BASEstar Open Server for OpenVMS

Inst. & Management Guide

Order Number: AA-QS3SB-TE

April 1999

This manual provides instructions for installing and managing BASEstar Open Server for OpenVMS on VAX and Alpha platforms.

Revision/Update Information:	This is a revised document for the current release.
Operating System and Version:	OpenVMS Version 6.1, 6.2, 7.1 and 7.2
Software Version:	BASEstar Open Server Version 3.2

Compaq Computer Corporation Huston,Texas

)

First printing, July 1996
Revised, April 1997
Revised, April 1999

© Digital Equipment Corporation 1999. All Rights Reserved.

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Compaq Computer Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

COMPAQ,the Compaq logo and the Digital logo are registered in the U.S. Patent and Trademark Office.

Tru64 is a trademark of Compaq Computer Corporation.

The following are trademarks of Digital Equipment Corporation: Alpha AXP, BASEstar, DEC, DECmessageQ, DECnet, DECnet–DOS, DECosap, DEComni, DIGITAL, DIGITAL UNIX, FMS, LN03, MicroVAX, NAS, OpenVMS, OpenVMS Alpha, PATHWORKS, PDAS, Rdb/VMS, ReGIS, ThinWire, TK, ULTRIX, VAX, VAXcluster, VAX COBOL, VAX FORTRAN, VAX Pascal, VAX RMS, VMS/ULTRIX Connection, and VT.

Microsoft and Windows NT are registered trademarks of Microsoft Corporation. The following are third-party trademarks:

Other product names mentioned herein may be the trademarks of their respective companies.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface xi

Part I Installing BASEstar Open Server

1 Preparing to Install BASEstar Open Server

1.1 Pre-Installation Check List 1–1
1.2 Checking the Contents of the Distribution Kit 1–1
1.3 Required Operating System Components 1–1
1.4 Pre-requisite Hardware 1–2
1.5 Pre-requisite Software 1–2
1.6 Pre-requisite Licenses 1–2
1.7 Backing Up Your System Disk 1–3
1.8 Preparing Your System 1–3
1.8.1 Account Privileges 1–3
1.8.2 Disk Space Requirements 1–4
1.8.3 Account Quotas for Installation 1–4
1.8.4 OpenVMS SYSGEN Parameters and Minimum Values 1–5
1.9 Error Recovery 1–6

2 Installing BASEstar Open Server

2.1 The Installation Procedure 2–1
2.1.1 VMSINSTAL Checks 2–1
2.1.2 Installation Steps 2–1

3 After Installing BASEstar Open Server

3.1 Post-Installation Checklist 3–1
3.2 Running the Installation Verification Procedure Separately 3–1
3.2.1 Running the BASEstar Open Server Installation Verification Procedure 3–1
3.2.2 Running the DEComni API Installation Verification Procedure 3–2
3.3 Setting Up and Starting the BASEstar Open Environment 3–2
3.4 User Account Requirements 3–2
3.5 Starting DEComni API 3–2
3.6 Stopping DEComni API 3–3
3.7 Getting Help and Reporting Problems 3–3

4 Installing BASEstar CIMfast

Part II Installing the Device Connection Manager

5 Before Installing the BASEstar Classic Device Manager

5.1 Overview 5–1
5.2 Main Installation Options 5–1
5.3 Installation of the Device Connection Management Software 5–1
5.4 Distribution Kit 5–2
5.5 Pre-requisite Hardware 5–2
5.6 Pre-requisite and Optional Software 5–2
5.7 Pre-installation Steps 5–2

6 Installing the BASEstar Classic Device Manager

6.1 The Installation Procedure 6–1
6.2 Installing BASEstar Software 6–2
6.3 Installing BASEstar Device Connection Management Software 6–5

7 After Installing the BASEstar Classic Device Manager

7.1 The Device Connection Manager Post-installation Procedures 7–1

Part III Environment Management

8 Introduction to Environment Management

8.1 Environment Management Overview 8–1
8.2 Environment and Application Components Relationship 8–1
8.3 BASEstar Open Environment Components 8–2
8.3.1 Nodes 8–2
8.3.2 PODB Nodes 8–3
8.3.3 Realms 8–3
8.3.4 Sample Environment Configuration 8–5

9 Environment Management Procedures

9.1 Setup Procedures 9–2
9.1.1 Setting Up a Node 9–2
9.1.2 Setting Up a Realm 9–2
9.1.2.1 Setting Up a Realm on a Node 9–2
9.1.2.2 Setting Up the Realm Database on the PODB Node 9–2
9.1.3 Setting Up the Sample Environment 9–2
9.2 Startup Procedures 9–4
9.2.1 Starting Up a Node 9–4
9.2.2 Starting Up a Realm 9–4
9.2.2.1 Starting Up a Realm on a Node 9–4
9.2.2.2 Starting Up the Database Server for a Realm on the PODB Node 9–4
9.2.3 Starting Up the Sample Environment 9–5
9.3 Shutdown Procedures 9–6

