....	3-16
ALT=node [,node] parameter, FILE (NBDIR).....	4-21
Alternate spooling device(s) .....	4-13
Applications, avoid modifying .....	iii
Archiving MPE files.....	13-10
ASCII, converting to.....	13-15
AutoRPM.....	4-2, 4-44
Defining a program for .....	8-3
Example.....	8-3
Load balancing.....	xii
NetBase RPM interface.....	xii
Notes .....	8-5
Options .....	8-6
Overview .....	8-2
Saving JCWs.....	xii
Transporting sessions.....	xii
AUTORPM parameter	
DISABLE (NBCTRL) .....	2-16
ENABLE (NBCTRL) .....	2-17
Avoiding application changes .....	iii

## Appendix C

---

### B

Background job, NetBase	
Data transport/communication .....	1-2
NBCOP.NB.NETBASE .....	1-2
NBLOG.DATA .....	1-3
NBLOG.PUB.NETBASE .....	1-3
Starting .....	1-2
Backlog information, SHOW POST (NBCTRL)....	2-28
Backups	
Checkpoint, NBFLUSH .....	5-37
Full backup .....	11-5
Partial backup .....	11-4
Using shadow data .....	5-36
Basename parameter, DSET (NBDIR) .....	4-17
Buffered Exporting parameter, CONFIG .....	3-28
Buffering	
Automatic feature .....	6-3
BUFONLY parameter, FILE (NBDIR) .....	4-21
NOBUF parameter, PROGRAM (NBDIR) .....	4-47
SHADOW-Export Buffer parameters .....	5-6
BUFFERING parameter	
DISABLE (NBCTRL) .....	2-16
ENABLE (NBCTRL) .....	2-17
BUFONLY parameter, FILE (NBDIR) .....	4-21
BUILD command, NBCTRL	
Examples .....	4-11
Parameters .....	4-11
BUILD command, NBDIR	
Syntax .....	4-11

### C

Change jsname to node name parameter, CONFIG	3-12
Change user on incoming parameter, CONFIG .....	3-12
Changing file limits .....	13-15
Changing the blocking factor .....	13-15
Check KSAM record counts parameter, CONFIG ..	3-17
CHGDIR command, NBCTRL	
Examples .....	2-11
Parameters .....	2-11
Syntax .....	2-11
class parameter	
(alternate) DEVICE (NBDIR) .....	4-13
(local) DEVICE (NBDIR) .....	4-12
(remote) DEVICE (NBDIR) .....	4-12
Client Access on Startup parameter, CONFIG .....	3-10
Client Inactivity Time-out parameter, CONFIG .....	3-9
Client Lock Held Time-out parameter, CONFIG...	3-10

CLIENT parameter	
ABORT (NBCTRL) .....	2-8
SHOW (NBCTRL) .....	2-25
START (NBCTRL) .....	2-37
STOP (NBCTRL) .....	2-44
Close Log file on suspend parameter, CONFIG .....	3-16
CLOSE parameter, FILE (NBDIR) .....	4-21
Closes information, SHOW POST (NBCTRL) .....	2-28
COBOLII .....	13-18
comarea parameter	
EXPORTEXIT .....	5-39
POSTUSEREXIT .....	5-45
POSTUSERINIT .....	5-43
CONFIG command, NBCTRL	
Entering CONFIG subsystem .....	2-12
Examples .....	2-12
Syntax .....	2-12
CONFIG parameter	
LIST (CONFIG) .....	3-33
MODIFY (CONFIG) .....	3-37
PRINT (CONFIG) .....	3-40
CONFIG subsystem	
Command summary .....	3-2
Commands	
ADD .....	3-30
DELETE .....	3-32
LIST .....	3-33
MODIFY .....	3-37
PRINT .....	3-40
Help facility .....	3-5
MPE commands .....	3-5
Node level parameters	
Node Connection Time-out .....	3-25
Node Name .....	3-24
SHADOW	
Abort Post on Errors .....	3-28
Buffered Exporting .....	3-28
Direction .....	3-26
Enable Post Logging .....	3-27
Export queue size .....	3-26
Import queue size .....	3-26
Initial Post Priority .....	3-27
Log all Post Transactions .....	3-27
Post Inactivity Timeout .....	3-27
Post queue size .....	3-27
Save Process Info on Post .....	3-28

- 
- SPOOL
    - Allow incoming job streams ..... 3-25
    - Direction ..... 3-25
    - Enable Export Flow Control ..... 3-26
    - Startup Access ..... 3-24
    - System Name ..... 3-24
  - Operational parameters
    - AUTORPM
      - Use Local Logon ..... 3-23
    - DIR
      - Allow File Equates ..... 3-15
      - Allow Wildcards ..... 3-16
      - Map Local Files with Dir ..... 3-16
    - GLOBAL
      - Error Enhancements ..... 3-7
      - Local Node Number ..... 3-7
      - Maximum Local Users ..... 3-8
    - NFA
      - Client Access on Startup ..... 3-10
      - Client Inactivity Time-out ..... 3-9
      - Client Lock Held Time-out ..... 3-10
      - Incoming Users on Startup ..... 3-10
      - Initial Server Allocation ..... 3-8
      - Lock Priority Boosting ..... 3-9
      - Maximum Remote Users ..... 3-8
      - Outgoing Users on Startup ..... 3-10
      - Release Server on Last Close ..... 3-8
      - Server Allocation Time-out ..... 3-9
      - Server Initial Priority ..... 3-9
      - Server LIB=load flag ..... 3-9
    - PASSWORD
      - Account Password ..... 3-23
      - Group Password ..... 3-23
      - User Password ..... 3-23
    - SHADOW
      - Check KSAM record counts ..... 3-17
      - Close Log file on suspend ..... 3-16
      - Disable file autoresync ..... 3-17
      - Disable Shift Messages ..... 3-20
      - Enable Logging ..... 3-21
      - Enable Quiet Period Flush ..... 3-18
      - Errors Before Disable ..... 3-17, 5-20
      - Export Ack Time-out ..... 3-19
      - Export Buffer Max BF ..... 3-19
      - Export Buffer size ..... 3-19
      - Export Buffer Time-out ..... 3-19
      - Export Downshift Trigger ..... 3-20
      - Export Initial Priority ..... 3-20
      - Export Log File Timeout ..... 3-21
      - Export on Startup ..... 3-22
      - Export Upshift Priority ..... 3-20
      - Export Upshift Trigger ..... 3-20
      - File Locking Required ..... 3-16
      - Flush Shadow File Posts ..... 3-17
      - Import Inactivity Time-out ..... 3-22
      - Import on Startup ..... 3-22
      - Log File Size ..... 3-21
      - Log Post Errors ..... 3-18
      - Post Flush Time-out ..... 3-22
      - Post on Startup ..... 3-22
      - Queue Full Warn Time-out ..... 3-21
      - Shift Check Interval ..... 3-20
      - Use IMAGE Log Files ..... 3-22, 5-53
    - SPOOL
      - Change jsname to node name ..... 3-12
      - Change user on incoming ..... 3-12
      - Maximum Block ..... 3-11
      - Maximum Files per Node ..... 3-11
      - Remote Transfer Time-out ..... 3-11
      - Spooling on Startup ..... 3-13
      - Transfer Server Allocation ..... 3-11
      - Use S' and J' on incoming ..... 3-12
    - SQL
      - Enable SQL Shadowing ..... 3-22
    - STATS
      - Expansion Increment ..... 3-14
      - Init File Allocation ..... 3-14
      - Log File Limit ..... 3-14
      - Low User Response Filter ..... 3-15
      - Max File Allocation ..... 3-14
      - Read-Read Filter ..... 3-14
      - Read-Write Filter ..... 3-15
  - Configuration
    - CONFIG, NBCTRL subsystem ..... 3-1
    - Configuration parameters ..... 3-6
    - Defining network nodes ..... 3-4
    - NetBase ..... 3-3
    - Network nodes ..... A-4
    - Operational parameters ..... 3-3
    - Parameters
      - Define NetBase operation environment ..... 3-6
      - Define network nodes ..... 3-6
    - QUICK parameters ..... 3-6
    - Remote spooling ..... 7-7
    - See also CONFIG*
    - Copies parameter, DEVICE (NBDIR) ..... 4-13
    - COPY option, NetBase Spooling ..... xi
    - COPY parameter, DEVICE (NBDIR) ..... 4-13