9.3.1 Shutting Down a Realm 9–6
9.3.1.1 Shutting Down the Database Server for a Realm on the PODB Node 9–6
9.3.1.2 Shutting Down a Realm on a Node 9–6
9.3.2 Shutting Down a Node 9–6
9.3.3 Shutting Down the Sample Environment 9–7
9.4 Unset Procedures 9–8
9.4.1 Unsetting a Realm 9–8
9.4.1.1 Unsetting a Realm from a Node 9–8
9.4.1.2 Unsetting a Realm Database on the PODB Node 9–8
9.4.2 Unsetting a Node 9–8
9.4.3 Unsetting the Sample Environment 9–8
9.5 Monitoring BASEstar Open 9–10

10 Environment Management Command Reference

10.1 Executing Environment Commands 10–1
10.1.1 Requirements and Constraints 10–1
10.1.1.1 Users and Commands 10–1
10.1.1.2 Executing a Command 10–3
10.2 Privileges for Running Environment Management Procedures 10–3
cli_bstr 10–5
bstr_env_show 10–7
bstr_node_setup 10–11
bstr_node_shut 10–15
bstr_node_start 10–16
bstr_node_unset 10–17
bstr_realm_check_env 10–18
bstr_realm_setup_db 10–20
bstr_realm_setup_node 10–22
bstr_realm_shut_db 10–23
bstr_realm_shut_node 10–24
bstr_realm_start_db 10–26
bstr_realm_start_node 10–27
bstr_realm_unset_db 10–29
bstr_run 10–30

11 BASEstar Open Environment Monitor

11.1 Introduction 11–1
11.2 Running the BASEstar Open Environment Monitor 11–2
11.3 General Characteristics 11–2
11.4 Views 11–3
11.4.1 Active Realm View 11–3
11.4.2 Active Node View 11–5

12 BASEstar Open-Supplied Servers

12.1 Servers, Activities and Owned Resources 12–2
12.1.1 Snapshots 12–2
12.1.2 Logged Information 12–2
12.2 Global Object Services 12–3
12.3 Application Management Services 12–4
12.3.1 Logical Names - Inheritance and Usage 12–4
12.3.1.1 Application Management Services Server Privileges 12–5
12.4 Event Services 12–6
12.4.1 Using Application Management Services to Manage an Event Services Server 12–6
12.4.2 Starting Up and Shutting Down an Event Services Server from the Command Interpreter 12–6
12.5 Packet Services 12–8
12.5.1 Using Application Management Services to Manage a Packet Services Server 12–8
12.5.2 Starting Up and Shutting Down a Packet Services Server from the Command Interpreter 12–8
12.6 Data Services 12–10
12.6.1 Environmental Variables 12–10
12.6.2 Using Application Management Services to Manage a Data Services Server 12–10
12.6.3 Starting Up and Shutting Down a Data Services Server from the Command Interpreter 12–11
12.7 Device Services 12–13
12.7.1 Global Variables 12–13
12.7.1.1 Specifying the Calling VMD 12–13
12.7.1.2 Specifying the Connection Retry Timeout to a VMD 12–13
12.7.2 Using Application Management Services to Manage a Device Services Server 12–14
12.7.3 Starting Up and Shutting Down a Device Services Server from the Command Interpreter 12–14
12.8 Combined Data & Device Services 12–16
12.8.1 Environmental Variables 12–16
12.8.2 Using the Data & Device Services to Support Passive Connections . . . 12–17
12.9 PC Communication Servers 12–18

13 Application Management Services Monitor

13.1 Introduction 13–1
13.2 Running Application Management Services Monitor 13–1
13.3 General Characteristics 13–2
13.4 Views 13–3
13.4.1 Actor View 13–3
13.4.2 Activity View 13–6
13.4.3 Program View 13–7
13.4.4 Process View 13–8
13.4.5 Node View 13–9

14 Log Services Features

14.1 Saving Copies of Log Files 14–1
14.1.1 Purging Log Files 14–2
14.2 Log File Record Format 14–2
14.2.1 Displaying Log Files 14–5
14.2.1.1 Displaying a Save Copy 14–5
14.2.1.2 Displaying a Working Copy 14–5
14.3 Log Services Server Activities 14–5

15 Log Services Command Reference

DISPLAY LOG 15–2
OPEN LOG 15–4

Part IV Installation and System Management Appendices

A Files and Logical Names Installed by BASEstar Open Server

A.1 Directories Created by BASEstar Open Server A–1
A.2 Files Created by BASEstar Open Server A–1
A.3 Systemwide Logical Names Created by BASEstar Open Server A–6
A.4 Files Created by DEComni API A–6
A.4.1 DEComni API Files for Application Development A–6
A.4.2 DEComni API Files in VMI$ROOT A–7
A.4.3 DEComni API Image Files in VMI$ROOT A–8

B BASEstar Open Installation Messages

B.1 BASEstar Open Message Listing B–1

C Sample Installation Log Files for BASEstar Open Server on OpenVMS

C.1 Installation of BASEstar Open Server for OpenVMS Alpha C–1
C.2 Installation of BASEstar Open Server for OpenVMS VAX C–4

D Additional Files Created During Installation

D.1 Files Created During DCM-Modeled Device Connectivity Installation D–1
D.2 Files Created During BASEstar Open Server Utility Installation D–6
D.3 Files Created During BASEstar CIMfast Installation D–6
D.4 Systemwide Logical Names Created by the Device Connection Manager D–10