## Appendix C

---

copyfilename parameter, SHADOW (NBDIR) .....	4-53
Correcting load imbalance .....	iii
CRITICAL parameter	
DISABLE (NBCTRL) .....	2-16
ENABLE (NBCTRL) .....	2-17
<b>D</b>	
Databases	
Copying entire.....	13-18
Dataset shadowing .....	5-51
Protection.....	5-57
Resyncing .....	5-25
Root file.....	6-6
Shadowed, structural changes.....	5-30
Dataset parameter, DSET (NBDIR).....	4-17
Dataset shadowing	
DSET command.....	5-51
Example.....	5-51
Notes .....	5-52
dbname parameter	
SQLIN (NBDIR) .....	4-56
SQLOUT (NBDIR) .....	4-58
TABLE (NBDIR) .....	4-61
DBGENERAL	
Requirements .....	10-4
Structural changes/shadow databases.....	5-31, 10-4
DBINFO calls, FILE INFO .....	4-21, 6-6
DBLOCK .....	4-22
DBPUT .....	13-17
DBX parameter, SHADOW (NBDIR).....	4-53
DEBUG parameter, PROGRAM (NBDIR) .....	4-47
DEFER	
Shadow updates .....	4-36
DEFER command, NBCTRL	
Examples .....	2-13
Parameters	
fileset.....	2-13
indirectfile .....	2-13
node.....	2-13
Syntax.....	2-13
DELETE command, CONFIG	
Examples .....	3-32
Parameters .....	3-32
Syntax.....	3-32
DEVICE command, NBDIR	
Examples.....	4-13
Parameters	
class (alternate) .....	4-13
class (local) .....	4-12
class (remote).....	4-12
Copies .....	4-13
COPY .....	4-13
Device .....	4-13
ldev (alternate) .....	4-13
ldev (local) .....	4-12
ldev (remote).....	4-12
node (alternate) .....	4-13
node (remote) .....	4-12
outpri .....	4-13
TEMP .....	4-13
Syntax .....	4-12
Device parameter, DEVICE (NBDIR) .....	4-13
deviceset parameter, LISTD (NBDIR).....	4-26
DIR command, NBCTRL	
Examples.....	2-14
Syntax .....	2-14
DIR command, NBDIR	
Examples.....	4-15
Parameters.....	4-15
Syntax .....	4-15
DIR parameter	
LIST (CONFIG) .....	3-33
MODIFY (CONFIG) .....	3-37
PRINT (CONFIG) .....	3-40
Direction parameter, CONFIG .....	3-25, 3-26
Directory, NetBase	
Adding entries .....	1-7
Changes to file access .....	vii
Creating/adding a directory.....	4-11
Dump directory.....	4-63
Editing .....	4-6
KSAM image of directory .....	1-7
Locally accessed files.....	vii
NBD.DATA.NETBASE.....	1-7
NBD0.DATA.NETBASE, KSAM file.....	1-7
NBDIR program .....	1-7
Network manager .....	vii
Shadowing entries .....	5-12
Using wildcards.....	1-7
<i>See also NBDIR program</i>	

- 
- Disable
    - NetBase services ..... 2-6, 4-46
    - NetBase using NBVERS..... 13-3
    - NFA buffering ..... 4-47
  - DISABLE command, NBCTRL
    - Examples..... 2-16
    - Parameters
      - AUTORPM ..... 2-16
      - CRITICAL..... 2-16
      - PENDLOCK..... 2-16
      - SHADOWING..... 2-16
      - SPOOLING ..... 2-16
      - STATISTICS ..... 2-16
    - Syntax ..... 2-15
  - Disable file autoresync parameter, CONFIG..... 3-17
  - Disable Shift Messages parameter, CONFIG..... 3-20
  - Disaster recovery
    - Implementing scenarios ..... 12-10
    - Multiple copies ..... iii, vii
    - Procedures..... 12-1
  - Discard file updates ..... 4-37
  - Discard state, resync shadow file..... 5-23
  - Discarding(D), shadow file state ..... 5-19
  - DISPLAY command, NBDIR
    - Examples..... 4-16
    - Parameters
      - names..... 4-16
      - Numbers..... 4-16
    - Syntax ..... 4-16
  - Distaster tolerance ..... 7-4
  - Distributed systems..... iii
  - DOWN state, SHOW STATUS (NBCTRL)..... 2-27
  - DOWN status, SHOW POST (NBCTRL)..... 2-28
  - DSET command, NBDIR
    - Examples..... 4-17
    - Parameters
      - basename ..... 4-17
      - Dataset ..... 4-17
      - RESET ..... 4-17
    - Syntax ..... 4-17
  - DSET parameter, LISTF (NBDIR) ..... 4-30
  - DUALACC parameter, FILE (NBDIR)..... 4-21
  - Dump directory ..... 4-63
  - Dynamic rollback recovery, shadowing..... 5-57
- E**
- EBCDIC ..... 13-15, 13-18, 13-22
  - ENABLE command, NBCTRL
    - Examples ..... 2-18
    - Parameters
      - AUTORPM..... 2-17
      - BUFFERING ..... 2-17
      - CRITICAL ..... 2-17
      - PENDLOCK ..... 2-18
      - SHADOWING ..... 2-18
      - SPOOLING..... 2-18
      - STATISTICS ..... 2-18
    - Syntax ..... 2-17
  - Enable Export Flow Control parameter, CONFIG. 3-26
  - Enable Logging parameter, CONFIG ..... 3-21
  - Enable NetBase using NBVERS..... 13-3
  - Enable Post Logging parameter, CONFIG..... 3-27
  - Enable Quiet Period Flush parameter, CONFIG.... 3-18
  - Enable SQL Shadowing parameter, CONFIG..... 3-22
  - Encrypting database passwords..... 4-22
  - ERR parameter, LISTS (NBDIR)..... 4-37
  - Error Enhancements parameter, CONFIG ..... 3-7
  - Errors
    - Detected by AutoRPM..... B-22
    - Detected by NBSERVER ..... B-27
    - Detected by NetBase control program(NBCOP) B-25
    - Detected by NMSERVER ..... B-28
    - Detected by shell..... B-30
    - Detected by Spooling processes ..... B-26
    - Directory file (NBD) ..... B-16
    - File system..... B-11
    - Network configuration file (NBI)..... B-17
    - Network control file (NBN)..... B-18
    - Shadowing queue file (NBM)..... B-19
    - Spooling files (NBSEC/NBSED)..... B-21
    - Statistics logging file (NBVL)..... B-20
  - Errors Before Disable parameter, CONFIG..... 3-17
  - EXCLUDE parameter, FILE (NBDIR)..... 4-21
  - EXCLUDE parameter, SQLOUT (NBDIR)..... 4-59
  - EXECPRI=BSICSIDSIES parameter
    - SQLIN (NBDIR) ..... 4-56
    - SQLOUT (NBDIR) ..... 4-58
  - Expansion Increment parameter, CONFIG ..... 3-14
  - Export Ack Time-out parameter, CONFIG ..... 3-19
  - Export Buffer Max BF parameter, CONFIG ..... 3-19
  - Export Buffer parameters ..... 5-6
  - Export Buffer size parameter, CONFIG ..... 3-19
  - Export Buffer Time-out parameter, CONFIG ..... 3-19
  - Export Downshift Trigger parameter, CONFIG..... 3-20
  - Export Inactivity Time-out parameter, CONFIG.... 3-21
  - Export Initial Priority parameter, CONFIG ..... 3-20
  - Export Log File Timeout parameter, CONFIG ..... 3-21

## Appendix C

---

Export on Startup parameter, CONFIG.....	3-22	Node # .....	4-20
EXPORT parameter		Node name .....	4-20
ABORT (NBCTRL) .....	2-8	NOKFILES .....	4-22
SHOW (NBCTRL) .....	2-26	NOLOG .....	4-22
START (NBCTRL) .....	2-37	Process/parameter list .....	4-19
STOP (NBCTRL) .....	2-44	PROTECT .....	4-22
Export process, queues shadow updates .....	viii	READ .....	4-21
Export queue size parameter, CONFIG .....	3-26	READPASS=password .....	4-22
Export Upshift Priority parameter, CONFIG .....	3-20	Remote-name .....	4-20
Export Upshift Trigger parameter, CONFIG .....	3-20	Shadow .....	4-20, 4-22
Export user exits .....	5-38	SYNC .....	4-22
EXPORT=priority parameter, SET (NBCTRL) .....	2-22	SYNCERR=FAIL .....	4-23
EXPORTEXTIT		SYNCERR=REMOTE .....	4-23
Parameters		UPDPASS=password .....	4-23
comarea .....	5-39	Syntax .....	4-20, 4-24, 4-25
exportnodes .....	5-40	File Locking Required parameter, CONFIG .....	3-16
header .....	5-41	File system intrinsics .....	1-5
new data .....	5-41	filename parameter	
orig .....	5-41	FILE (NBDIR) .....	4-20
userarea .....	5-40	RESET (NBDIR) .....	4-51
validnodes .....	5-40	SHADOW (NBDIR) .....	4-53
Syntax .....	5-39	Files information, SHOW POST (NBCTRL) .....	2-28
exportnodes parameter, EXPORTEXTIT .....	5-40	FILES parameter, STATS (NBCTRL) .....	2-40
Extracting data set entries .....	13-15	fileset parameter	
<b>F</b>		LISTF (NBDIR) .....	4-30
fakedbe parameter		LISTP (NBDIR) .....	4-35
SQLIN (NBDIR) .....	4-57	LISTS (NBDIR) .....	4-37
SQLOUT (NBDIR) .....	4-59	filesize parameter, LIMIT (NBDIR) .....	4-25
Fault-tolerant environment .....	iii	Flush Shadow File Posts parameter, CONFIG .....	3-17
FCOPY .....	13-16	FOS upgrades .....	11-7
FILE command, NBDIR		Fourth GLs .....	6-6
Examples .....	4-23, 4-24, 4-25	Fourth GLs, FILE INFO .....	4-21
Parameters .....	4-24, 4-25	Functionality, NetBase	
ALT=node [,node] .....	4-21	Disaster recovery .....	iii
BUFONLY .....	4-21	Distribution performance .....	iii
CLOSE .....	4-21	Maintaining multiple copies .....	iii
DUALACC .....	4-21	Minimizing spooling overhead .....	iv
EXCLUDE .....	4-21	<b>G</b>	
filename .....	4-20	GLOBAL parameter	
IMAGELOG .....	4-21	LIST (CONFIG) .....	3-33
INCLUDE .....	4-21	MODIFY (CONFIG) .....	3-37
INFO .....	4-21	PRINT (CONFIG) .....	3-40
LOCAL .....	4-21	GROUP command, NBSECURE .....	6-22
LOCALACC .....	4-21	Group Password parameter, CONFIG .....	3-23
LOCK .....	4-21	groupname parameter, GROUP (NBDIR) .....	4-24
LOGONLY .....	4-22		
MAP .....	4-22		
MASTER .....	4-21		

## H

header parameter  
 EXPORTEXTIT..... 5-41  
 POSTUSEREXIT ..... 5-46  
 HELP facility, NBCTRL ..... 2-4  
 home-partition parameter, SLOUT (NBDIR) ..... 4-58

## I

I/O intrinsic counts/timings, statistics..... 9-2  
 IDLE status, SHOW POST (NBCTRL) ..... 2-28  
 ILOG [file] parameter  
 START (NBCTRL) ..... 2-37  
 STOP (NBCTRL) ..... 2-44  
 IMAGE data set shadowing..... 4-21  
 IMAGE intrinsics..... 1-5  
 IMAGE parameter, STATS (NBCTRL)..... 2-40  
 IMAGELOG parameter, FILE (NBDIR)..... 4-21  
 IMPORT [node] parameter  
 START (NBCTRL) ..... 2-37  
 STOP (NBCTRL) ..... 2-44  
 Import Inactivity Time-out parameter, CONFIG .... 3-22  
 IMPORT *node-id* parameter, ABORT (NBCTRL).... 2-8  
 Import on Startup parameter, CONFIG..... 3-22  
 IMPORT parameter, SHOW (NBCTRL) ..... 2-26  
 Import queue size parameter, CONFIG..... 3-26  
 in sync(--), shadow file state ..... 5-19  
 INACTIVE state, SHOW STATUS (NBCTRL)..... 2-27  
 INCLUDE parameter  
 FILE (NBDIR)..... 4-21  
 SLOUT (NBDIR)..... 4-59  
 Incoming Users on Startup parameter, CONFIG .... 3-10  
 INFO parameter  
 FILE (NBDIR)..... 4-21  
 LISTS (NBDIR)..... 4-37  
 RUN..... 2-3  
 Init File Allocation parameter, CONFIG..... 3-14  
 Initial Post Priority parameter, CONFIG..... 3-27  
 Initial Server Allocation parameter, CONFIG ..... 3-8  
 Installation  
 Available configuration parameters ..... 3-6  
 Before installing ..... A-1  
 Configuring network nodes ..... A-4  
 Convert from account to global ..... A-13  
 Global installation ..... A-11  
 NetBase shell ..... A-7  
 Overview ..... A-1  
 Steps for installing ..... A-3  
 System startup procedures ..... A-11

Update from pre-9.6 global..... A-13  
 Updating from 9.6 global..... A-2  
 Validate/modify startup..... A-9  
 Intercepting calls ..... vii  
 Intrinsic  
 File system..... 1-5  
 IMAGE ..... 1-5  
 OMNIDEX ..... 6-8  
 Other ..... 1-5

## J

JCWSTATUS parameter, SHOW (NBCTRL) ..... 2-26

## K

KSAM file ..... 13-15  
 Copying ..... 13-22  
 NOKSAM access ..... 13-16  
 REBUILD command ..... 4-50