E Device Connection Manager Installation Messages

E.1 Message Format E–1
E.2 Device Connection Manager Message Listing E–1

F Sample Installation Log File of the Device Connection Manager

G BASEstar Open Global Variables

G.1 Global Variable List G–1
G.2 Installation-specific Global Variables G–4
G.3 BSTR_REALM Global Variable G–4
G.4 Global Variables for Server Activation G–4
G.5 Application Management Services Global Variables G–4

H Environment and Parameter Files

H.1 General Record Format H–1
H.2 Format of the File for the Node *environment* Attribute H–2
H.3 Format of the File for the Program *parameters* Attribute H–2
H.3.1 File Format H–3

I Environment Component Processes

J Managing Snapshot Files in a Distributed Environment

J.1 Using Configuration Management Commands in a Multi-Node Environment J–1
J.2 Snapshot File Location J–1
J.3 The BSTR_DBACCESS_KEY Global Variable J–2
J.4 Making Snapshot Files Accessible J–2
J.4.1 Installing Snapshot Directories on a Shared Disk J–2
J.4.2 Copying Snapshot Files Via DECnet J–2

Index

Examples

14–1 Node Log File Example 14–3
14–2 Realm Log File Example 14–4
H–1 Program Parameter File (*flag* not set) H–4
H–2 Program Parameter File (*flag* set to NOUAF) H–5

Figures

8–1 Sample BASEstar Open Environment 8–6
11–1 Active Realm View 11–3
11–2 Active Node View 11–5
13–1 Cursor Mode 13–3
13–2 Actor View with Actor Attributes 13–4
13–3 Actor View with Activity Attributes 13–5
13–4 Activity View 13–6
13–5 Program View 13–7
13–6 Process View 13–8
13–7 Node View 13–9

Tables

1	Conventions Used in BASEstar Open	xiii
1–1	Pre-Installation Check List	1–1
1–2	Hardware Requirements for Installing BASEstar Open Server	1–2
1–3	Disk Space Requirements for BASEstar Open Server Components	1–4
1–4	Minimum Quotas for Installing BASEstar Open Server on OpenVMS	1–4
1–5	Minimum Required System Parameter Values - OpenVMS Alpha	1–5
3–1	Post-Installation Check List	3–1
5–1	BASEstar Licenses	5–2
5–2	BASEstar Minimum System Parameter Values	5–3
7–1	Process Resource Limits	7–2
7–2	VMS Privileges Required to Start BASEstar Software	7–3
10–1	OpenVMS Relationship Between Environment Management Commands and Users	10–2
10–2	Environment Management Privileges	10–3
12–1	Logical Names - Validity and Usage	12–5
12–2	Data Management—Environmental Variables	12–10
12–3	Device Management—Environmental Variables	12–13
12–4	Data and Device Management—Environmental Variables	12–16
14–1	Log Services CLI Commands	14–1
14–2	Log File Component Identifier	14–2
A–1	BASEstar Open Server Directories	A–1
A–2	Files Installed by BASEstar Open Server	A–1
A–3	Files Installed by DEComni API	A–6
A–4	Files Installed by DEComni API	A–7
A–5	Image Files Installed by DEComni API	A–9
D–1	DCM-Modeled Device Connectivity Installation Files	D–1
D–2	BASEstar Open Server Utility Installation Files	D–6
D–3	BASEstar CIMfast Directories	D–6
D–4	BASEstar CIMfast Installation Files	D–7
D–5	BASEstar CIMfast LSE Files	D–9
D–6	Logical Names Created by the Device Connection Manager Installation Procedures	D–10
G–1	BASEstar Open Global Variables	G–1
H–1	Valid Values of the File for the Node *environment* Attribute	H–2
H–2	Valid Values of the File for the Program *parameters* Attribute	H–3
I–1	Processes	I–2

Preface

Intended Audience

This manual is addressed to system managers responsible for installing and setting up BASEstar Open software. You must be familiar with the OpenVMS installation procedure (VMSINSTAL).

Structure of this Document

This manual is organized as follows:

- Chapter 1 describes the operating system and hardware requirements for installation and related procedures that you must complete before installing BASEstar Open Server.
- Chapter 2 describes the BASEstar Open Server installation procedure.
- Chapter 3 describes the Installation Verification Procedure (IVP) and any post-installation procedures that you must complete in order to use BASEstar Open Server.
- Chapter 4 describes the BASEstar CIMfast installation procedure.
- Chapter 5 describes the pre-requisites for installing the BASEstar Classic Device Connection Manager.
- Chapter 6 describes the BASEstar Classic Device Connection Manager installation procedure.
- Chapter 7 describes the post-installation tasks for the BASEstar Classic Device Connection Manager.
- Chapter 8 provides an overview of environment management issues.
- Chapter 9 describes the procedures that allow you to set up, start up and shut down the components of your BASEstar Open environment.
- Chapter 10 contains reference information for the above procedures (e.g., pre-reqisites, privileges, etc.).
- Chapter 11 describes the BASEstar Open Environment Monitor, an interactive tool that enables you to monitor the BASEstar Open environment at run-time.
- Chapter 12 explains how to start up, shut down and manage BASEstar Open-supplied servers.
- Chapter 13 describes the Application Management Services Monitor, an interactive tool that enables you to monitor Application Management Services objects.
- Chapter 14 describes how Node-specific log services collect information from the environment components.