## L

ldev parameter  
 (alternate) DEVICE (NBDIR) ..... 4-13  
 (local) DEVICE (NBDIR) ..... 4-12  
 (remote) DEVICE (NBDIR) ..... 4-12  
 LIB= parameter, PROGRAM (NBDIR) ..... 4-47  
 LIST command, CONFIG  
 Examples ..... 3-34  
 Parameters  
 ALL ..... 3-33  
 CONFIG ..... 3-33  
 DIR ..... 3-33  
 GLOBAL ..... 3-33  
 NFA ..... 3-33  
 NODES ..... 3-33  
 QUICK ..... 3-33  
 SHADOW ..... 3-33  
 SPOOL ..... 3-33  
 SQL ..... 3-33  
 STATS ..... 3-33  
 Syntax ..... 3-33  
 LISTD command, NBDIR  
 Examples ..... 4-27  
 Fields ..... 4-26  
 Parameters ..... 4-26  
 Syntax ..... 4-26



- 
- Maximum Local Users parameter, CONFIG..... 3-8
  - Maximum Remote Users parameter, CONFIG ..... 3-8
  - MGR.NETBASE user password..... 3-12
  - Minimizing network transactions ..... v
  - Minimizing spooling overhead ..... iv
  - MODE1 parameter, SHADOW (NBDIR) ..... 4-53
  - MODIFY command, CONFIG
    - Examples..... 3-38
    - Parameters
      - ALL ..... 3-37
      - CONFIG..... 3-37
      - DIR ..... 3-37
      - GLOBAL..... 3-37
      - NFA ..... 3-37
      - NODES..... 3-37
      - QUICK..... 3-37
      - SHADOW ..... 3-37
      - SPOOL..... 3-37
      - SQL ..... 3-37
      - STATS ..... 3-37
    - Passwords, modifying ..... 3-38
    - Special characters..... 3-38
    - Syntax ..... 3-37, 3-40
  - MPE
    - Archiving files..... 13-10
    - Builds/purges/renames..... 0-9
    - Commands ..... 3-5
    - Execute from CONFIG ..... 3-5
    - NetBase shell, passes to..... 1-4
  - MPEX
    - Commands ..... 10-7
    - Notes..... 10-7
    - Requirements ..... 10-7
  - N**
    - Name information, SHOW POST (NBCTRL) ..... 2-28
    - names parameter, DISPLAY (NBDIR) ..... 4-16
    - NBCOP.NB.NETBASE ..... 1-2
    - NBCOPY utility ..... 13-15
    - NBCTRL
      - Accessing subsystems ..... 2-5
      - Command summary ..... 2-2
      - Commands
        - ABORT ..... 2-7
        - CHGDIR ..... 2-11
        - CONFIG..... 2-12
        - DEFER..... 2-13
        - DIR ..... 2-14
        - DISABLE ..... 2-15
        - ENABLE ..... 2-17
        - ROLL FORWARD..... 2-19
        - SECURITY ..... 2-20
        - SERVER..... 2-21
        - SET ..... 2-22
        - SHOW ..... 2-25
        - SHOW CLIENT ..... 2-31
        - SHOW EXPORT ..... 2-31
        - SHOW IMPORT ..... 2-32
        - SHOW LOCAL..... 2-30
        - SHOW POST ..... 2-28, 2-32, 0-2
        - SHOW POST <import node> ..... 2-29
        - SHOW REMOTE ..... 2-31
        - SHOW SERVICES ..... 2-33
        - SHOW SQL..... 2-33
        - SHOW STATUS..... 2-26
        - SHOW VARSTATUS ..... 2-34
        - START ..... 2-36
        - STATS..... 2-39, 9-2
        - STOP..... 2-43
        - TRACE..... 2-46
        - UNDEFER..... 2-48
      - CONFIG subsystem..... 3-1
      - HELP facility ..... 2-4
      - Implementing NetBase UDCs..... 2-3
      - MPE commands ..... 2-4
      - NBDISABLE JCW ..... 2-6
      - Overview ..... 1-8
      - REDO command, using ..... 2-4
      - RUN command, MPE..... 2-3
      - Running NBCTRL ..... 2-3
    - NBD.DATA.NETBASE ..... 1-7
    - NBD0.DATA.NETBASE, KSAM file..... 1-7
    - NBDIR
      - Command summary..... 4-2
      - Commands
        - ACCOUNT ..... 4-9
        - BUILD..... 4-11
        - DEVICE ..... 4-12
        - DIR..... 4-15
        - DISPLAY ..... 4-16
        - DSET..... 4-17
        - FILE ..... 4-19
        - GROUP ..... 4-24
        - LIMIT..... 4-25
        - LISTD..... 4-26
        - LISTF ..... 4-28
        - LISTL ..... 4-32
        - LISTP ..... 4-34
        - LISTS ..... 4-36

## Appendix C

---

LISTSQLI .....	4-39	LISTSEC .....	6-23
LISTSQLO .....	4-41	NODE .....	6-24
LISTX .....	4-43	REMOVE .....	6-25
LOGON .....	4-44	Running .....	6-17
PROGRAM .....	4-46	Special commands .....	6-17
REBUILD .....	4-50	NBSpool program	
RESET .....	4-51	Features .....	7-10
SHADOW .....	4-53	Max number of spool files .....	7-10
SQLIN .....	4-56	NBSTATLG, statistics log files .....	9-2
SQLOUT .....	4-58	NBSYNC utility	
TABLE .....	4-61	Entry points .....	5-28
UNLOAD .....	4-63	Examples .....	5-24
XL .....	4-64	Resyncing shadow files .....	5-23
Editing a NetBase directory .....	4-6	Sample job .....	5-22
Running NBDIR .....	4-4	Syntax .....	5-28
Special command features .....	4-8	VERIFY .....	5-20
Using wildcards .....	4-5	NBSYNC, VERIFY .....	B-8
NBDISABLE JCW		NBSYNC.PUB.NETBASE .....	5-20
Disabling NetBase services .....	2-6	NBTRACE utility .....	13-8
Syntax .....	2-6	NBVERS utility .....	13-2
NBFLUSH		NetBase	
Entry points .....	5-33	Avoiding application changes .....	1-4
Syntax .....	5-33	Components	
Values .....	5-36	Background job .....	1-2
NBLOG .....	5-53	Directory .....	vii, 4-1
NBLGDUMP .....	9-4	NetBase control program, NBCTRL .....	1-8
NBLGTRAN		NetBase directory .....	1-7
Example .....	9-15	NetBase shell .....	1-4
Record types .....	9-13	Correcting load imbalance .....	iii
Running .....	9-14	Disaster recovery .....	iii, 12-1
Scenario file format .....	9-16	Distribution performance .....	iii
NBLOG utility .....	13-5	Export queues shadow updates .....	viii
NBLOG.DATA logfile .....	1-2	Fault-tolerant environment .....	iii
NBLOG.PUB.NETBASE .....	1-3	File access .....	vii
NBM.IPC.NETBASE .....	2-44	Improving data access .....	vi
NBPOST .....	5-42	Saving system resources .....	vi
NBRPC		Services	
Parameters .....	6-12	AutoRPM .....	xii, 8-1
Syntax .....	6-12	NetBase Client .....	xv
NBRPT01 .....	9-7	NetBase NFA .....	ix, 6-1
NBRPT02 .....	9-10	NetBase Shadowing .....	vii
NBSECURE		NetBase Spooling .....	x, 7-1
Command summary .....	6-16	NetBase SQL Shadowing .....	xiv
Commands		NetBase Statistics .....	xiii, 9-1
ACCESS .....	6-18	Shadowing .....	5-1
ACCOUNT .....	6-19	Structural changes/shadow databases .....	5-30
ALLOW .....	6-20	System operations/maintenance .....	11-1
GROUP .....	6-22	Utilities .....	13-1
		NETBASE account .....	3-1, 3-3
		NETBASE account password .....	3-23