- Chapter 15 contains reference information for the log services commands.
- Appendix A describes the files and directories created on your system by the installation procedure.
- Appendix B provides a comprehensive list of the BASEstar Open Server installation messages.
- Appendix C contains sample listings for BASEstar Open Server installation on OpenVMS VAX and Alpha platforms.
- Appendix D lists the additional files created during the installation of the BASEstar Open Server software (including files pertaining to device connection management, library management and utility software).
- Appendix E explains the format used for the Device Connection Manager installation messages. It also provides a comprehensive list of the error messages associated with the installation procedure, together with the recommended recovery procedure for each error.
- Appendix F contains a sample installation log for installation of the Device Connection Manager on OpenVMS VAX and Alpha platforms.
- Appendix G lists and describes the BASEstar Open Server global variables.
- Appendix H describes the format and contents of:
  - The environment file, as specified by the *environment* attribute of a Node object
  - The parameter file, as specified by the *parameters* attribute of a Program object.
- Appendix I lists the platform-dependent processes that implement each BASEstar Open environment component.
- Appendix J provides information about managing snapshot files in a multinode environment.

BASEstar Open Documentation Set

This manual describes a software component or aspects of the BASEstar Open family of products. The complete set of documents relating to the BASEstar Open family is as follows:

- BASEstar Open:
  - *BASEstar Open Introduction*
  - *BASEstar Open Reference Guide*
  - *BASEstar Open Command Language Interface*
  - *BASEstar Open Application Programming Interface*
  - *BASEstar Open Messages*
  - Platform-specific installation and management guides
  - *BASEstar Open Guide to DCM-Modeled Device Connectivity* (for supported platforms)
- BASEstar CIMfast (for supported platforms):
  - *BASEstar CIMfast User's Guide*

- *BASEstar CIMfast Programmer's Reference Guide*
- *BASEstar CIMfast Guide to DECmessageQ Support* (not available on Windows NT platforms)
- *BASEstar CIMfast Guide to SQL Support*

- DEComni API:
  - *DEComni API and MMS User Guide*
  - *DEComni API Guide to Using Omni Directory Services* (for supported platforms)
  - *DEComni API Guide to Using OmniView* (for supported platforms)
  - *DEComni API Omni Definition Facility User Guide* (for supported platforms)
- Device Access Software documentation
  - Specific manuals for each supported device

Conventions

Table 1 lists the conventions used in the BASEstar Open documentation set.

Table 1 Conventions Used in BASEstar Open

%	The default user prompt is your system name followed by a right angle bracket (>). In the BASEstar Open docset, a percent sign (%) is used to represent this prompt.
Ctrl/x	Indicates that you must hold down the Ctrl key while you press another key.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
. . .	A vertical ellipsis indicates the omission of items from a code example or sample command; the items are omitted because they are not important to the topic being discussed.
. . .	A horizontal ellipsis in format descriptions or in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
[ ]	In format descriptions, brackets indicate optional elements; you can select none, one, several, or all of the choices. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification, or in the syntax of a substring specification in an assignment statement.)
( )	In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.

(continued on next page)

Table 1 (Cont.) Conventions Used in BASEstar Open

boldface text	Boldface text represents one of the following cases: • user input • the introduction of a new term • the status values **true** or **false**
italic type	Indicates titles of manuals, variables, arguments, data structures, fields, callable functions, operands, and utilities. Italic text also represents information that can vary in system messages (for example, Internal error *number*), command lines (for example, /PRODUCER=*name*), and command operands in text.
`code type`	Indicates information that is part of the code for a program or application.
numbers	Unless otherwise noted, all numbers in the text are assumed to be decimal. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.
UPPERCASE TEXT	Uppercase text indicates a command, the name of a file, the name of a file protection code, the abbreviation for a system privilege, the name of a field, or the value of an attribute where attributes are chosen from a list.
"point_a + point_b"	Literal string. Using quotation marks is optional unless there is a space, slash (/), or parenthesis in the string.
BASEstar Open names	Valid characters for the names of BASEstar Open objects are the alphanumeric characters (A–Z) and (0–9), the underscore (_), the dollar sign ($), and the following multinational characters: ÅÀÁÂÃÄÇÈÉÊËÌÍÎÏÒÓÔÕÖÙÚÛÜÆÑ@Øß.

Part I

Installing BASEstar Open Server

The section consists of three chapters that describe the following:

- Pre-installation tasks — the tasks and checks you must perform before beginning installation of BASEstar Open Server
- The installation procedure
- Post-installation tasks — the tasks you must perform immediately after installing BASEstar Open Server.

1

Preparing to Install BASEstar Open Server

1.1 Pre-Installation Check List

Table 1–1 summarizes the tasks you must perform before installing BASEstar Open Server. Make sure that you have performed all these tasks before proceeding with the installation.

Table 1–1 Pre-Installation Check List

Task	For More Information
Check that you have the complete kit	See Section 1.2
Check that you have the pre-requisite hardware, software, and license	See Section 1.4, Section 1.5, and Section 1.6
Back up your system disk	See Section 1.7
Check that you have the required privileges	See Section 1.8.1
Check that you have enough disk space	See Section 1.8.2
Check account quotas	See Section 1.8.3
Check system parameters	See Section 1.8.4

1.2 Checking the Contents of the Distribution Kit

Your bill of materials (BOM) and indented bills report (BIL) specify the number and contents of your media. Be sure to verify the contents of your kit with this information. If your kit is damaged or if you find that parts of it are missing, contact your COMPAQ representative.

COMPAQ provides a BASEstar Open Server cover letter with the kit, and online release notes in the following files:

- SYS$HELP:BSTROSVA032.RELEASE_NOTES (OpenVMS Alpha)
- SYS$HELP:BSTROSVV032.RELEASE_NOTES (OpenVMS VAX)

COMPAQ strongly recommends that you read both the cover letter and the release notes before proceeding with the installation.

1.3 Required Operating System Components

Both the OpenVMS Alpha and the OpenVMS VAX operating systems come with a variety of support options, or classes. Classes include such features as networking and RMS Journaling.

For a complete list of the required classes, refer to the BASEstar Open Server *Software Product Description* (SPD).

1.4 Pre-requisite Hardware

Table 1–2 describes the hardware items needed to install BASEstar Open Server.

Table 1–2 Hardware Requirements for Installing BASEstar Open Server

Item	Purpose
RRD40 CD Reader[1] RRD40 CD Reader, nine-track magnetic tape drive, TK50 or TK70 as appropriate for distribution kit media[2]	To install from media
ANSI-compatible video display terminal such as the VT300 or VT400 series	To communicate with the Operating System and respond to prompts.
Any VAXstation[2] or terminal that supports the X Window System	For graphic screen development.

[1]For OpenVMS Alpha only.

[2]For OpenVMS VAX only.

1.5 Pre-requisite Software

You can install BASEstar Open Server Version 3.2 on:

- OpenVMS Version 6.1, 6.2, 7.1 and 7.2

The pre-requisite software also includes the following layered products:

- DEC TCP/IP Services for OpenVMS
- DEC Rdb for OpenVMS (optional)
- DECnet OSI (for certain DASes)
- OSAK (for MMS DASes only)

For further details of these products, including the currently supported version numbers, refer to the BASEstar Open Server *Software Product Description* (SPD 47.87.xx).

The following command displays the OpenVMS operating system version that you are running:

```
$ WRITE SYS$OUTPUT F$GETSYI("VERSION")
```

1.6 Pre-requisite Licenses

Before you install and run BASEstar Open Server Version 3.2 on a newly-licensed node or cluster, you must first register a License Product Authorization Key (License PAK) using the License Management Facility (LMF). The License PAK may be shipped along with the kit if you ordered the license and media together; otherwise, it is shipped separately to a location based on your license order.

If you are installing BASEstar Open Server as an update on a node or cluster already licensed for this software, you have already completed the License PAK registration requirements. Refer to the *OpenVMS License Management Utility Manual* for further information about the License Management Utility.

VMSINSTAL checks which license has been installed and installs the required components of the BASEstar Open Server for OpenVMS (VAX or Alpha) kit. **If no license is installed, VMSINSTAL only installs the components for the runtime option of BASEstar Open Server**, and asks if you wish to continue with the installation. If you respond affirmatively, the IVP is not executed and BASEstar Open Server will not run. However, if a valid license is installed later, you can still run BASEstar Open Server.

If you are installing pre-requisite or optional software along with BASEstar Open Server, review the PAK status and install the PAKs for any pre-requisite or optional software before you install BASEstar Open Server.

Refer to the BASEstar Open Server *Software Product Description* for further details of the licenses available for BASEstar Open Server.

1.7 Backing Up Your System Disk

At the beginning of the installation, the procedure asks if you have backed up your system disk. COMPAQ recommends that you do a system disk backup before installing any software. If a system failure occurs at a critical point during the installation, some of the files could be corrupted. For further information about backing up your system disk, refer to the section on the Backup Utility in the *OpenVMS System Manager's Manual*.

Also ensure that the VMS Help library is not used during the period of the BASEstar Open Server installation.

1.8 Preparing Your System

This section discusses the disk space and the privileges and parameter values that are required for installing BASEstar Open Server for OpenVMS (Alpha and VAX).

VMSINSTAL asks if you wish to install DEComni; answer **no** if you do not wish to do so. Account privileges, quotas, and system parameters vary according to whether or not VMSINSTAL performs the DEComni installation.

If you are running a multiversion of DEC Rdb, you must choose the version to be used before installing BASEstar Open Server, by running the following command line:

```
$ @SYS$LIBRARY:RDBVMS_SETVER
version_number
```

VMSINSTAL checks for the required values and reports any errors before deciding if to fail.

1.8.1 Account Privileges

To install BASEstar Open Server, you must be logged in to the system management account SYSTEM, and have ALL privileges set.

If you are not logged into the SYSTEM account and VMSINSTAL will be installing or reinstalling DEComni, the BASEstar Open Server installation procedure warns you that DEComni IVP will not be able to be run.

Note that the installation procedure, VMSINSTAL, turns off BYPASS privilege at the start of the installation.

1.8.2 Disk Space Requirements

To install BASEstar Open Server, you require at least 110,000 blocks (Alpha systems) or 83,000 blocks (VAX systems) of free disk space (peak block count) on your system disk. Table 1–3 lists the requirements of each individual component.