- 
- NetBase Client
    - Access for applications ..... xv
    - Performance increases ..... xv
  - NetBase directory ..... *See Directory, NetBase*
  - NETBASE parameter
    - ABORT (NBCTRL)..... 2-8
    - START (NBCTRL) ..... 2-37
    - STOP (NBCTRL) ..... 2-44
  - NetBase shell
    - Avoiding application changes..... 1-4
    - Background job, independant of..... 1-4
    - Error codes greater than 1000 ..... 1-5
    - File system intrinsics ..... 1-5
    - IMAGE intrinsics ..... 1-5
    - Installing ..... A-7
    - OMNIDEX enhancement ..... 6-8
    - Other intrinsics ..... 1-5
    - Trapping program activity..... 1-4
  - Network
    - Configuring nodes ..... A-4
    - Directory of available files/databases ..... iii
    - Failure option, COPY ..... xi
    - HP systems ..... iii
    - Monitoring NFA ..... 6-9
    - Spooling ..... x
  - Network File Access ..... *See NFA, NetBase*
  - new data parameter
    - EXPORTEXTIT ..... 5-41
    - POSTUSEREXIT ..... 5-47
  - NFA parameter
    - LIST (CONFIG) ..... 3-33
    - MODIFY (CONFIG) ..... 3-37
    - PRINT (CONFIG) ..... 3-40
  - NFA, NetBase
    - Buffering ..... 6-3
    - Configuring ..... 6-4
    - Defining a file for NFA..... 6-5
    - Defining NFA parameters ..... 3-6, 6-4
    - Examples..... 6-6, 6-9
    - FILE options..... 6-5
    - Monitoring ..... 6-9
    - NBCTRL commands..... 6-9
    - NFA security ..... 6-16
    - Overview..... 6-2
    - Remote procedure calls..... 6-11
    - Technical support tip ..... B-6
    - Using OMNIDEX ..... 6-8
  - NOBUF parameter, PROGRAM (NBDIR) ..... 4-47
  - NOBUF/NOMR access, preventing ..... 4-21
  - Node # parameter, FILE (NBDIR) ..... 4-20
  - NODE [node] parameter, START (NBCTRL)..... 2-37
  - NODE command, NBSECURE..... 6-24
  - Node Connection Time-out parameter, CONFIG... 3-25
  - Node information, SHOW POST (NBCTRL)..... 2-28
  - Node Name parameter, CONFIG ..... 3-24
  - Node name parameter, FILE (NBDIR) ..... 4-20
  - NODE *node-id* parameter, ABORT (NBCTRL) ..... 2-8
  - node parameter
    - LISTF (NBDIR) ..... 4-32
    - LOGON (NBDIR)..... 4-44
    - SQLIN (NBDIR) ..... 4-56
    - SQLOUT (NBDIR) ..... 4-58
    - TABLE (NBDIR) ..... 4-61
  - node parameter, (alternate) DEVICE (NBDIR) ..... 4-13
  - node parameter, (remote) DEVICE (NBDIR) ..... 4-12
  - nodeld parameter, PROGRAM (NBDIR) ..... 4-48
  - NODES parameter
    - LIST (CONFIG) ..... 3-33
    - MODIFY (CONFIG) ..... 3-37
    - PRINT (CONFIG) ..... 3-40
  - NOEQ parameter, PROGRAM (NBDIR) ..... 4-48, 8-6
  - NOJCW parameter, PROGRAM (NBDIR) .... 4-48, 8-6
  - NOKFILES parameter, FILE (NBDIR) ..... 4-22
  - NOKSAM ..... 13-16
  - NOLOG parameter, FILE (NBDIR) ..... 4-22
  - NONB parameter, PROGRAM (NBDIR) ..... 4-47
  - NOPEND parameter, PROGRAM (NBDIR) ..... 4-47
  - NOREL parameter, PROGRAM (NBDIR) ..... 4-47
  - NOSTAT parameter, PROGRAM (NBDIR) ..... 4-47
  - Numbers parameter, DISPLAY (NBDIR) ..... 4-16
- O**
- old data parameter, POSTUSEREXIT ..... 5-47
  - OMNIDEX
    - Accessing a remote database ..... 10-5
    - Installation for shadowing ..... 10-6
    - Installation requirements ..... 10-5
    - NetBase NFA ..... 6-8
  - Operations..... *See System operations/maintenance*
  - orig parameter, EXPORTEXTIT ..... 5-41
  - ORIGDBE parameter, SQLIN (NBDIR)..... 4-57
  - Out of sync(S), shadow file state ..... 5-19
  - Outgoing Users on Startup parameter, CONFIG... 3-10
  - outpri parameter, DEVICE (NBDIR) ..... 4-13
  - OVERRIDE parameter, SHADOW (NBDIR)..... 4-54

## Appendix C

---

### P

PARM= parameter, PROGRAM (NBDIR).....	4-47
PART=part parameter	
SQLIN (NBDIR) .....	4-57
SQLOUT (NBDIR) .....	4-59
PART=partition parameter, SQLOUT (NBDIR)....	4-58
PASS parameter, LISTS (NBDIR) .....	4-37
PASS=password parameter, SHADOW (NBDIR) .	4-54
Password	
Account.....	3-23
Case sensitivity.....	4-23
Changing passwords in the NetBase job .....	3-23
Defining the node password .....	3-30
Encrypt database passwords .....	4-22
Group .....	3-23
MGR.NETBASE.....	3-12
Modifying .....	3-38
Read access.....	4-22
Update access .....	4-23
User .....	3-23
Write access.....	4-23
Patches, HP.....	11-6
Pending locks, disable.....	4-22
PENDLOCK parameter	
DISABLE (NBCTRL) .....	2-16
ENABLE (NBCTRL) .....	2-18
Performance increases.....	vi
Performance, NetBase	
Anticipatory reading.....	v
Improving data access .....	vi
Minimizing overhead costs.....	v
NetBase buffering area.....	v
Network spooling.....	x
Performance increases.....	v, vi
Piggybacking calls.....	v
Predictability.....	v
Saving system resources.....	vi
Servicing remote calls.....	v
Transporting requests.....	v
POST [import node] parameter, SHOW (NBCTRL)2-26	
POST [node] parameter	
START (NBCTRL).....	2-37
STOP (NBCTRL).....	2-44
Post Flush Time-out parameter, CONFIG .....	3-22
Post Inactivity Timeout parameter, CONFIG.....	3-27
POST node=priority parameter, SET (NBCTRL) ..	2-22
Post on Startup parameter, CONFIG.....	3-22
Post queue file.....	4-36
Post queue size parameter, CONFIG.....	3-27
Post user exits.....	5-42
Posting process	
Deferred .....	4-36
Queues changes .....	viii
POSTUSEREXIT	
Parameters	
comarea.....	5-45
header .....	5-46
new data.....	5-47
old data.....	5-47
userarea.....	5-46
Syntax .....	5-44
POSTUSERINIT	
Parameters	
comarea.....	5-43
Syntax .....	5-42
PRINT command, CONFIG	
Examples.....	3-41
Parameters	
ALL .....	3-40
CONFIG.....	3-40
DIR .....	3-40
GLOBAL.....	3-40
NFA .....	3-40
NODES .....	3-40
QUICK.....	3-40
SHADOW .....	3-40
SPOOL.....	3-40
SQL .....	3-40
STATS .....	3-40
procname	
Parameters.....	6-13
Syntax .....	6-13
programe parameter, PROGRAM (NBDIR).....	4-47
PROGRAM command, NBDIR	
Examples.....	4-48, 4-50
Parameters.....	4-50
DEBUG.....	4-47
LIB=.....	4-47
LOCAL .....	4-47
MAP .....	4-47
NOBUF .....	4-47
nodeld .....	4-48
NOEQ .....	4-48
NOJCW.....	4-48
NONB .....	4-47
NOPEND .....	4-47
NOREL.....	4-47
NOSTAT.....	4-47