Table 1–3 Disk Space Requirements for BASEstar Open Server Components

Component	Blocks Required (Alpha)	Blocks Required (VAX)
BASEstar Open Server	31,000	30,000
DEComni API	66,000	45,000
BASEstar CIMfast	13,000	8,000
Total:	110,000	83,000

After BASEstar Open Server is installed, the BASEstar Open environment requires 29,000 blocks of free disk space. This value does not take into account disk space required by the Permanent Object Database.

To determine the number of free disk blocks, enter the following command at the DCL prompt:

```
# SHOW DEVICE device_name
```

where *device_name* is the device on which you want to install BASEstar Open Server for OpenVMS.

1.8.3 Account Quotas for Installation

The account from which BASEstar Open Server is installed must have certain minimum quotas. Before attempting to install BASEstar Open Server, verify that the account has the correct quotas.

Table 1–4 summarizes the minimum quotas required (both with and without the DEComni installation stage) for the installation of BASEstar Open Server on OpenVMS Alpha and OpenVMS VAX.

Table 1–4 Minimum Quotas for Installing BASEstar Open Server on OpenVMS

	Minimum Values	
User Account Parameter	Without DEComni API	With DEComni API
ASTLM	20	24
BIOLM	40	40
BYTLM	80000	80000
DIOLM	20	20
ENQLM	20	2000
FILLM	20	50
JTQUOTA	1024	1024

(continued on next page)

Table 1–4 (Cont.) Minimum Quotas for Installing BASEstar Open Server on OpenVMS

User Account Parameter	Minimum Values	
	Without DEComni API	With DEComni API
PGFLQUOTA	10,000	30,000
PRCLM	4	4
TQELM	6	6
WSDEFAULT	512	512
WSQUOTA	1024	1024
WSEXTENT	2048	3072

Use the Authorize utility to display the process quotas, as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW SYSTEM
```

If no changes are necessary, enter the EXIT command to leave the Authorize utility.

To modify system account quotas (for example, BYTLM), use the following command line:

```
UAF> MODIFY SYSTEM/BYTLM=32768
```

The utility responds with a message confirming the change. You can then enter the EXIT command to exit from the utility. You must log out and then log in again to reset the system account quotas.

The Authorize utility is described in the *OpenVMS System Manager's Manual*.

1.8.4 OpenVMS SYSGEN Parameters and Minimum Values

For the BASEstar Open Server installation to complete successfully some System Generation Utility (SYSGEN) parameters must be set to certain values before the installation. Table 1–5 and lists the minimum required values for the system parameters.

To install and run BASEstar Open Server, you must have sufficient free global pages and global sections. You must first determine how many free global pages and sections you have on your system. Then use AUTOGEN if you need to increase the values of the GBLPAGES and GBLSECTIONS system parameters.

Table 1–5 Minimum Required System Parameter Values - OpenVMS Alpha

System Parameter	Minimum Values[1]	
	Without DEComni	With DEComni
Available GBLPAGES	7000 (Alpha) 3100 (VAX)	7995 (Alpha) 4095 (VAX)
Available GBLSECTIONS	50	65

[1]The values listed for these system parameters represent the minimum number of free global pages and global sections required to run BASEstar Open Server in addition to the requirements of your current system and other software. These values are not the total number you need to run your system and other software.

Checking System Parameter Values

To check the values of your system parameters, use either of the following methods:

- For the number of remaining free pages and free sections

  Use the DCL lexical F$GETSYI with a system parameter *item* (for example, FREE_GBLPAGES and FREE_GBLSECTS) as follows:

  ```
  $ WRITE SYS$OUTPUT F$GETSYI("FREE_GBLPAGES")
  5680
  $ WRITE SYS$OUTPUT F$GETSYI("FREE_GBLSECTS")
  128
  ```

  If the values for the free global pages or global sections displayed by the system are greater than the values in Table 1–5, you do not need to increase them. If the values are less, you must increase the system parameter setting.

- For the static number of free pages and free sections

  Invoke the System Generation Utility (SYSGEN):

  ```
  #
  RUN SYS$SYSTEM:SYSGEN
  SYSGEN>
  ```

  At the SYSGEN> prompt, enter the SHOW command to display the value of a system parameter. The values displayed should equal or exceed the value of each system parameter listed in Table 1–5. The following example displays the value for the GBLPAGES system parameter:

  ```
  SYSGEN>
  SHOW GBLPAGES
  ```

  After checking the parameters with the SHOW command, enter the EXIT command at the SYSGEN> prompt to return to DCL level.

1.9 Error Recovery

If errors occur during the installation, VMSINSTAL displays failure messages. If the installation itself fails, you see the following message:

```
%VMSINSTAL-E-INSFAIL, The installation of BSTROSVA032 V3.2
has failed.
```

Errors can occur during the installation if any of the following conditions exist:

- The operating system version is incorrect
- A pre-requisite software version is incorrect
- Quotas necessary for successful installation are insufficient[1]
- System parameter values for successful installation are insufficient
- The OpenVMS Help library is currently in use.

For descriptions of the error messages generated by these conditions, refer to the OpenVMS documentation on system messages, recovery procedures, and OpenVMS software installation. If you receive notification that one of these conditions exist, take the appropriate action as described in the message.

[1] If GBLPAGES and GBLSECTIONS are insufficient, the installation continues but displays warning messages that BASEstar Open Server may fail during the IVP or at run-time.

Preparing to Install BASEstar Open Server 1.9 Error Recovery

If an error occurs while using BASEstar Open Server and you believe the error is caused by a problem with the product, take the appropriate action as described in Section 3.7.

2
Installing BASEstar Open Server

2.1 The Installation Procedure

Before you begin installing BASEstar Open Server, make sure that your system meets the requirements outlined in Chapter 1 and that you have completed all the tasks listed in Table 1–1.

The BASEstar Open Server installation procedure consists of a series of questions and informational messages. To abort the installation procedure at any time, press Ctrl/Y. When you press Ctrl/Y, the installation procedure deletes all files it has created up to that point and exits. You can then start the installation again.

The approximate time required for the full installation of BASEstar Open Server is as follows:

- 20 to 40 minutes, for OpenVMS Alpha
- 35 to 50 minutes, for OpenVMS VAX

However, the exact time depends largely on the type of media you are using and your system configuration.

2.1.1 VMSINSTAL Checks

When you invoke VMSINSTAL, it checks whether:

- You have set your default device and directory to SYS$UPDATE
- You are logged in to the system management account SYSTEM
- DECnet is running
- Any users are logged in to the system

If VMSINSTAL detects any problems during the installation, it notifies you and asks if you want to continue the installation. In some instances, you can enter YES to continue. To stop the installation process and correct the situation, enter NO or press Return. Then correct the problem and restart the installation.

2.1.2 Installation Steps

The following steps describe how to install BASEstar Open Server on an OpenVMS Alpha system:

1. **Start the installation**

   To start the installation, invoke the SYS$UPDATE:VMSINSTAL.COM procedure from a privileged account (for example, SYSTEM). Use the kit name **BSTROSVA032** as the first parameter to the command VMSINSTAL. Use the device or directory on which the installation kit is located as the second parameter.

For example, if the media containing the BASEstar Open Server software is on device mua0, then you must issue the following command to install BASEstar Open Server (on an OpenVMS Alpha system):

```
# @SYS$UPDATE:VMSINSTAL BSTROSVA032
SYS$MANAGER

OpenVMS AXP Software Product Installation Procedure V7.1

It is dd-mmm-yyyy at hh:mm.

Enter a question mark (?) at any time for help.
```

If you do not supply either the product name or the device name, VMSINSTAL prompts you for this information later in the installation procedure.

2. **Confirm system backup**

```
* Are you satisfied with the backup of your system disk [YES]?
```

VMSINSTAL asks if you are satisfied with your system backup. You should always back up your system disk before performing an installation. If you are satisfied with the backup of your system disk, press Return. Otherwise, enter NO to discontinue the installation. After you back up your system disk, you can restart the installation.

3. **Specify the drive on which you will mount the distribution volumes**

```
* Where will the distribution volumes be mounted:
```

Enter the appropriate device/directory name.

4. **Enter the name(s) of the product(s) to be installed**

```
Enter the products to be processed from the first distribution volume set.
* Products: bstrosva032

* Enter installation options you wish to use (none):
```

Specify any installation options to be used.

```
The following products will be processed:

  BSTROSVA V3.2
```

5. **Begin the installation**

At this point, VMSINSTAL checks which license has been installed and continues with the installation as follows:

```
Beginning installation of BSTROSVA V3.2 at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
******************************************************************************
Copyright Digital Equipment Corporation. 1999. All Rights Reserved.

   Restricted Rights: Use, duplication, or disclosure by the U.S.
   Government is subject to restrictions as set forth in subparagraph
   (c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
   52.227-14 Alt. III, as applicable.

   This software is proprietary to and embodies the confidential
   technology of Digital Equipment Corporation. Possession, use, or
   copying of this software and media is authorized only pursuant to a
   valid written license from Digital or an authorized sublicensor.

                - BASEstar Open V3.2-0 Installation Procedure -
```

```
   This procedure decides from the license which component is to be
   installed. If you do not have any license, the procedure allows you
   to install the run-time option, but does not execute the IVP.
```

For more information on the license requirement for the installation of BASEstar Open Server, refer to Section 1.6.

6. **Check whether DEComni is already installed**

   VMSINSTAL checks whether DEComni API is already installed. It then informs you that DEComni API will be installed automatically if it is not already installed or if an incorrect version is installed. If the correct version of DEComni API has already been installed, VMSINSTAL asks if you wish to reinstall it, as follows:

```
This procedure will also give you the option of installing the Device
Connection component (DEComni). If you wish to use the device access capability
of this version of BASEstar Open, you must install the Device Connection
component included in this kit.

******************************************************************************

BASEstar Open Development option will be installed

* Do you want to install the Device Connection component (DEComni) [Y]?)
```

7. **Specify the target disk**

   VMSINSTAL asks you where you would like to have BASEstar Open Server installed. If your system is a VAXcluster and you would like to have BASEstar Open Server available on all or a set of nodes in the cluster you must install BASEstar Open Server on a disk that is visible to those nodes in the cluster.

```
BASEstar Open can be installed on any disk of your system.
    There is no default disk: you must enter a disk name.
    The following disks are available on the system.