- 
- PARM=..... 4-47
  - progname ..... 4-47
  - remoteprog ..... 4-47
  - SHADOW ..... 4-48
  - TRACE ..... 4-48
  - XL= ..... 4-47
  - Syntax ..... 4-46, 4-50
  - Prompts
    - CONFIG ..... 2-5
    - NBCTRL ..... 2-5
    - NBDIR ..... 2-5
    - NBSECURE ..... 2-5
  - PROTECT parameter, FILE (NBDIR) ..... 4-22
  - PUB.NETBASE group password ..... 3-23
  - Q**
    - QCINPUT ..... 13-19
    - QCOUTPUT ..... 13-17
    - QUARK utility ..... 13-10
    - QUARK+ utility ..... 13-11
    - QUERY program, FILE INFO ..... 4-21, 6-6
    - Queue Full Warn Time-out parameter, CONFIG ... 3-21
    - Queue state, recover shadow file ..... 5-22
    - QUEUE state, SHOW STATUS (NBCTRL) ..... 2-27
    - Queuing(Q), shadow file state ..... 5-19
    - QUICK parameter
      - LIST (CONFIG) ..... 3-33
      - MODIFY (CONFIG) ..... 3-37
      - PRINT (CONFIG) ..... 3-40
  - R**
    - READ parameter, FILE (NBDIR) ..... 4-21
    - READPASS=password parameter, FILE (NBDIR) 4-22
    - Read-Read Filter parameter, CONFIG ..... 3-14
    - Read-Write Filter parameter, CONFIG ..... 3-15
    - Reblocking files ..... 13-15
    - recordtype parameter, RESET (NBDIR) ..... 4-51
    - Redirect spool output ..... 4-12
    - REDO command, using ..... 2-4
    - Release Server on Last Close parameter ..... 3-8, 4-47
    - REMOTE [n] parameter, SHOW (NBCTRL) ..... 2-25
    - REMOTE [node] parameter
      - START (NBCTRL) ..... 2-37
      - STOP (NBCTRL) ..... 2-43
    - Remote device definitions ..... 4-12
    - Remote name ..... 4-19
    - REMOTE parameter, STATS (NBCTRL) ..... 2-40
    - Remote procedure calls ..... 6-11
    - Remote process management
      - LISTP command, NBDIR ..... 4-34
      - LISTX command ..... 4-43
      - LOGON command ..... 4-44
      - PROGRAM command ..... 4-46
      - XL command ..... 4-64
    - Remote Transfer Time-out parameter, CONFIG... 3-11
    - Remote user information ..... 2-31
    - REMOTE=limit parameter, SET (NBCTRL) ..... 2-22
    - Remote-name parameter, FILE (NBDIR) ..... 4-20
    - remoteprog parameter, PROGRAM (NBDIR) ..... 4-47
    - REMOVE command, NBSECURE ..... 6-25
    - Removing NetBase ..... 11-9
    - Rename remote file ..... 4-22
    - RESET command, NBDIR
      - Examples ..... 4-52
      - Parameters
        - filename ..... 4-51
        - recordtype ..... 4-51
        - Syntax ..... 4-51
    - RESET parameter
      - DSET (NBDIR) ..... 4-17
      - SQLOUT (NBDIR) ..... 4-59
      - TABLE (NBDIR) ..... 4-61
    - RESTORE utility
      - Examples ..... 5-24
      - Resyncing shadow files ..... 5-23
    - RESUME parameter, STATS (NBCTRL) ..... 2-40
    - Resyncing shadow files
      - Examples ..... 5-24
      - Reset directory flags ..... 5-26
      - Steps ..... 5-23
      - Using DSCOPY ..... 5-27
    - ROLL FORWARD command, NBCTRL
      - Examples ..... 2-19
      - Parameters ..... 2-19
      - Syntax ..... 2-19
    - ROLL FORWARD statistic, SHOW POST ..... 2-29
    - Root file, database ..... 6-6
    - RUN command, MPE ..... 2-3
  - S**
    - Save Process Info on Post parameter, CONFIG ..... 3-28
    - SECURITY command, NBCTRL
      - Examples ..... 2-20
      - Syntax ..... 2-20
    - Security maintenance ..... *See NBSECURE*
    - SERVER [n] parameter, SHOW (NBCTRL) ..... 2-25

## Appendix C

---

SERVER command, NBCTRL	
Examples .....	2-21
Parameters	
ABORT .....	2-21
ALLOW .....	2-21
Syntax .....	2-21
Server Initial Priority parameter, CONFIG .....	3-9
Server LIB= load flag parameter, CONFIG .....	3-9
SERVER <i>server-id</i> parameter, ABORT (NBCTRL) .....	2-8
SERVERS=allocation parameter, SET (NBCTRL) .....	2-22
SERVICES parameter, SHOW (NBCTRL) .....	2-26
SET command, NBCTRL .....	2-22
Examples .....	2-23
Parameters	
EXPORT=priority .....	2-22
LOCAL=limit .....	2-22
POST node=priority .....	2-22
REMOTE=limit .....	2-22
SERVERS=allocation .....	2-22
Syntax .....	2-22
SHADOW command, NBDIR	
Examples .....	4-54
Options .....	5-11
Parameters	
copyfilename .....	4-53
DBX .....	4-53
filename .....	4-53
MODE1 .....	4-53
OVERRIDE .....	4-54
PASS=password .....	4-54
USEREXIT .....	4-54
Syntax .....	4-53
Shadow file, display information	
Discard information .....	2-29
Name of file .....	2-29
Number of updates applied to file .....	2-29
Synchronization status .....	2-29
Time/date of last record .....	2-29
Type of file .....	2-29
Updates queue .....	2-29
Shadow files	
Check for synchronization .....	5-21
Checking the states .....	5-20
Sample NBSYNC job .....	5-22
States .....	5-19, 5-22, 5-23
VERJCW values .....	5-21
SHADOW parameter	
FILE (NBDIR) .....	4-20, 4-22
LIST (CONFIG) .....	3-33
MODIFY (CONFIG) .....	3-37
PRINT (CONFIG) .....	3-40
PROGRAM (NBDIR) .....	4-48
Shadow system	
comarea fields, EXPORTEXTIT .....	5-40
comarea fields, POSTUSERINIT .....	5-44
FILE options .....	5-9
Monitoring shadowing .....	5-17
NetBase shadowing .....	5-4
Post user exits .....	5-42
Resyncing shadow files .....	5-23
Setting up shadowing .....	5-16
Structural changes/shadow databases .....	5-32
SHADOWING parameter	
DISABLE (NBCTRL) .....	2-16
ENABLE (NBCTRL) .....	2-18
Shadowing, NetBase	
Backups .....	vii, viii
Backups using shadow data .....	5-36
Checking the state of shadow files .....	5-20
Command options .....	5-11
Configuration parameters .....	5-5
Database protection .....	5-57
Databases, structural changes .....	5-30
Dataset shadowing .....	5-51
Defining a file for shadowing .....	5-8
Defining SHADOW parameters .....	3-6, 5-5
Disabling .....	2-15, 2-17
Disaster recovery .....	vii
Examples of directory entries .....	5-12
Examples of resyncing .....	5-24
Export Buffer parameters .....	5-6
FILE command .....	4-19
FILE options for master system .....	5-9
FILE options for shadow system .....	5-9
File types, all .....	viii
Flushing shadow updates .....	5-33
Hardware failure .....	viii
IMAGE data set .....	4-21
Maintaining concurrent copies .....	vii
Master system .....	5-2
Monitoring .....	5-17
NBDIR entries .....	5-8
NFA integration .....	viii
Process overview .....	5-2
Reducing I/O contention .....	vii
Reset logging process .....	5-56

- 
- Setting up ..... 5-16
  - SHADOW command ..... 4-53
  - Shadow computer down ..... viii
  - Shadow system ..... 5-4
  - Shadow transport ..... 5-2
  - Shadowing TurboIMAGE log files ..... 5-53
  - SQLIN command ..... 4-56
  - SQLOUT command ..... 4-58
  - States of shadow files ..... 5-19, 5-22
  - Synchronization loss ..... 5-20, 5-22
  - TPS for TurboIMAGE databases ..... 5-14
  - Updating shadow copies ..... viii
  - User exits ..... 5-38
  - With dynamic rollback recovery ..... 5-57
  - shadownode, FILE command ..... 5-8
  - Shared access ..... 4-37
  - Shell ..... *See NetBase shell*
  - Shift Check Interval parameter, CONFIG ..... 3-20
  - SHOW command, NBCTRL
    - Parameters
      - CLIENT [n] ..... 2-25
      - EXPORT ..... 2-26
      - IMPORT ..... 2-26
      - JCWSTATUS ..... 2-26
      - LOCAL [n] ..... 2-25
      - POST [import node] ..... 2-26
      - REMOTE [n] ..... 2-25
      - SERVER [n] ..... 2-25
      - SERVICES ..... 2-26
      - SQL ..... 2-26
      - STATUS ..... 2-26
      - VARSTATUS ..... 2-26
    - Syntax ..... 2-25
  - SHOW parameter, STATS (NBCTRL) ..... 2-40
  - SHOW POST <import node> command, NBCTRL
    - Display Information
      - DIS ..... 2-29
      - File ..... 2-29
      - QUE ..... 2-29
      - SYN ..... 2-29
      - Time Stamp ..... 2-29
      - Type ..... 2-29
      - Updates ..... 2-29
    - Examples ..... 2-30
  - SHOW POST command, NBCTRL
    - Display information
      - ACTIVE status ..... 2-28
      - Backlog ..... 2-28
      - Closes ..... 2-28
      - DOWN status ..... 2-28
      - Files ..... 2-28
      - IDLE status ..... 2-28
      - Name ..... 2-28
      - Node ..... 2-28
    - ROLL FORWARD statistic ..... 2-29
  - SHOW STATUS command, NBCTRL
    - Examples ..... 2-27
    - States
      - (blank) ..... 2-27
      - ACTIVE ..... 2-27
      - DOWN ..... 2-27
      - INACTIVE ..... 2-27
      - QUEUE ..... 2-27
      - STOP ..... 2-27
      - WAIT ..... 2-27
      - XFER ..... 2-27
  - Shutdown system ..... 11-3
  - Spool files, maximum number ..... 7-10
  - SPOOL IN parameter
    - ABORT (NBCTRL) ..... 2-8
    - START (NBCTRL) ..... 2-37
    - STOP (NBCTRL) ..... 2-44
  - SPOOL OUT parameter
    - ABORT (NBCTRL) ..... 2-8
    - START (NBCTRL) ..... 2-38
    - STOP (NBCTRL) ..... 2-44
  - SPOOL parameter
    - LIST (CONFIG) ..... 3-33
    - MODIFY (CONFIG) ..... 3-37
    - PRINT (CONFIG) ..... 3-40
  - SPOOL UNIX parameter
    - ABORT (NBCTRL) ..... 2-8
    - START (NBCTRL) ..... 2-38
    - STOP (NBCTRL) ..... 2-44
  - Spooling on Startup parameter, CONFIG ..... 3-13
  - SPOOLING parameter
    - ABORT (NBCTRL) ..... 2-8
    - DISABLE (NBCTRL) ..... 2-16
    - ENABLE (NBCTRL) ..... 2-18
    - START (NBCTRL) ..... 2-37
    - STOP (NBCTRL) ..... 2-44

## Appendix C

---

Spooling, NetBase	
Backup copies .....	xi
Creating local copies .....	xi
Defining SPOOL parameters .....	3-6, 7-7
LISTD command .....	4-26
Multiple output lines .....	x
NBSpool program .....	7-10
Option for network failure .....	xi
Overview .....	7-2
Performance increases .....	x
Reducing disk I/O .....	x
Remote spooling	
Configuring .....	7-7
DEVICE command .....	7-3
Disaster tolerance .....	7-4
Example .....	7-8
Operating principles .....	7-3
Setting up .....	7-6
Using pseudo devices .....	7-8
Sharing remote devices .....	x
Transferring spool files .....	x
SQL IN parameter	
START (NBCTRL) .....	2-38
STOP (NBCTRL) .....	2-44
SQL OUT parameter	
START (NBCTRL) .....	2-38
STOP (NBCTRL) .....	2-44
SQL parameter	
ABORT (NBCTRL) .....	2-8
LIST (CONFIG) .....	3-33
MODIFY (CONFIG) .....	3-37
PRINT (CONFIG) .....	3-40
SHOW (NBCTRL) .....	2-26
START (NBCTRL) .....	2-38
STOP (NBCTRL) .....	2-44
SQL shadowing	
Processing, additional .....	xiv
Tables .....	xiv
Vertical partitioning .....	xiv
SQLIN command, NBDIR	
Examples .....	4-57
Parameters	
dbname .....	4-56
EXECPRI=BSICSIDSIES .....	4-56
fakedbe .....	4-57
node .....	4-56
ORIGDBE .....	4-57
PART=part .....	4-57
transaction-priority .....	4-56
USEREXIT .....	4-57
Syntax .....	4-56
SQLOUT command, NBDIR	
Examples .....	4-59
Parameters	
dbname .....	4-58
EXCLUDE .....	4-59
EXECPRI=BSICSIDSIES .....	4-58
fakedbe .....	4-59
home-partition .....	4-58
INCLUDE .....	4-59
node .....	4-58
PART=part .....	4-59
PART=partition .....	4-58
RESET .....	4-59
USEREXIT .....	4-59
wakeup-seconds .....	4-58
Syntax .....	4-58
START command, NBCTRL	2-36
Examples .....	2-38
Parameters	
CLIENT .....	2-37
EXPORT [node] .....	2-37
ILOG [file] .....	2-37
IMPORT [node] .....	2-37
LOCAL [node] .....	2-37
NETBASE .....	2-37
NODE [node] .....	2-37
POST [node] .....	2-37
REMOTE [node] .....	2-37
SPOOL IN .....	2-37
SPOOL OUT .....	2-38
SPOOL UNIX .....	2-38
SPOOLING .....	2-37
SQL .....	2-38
SQL IN node .....	2-38
SQL OUT node .....	2-38
Syntax .....	2-36
START parameter, STATS (NBCTRL) .....	2-40
Startup Access parameter, CONFIG .....	3-24
Startup system .....	11-2
STATISTICS parameter, DISABLE (NBCTRL) .....	2-16
STATISTICS parameter, ENABLE (NBCTRL) .....	2-18
Statistics records .....	9-2

- 
- Statistics, NetBase
    - Collecting statistics..... 9-2
    - Consolidating records ..... 9-17
    - Defining STATS parameters ..... 3-6, 9-2
    - Generating reports ..... xiii
    - Generating scenarios..... 9-13
    - Minimal overhead incurred..... xiii
    - Modeling network strategies ..... xiii
    - Moving records..... 9-18
    - NBLGDUMP..... 9-4
    - NBRPT01..... 9-7
    - NBRPT02..... 9-10
    - Removing records..... 9-17
    - Reporting ..... 9-4
    - Scenario generator ..... xiii
  - STATS command, NBCTRL..... 2-39
    - Collecting statistics..... 9-2
    - Examples..... 2-41
    - Parameters
      - ALL ..... 2-40
      - FILES..... 2-40
      - IMAGE ..... 2-40
      - LOCAL ..... 2-40
      - REMOTE ..... 2-40
      - RESUME ..... 2-40
      - SHOW ..... 2-40
      - START..... 2-40
      - STOP ..... 2-40
      - SWITCH ..... 2-40
    - Syntax ..... 2-39
  - STATS parameter
    - LIST (CONFIG) ..... 3-33
    - MODIFY (CONFIG) ..... 3-37
    - PRINT (CONFIG) ..... 3-40
    - SHADOW (NBDIR)..... 4-54
    - SQLIN (NBDIR)..... 4-57
    - SQLOUT (NBDIR)..... 4-59
  - STATS STOP command ..... 9-3
  - STATS SWITCH command ..... 9-3
  - STATUS parameter, SHOW (NBCTRL) ..... 2-26
  - STOP command, NBCTRL
    - Examples..... 2-44
    - Parameters
      - ALL ..... 2-44
      - CLIENT ..... 2-44
      - EXPORT [node] ..... 2-44
      - ILOG [file] ..... 2-44
      - IMPORT [node]..... 2-44
      - LOCAL [node]..... 2-43
      - NETBASE..... 2-44
      - POST [node]..... 2-44
      - REMOTE [node]..... 2-43
      - SPOOL IN [node]..... 2-44
      - SPOOL OUT [node]..... 2-44
      - SPOOL UNIX..... 2-44
      - SPOOLING..... 2-44
      - SQL..... 2-44
      - SQL IN [node] ..... 2-44
      - SQL OUT node..... 2-44
      - Syntax..... 2-43
      - STOP parameter, STATS (NBCTRL)..... 2-40
      - STOP state, SHOW STATUS (NBCTRL)..... 2-27
      - Stopping NETBASE..... 2-44
    - Subsystems, NBCTRL
      - Accessing..... 2-5
      - Prompts ..... 2-5
    - SWITCH parameter, STATS (NBCTRL)..... 2-40
    - SYNC parameter, FILE (NBDIR) ..... 4-22
    - Sync state, resync shadow file..... 5-23
    - SYNCERR=FAIL parameter, FILE (NBDIR)..... 4-23
    - SYNCERR=REMOTE parameter, FILE (NBDIR) 4-23
    - System Name parameter, CONFIG ..... 3-24
    - System operations/maintenance
      - FOS upgrades ..... 11-7
      - Full backup..... 11-5
      - HP patches..... 11-6
      - NetBase updates ..... 11-8
      - Partial backup ..... 11-4
      - Removing NetBase..... 11-9
      - Shutdown..... 11-3
      - Startup..... 11-2
- T
- TABLE command, NBDIR
    - Examples ..... 4-61
    - Parameters
      - dbname ..... 4-61
      - node..... 4-61
      - RESET ..... 4-61
      - tablename ..... 4-61
    - Syntax..... 4-61
    - tablename parameter, TABLE (NBDIR) ..... 4-61
    - TEMP parameter, DEVICE (NBDIR) ..... 4-13
    - Third party products
      - ADAGER..... 10-2
      - DBGGENERAL..... 10-4
      - MPEX..... 10-7
      - OMNIDEX ..... 10-5

## Appendix C

---

Third party shadowing	
NBTPSUTL program .....	5-14
NetBase procedures.....	5-14
TurboIMAGE databases .....	5-14
TRACE command, NBCTRL	
Examples .....	2-47
Parameters .....	2-46
Syntax.....	2-46
TRACE parameter, PROGRAM (NBDIR) .....	4-48
Tracing	
Slowing the user process .....	2-46
Tracing an application.....	13-8
transaction-priority parameter, SQLIN (NBDIR)...	4-56
Transfer Server Allocation parameter, CONFIG....	3-11
Troubleshooting	
Aborting NetBase.....	2-7, 2-10
Adding IMAGE shadow records .....	5-5
Counts, TICOUNTS.DATA.NETBASE.....	5-35
Disabling shadowing .....	2-15, 2-17
Error messages.....	0-10
MPE builds/purges/renames.....	0-9
NBSYNC, VERIFY .....	0-8
NMCMDUDC disabled.....	0-11
QUARK fails .....	13-11
Resyncing a database .....	5-25
Shadowing synchronization .....	2-7, 2-10
Shadowing TurboIMAGE databases .....	5-57
Shadowing with dynamic rollback .....	5-57
Shutting down NetBase .....	11-3
Technical support tips .....	B-2
Tracing .....	2-46
VERJCW value.....	5-20
TurboIMAGE log files, shadowing	
Map into another account/group .....	5-55
Shadow into shadowed database.....	5-53
Shadowing files.....	5-54
Type information, SHOW POST < node> (NBCTRL)2-29	
<b>U</b>	
UNDEFER command, NBCTRL	
Examples .....	2-48
Parameters .....	2-48
Syntax.....	2-48
UNLOAD command, NBDIR	
Examples .....	4-63
Parameters .....	4-63
Syntax.....	4-63
Updates	
Shadow status.....	4-36
NetBase.....	11-8
UPDPASS=password parameter, FILE (NBDIR)...	4-23
Upgrades, FOS .....	11-7
Use IMAGE Log Files parameter, CONFIG .....	3-22
Use Local Logon parameter, CONFIG.....	3-23
Use S' and J' on incoming parameter, CONFIG.....	3-12
User exits	
Export user exit.....	5-38
EXPORTEXIT.....	5-38
Post user exit.....	5-42
Post user exit COBOL example .....	5-48
Post user exit notes.....	5-47
POSTUSEREXIT .....	5-42, 5-44
POSTUSERINIT.....	5-42
user parameter, LISTF (NBDIR) .....	4-32
User Password parameter, CONFIG.....	3-23
userarea parameter	
EXPORTEXIT.....	5-40
POSTUSEREXIT .....	5-46
Utilities, NetBase	
NBCOPY .....	13-15
NBLOG.....	13-5
NBTRACE .....	13-8
NBVERS.....	13-2
QUARK .....	13-10
QUARK+ .....	13-11
<b>V</b>	
validnodes parameter, EXPORTEXIT .....	5-40
VARSTATUS parameter, SHOW (NBCTRL) .....	2-26
VERJCW values .....	5-21
Versions, NetBase .....	B-4
<b>W</b>	
WAIT state, SHOW STATUS (NBCTRL).....	2-27
wakeup-seconds parameter, SQLOUT (NBDIR)....	4-58
<b>X</b>	
XFER state, SHOW STATUS (NBCTRL).....	2-27
XL command, NBDIR	
Examples.....	4-64
Parameters.....	4-64
Syntax .....	4-64
XL lists.....	4-43
XL= parameter, PROGRAM (NBDIR).....	4-47