$1$DKA0:      (VONALB)  Mounted             0  PCTS            126180     2   1
$1$DKB0:      (VONALB)  Mounted             0  BASESTAR        355743     7   1
$1$DKB400:    (VONALB)  Mounted             0  ALPHASYS        147556   417   1

* Target disk: $1$DKA0:

* BSTROSVA V3.2 root directory [[BSTROSVA032]]:

    You are going to install BSTROSVA V3.2 on

    Target disk:    $1$DKA0:
    Root directory: [BSTROSVA032]

* Is this correct [Y]?
      ********************************************************
```

   For more information on managing BASEstar Open Server, refer to Part III, Environment Management.

8. **Specify the DEComni API/DAS product(s) that you wish to install**

```
Select which products you want to install:

      1  DEComni API
      2  DASTkRUN
      3  DASTk 232
      4  ALL of the above

      ********************************************************

* Enter number(s): [1]:4
```

```
        ********************************************************
        DEComni API V3.1 is installed on this system.
        It is recommended that the database files be renamed or
        deleted if you do not wish to use the existing database.
        If you choose not to rename or delete the database,the
        existing database will continue to be used.
        ********************************************************

                1. Rename existing database
                2. Continue to use existing database

* Enter a number [2]:

 A directory for Omni Directory Services will be created. You
 need to provide the disk name in which the directory will be
 located. If you do not specify a location, the directory will
 be placed in SYS$SYSDEVICE:[OMNI...]. Include the colon after the
 disk name.

* Enter the disk now [ddcu:] :
* Do you want to purge files replaced by this installation [YES]?
```

9. **Run the BASEstar Open Server and DEComni API IVPs**

   The installation procedure now asks if you wish to run the BASEstar Open Server Installation Verification Procedure (IVP) (which calls the DEComni IVP) after the installation is complete, as follows:

   ```
   * Do you want to run the IVP after the installation [YES]?
   ```

   For more information on the IVP, refer to Section 3.2.

   There are no further questions concerning the installation of BASEstar Open Server.

10. **Read informational messages**

    At this point, the installation procedure displays a number of informational messages that report on the progress of the installation.

    For further information about how to run the BASEstar Open Server IVP separately, refer to Section 3.2.1.

For full example logs of BASEstar Open Server installations on both OpenVMS Alpha and VAX systems, refer to Appendix C.

3

After Installing BASEstar Open Server

3.1 Post-Installation Checklist

Table 3–1 summarizes the tasks you should perform after installing BASEstar Open Server. Make sure that you perform these tasks before using BASEstar Open Server.

Table 3–1 Post-Installation Check List

Task	Notes
Run the Installation Verification Procedure (IVP)	See Section 3.2
Set up and start the BASEstar Open environment	See Section 3.3

3.2 Running the Installation Verification Procedure Separately

After BASEstar Open Server is installed, COMPAQ recommends that you run the Installation Verification Procedure (IVP) to verify that the software is available on your system. You might also want to run the IVP after a system failure to make sure that users can access BASEstar Open Server.

The approximate duration of the IVP is as follows:

- 15 minutes (OpenVMS Alpha)
- 20 minutes (OpenVMS VAX)

3.2.1 Running the BASEstar Open Server Installation Verification Procedure

To run the BASEstar Open Server IVP, you must:

- Choose the version of DEC Rdb to be used before installing BASEstar Open Server (only necessary if you are running a multiversion of DEC Rdb):

  ```
  $ SYS$LIBRARY:RDBVMS_SETVER version_number
  ```

- Execute the following command from an account that has all privileges set:

  ```
  # @BSTR$KIT:[ETC]BSTR_IVP
  ```

You are not asked any questions during the IVP. If the IVP has terminated successfully, a message similar to the following is displayed:

```
BASEstar Open

 Version 3.2 IVP completed
successfully
```

Refer to Appendix C for a complete log of the IVP phase.

3.2.2 Running the DEComni API Installation Verification Procedure

To run the DEComni API IVP, follow these steps:

1. Ensure that DEComni API is running
2. Enable the following privileges:

   NETMBX
   PRMMBX
   SYSLCK
   WORLD
   GROUP

3. Execute the following command:

```
# @SYS$TEST:OMNI_IVP
```

3.3 Setting Up and Starting the BASEstar Open Environment

After installing BASEstar Open Server, you must perform procedures which set up and start up Nodes and Realms. Refer to Chapter 9 for a description of these procedures.

3.4 User Account Requirements

To work with BASEstar Open Server, user accounts on your system must have certain privileges and quotas. Refer to Section 1.8.3 for a description of how to verify and change account quotas.

3.5 Starting DEComni API

To start DEComni API, follow these steps:

1. Enable the following privileges:

   DETACH
   PRMMBX
   SYSNAM
   CMKRNL
   SYSGBL
   PRMGBL
   WORLD
   GROUP
   SYSLCK

2. Enter the following command line:

```
# @SYS$STARTUP:OMNI_STARTUP
```

   You can add this startup command to your system startup file so that DEComni API starts each time you load the system.

Note

You must not start DEComni API simultaneously started on nodes in a VAXcluster configuration because the startup procedures will fail while checking the ODS schema files. You must execute the DEComni API startup one node at a time.

3.6 Stopping DEComni API

To stop DEComni API, follow these steps:

1. Enable the following privileges:

   DETACH
   PRMMBX
   SYSNAM
   CMKRNL
   SYSGBL
   PRMGBL
   WORLD
   GROUP

2. Enter the following command line:

   ```
   # @SYS$MANAGER:OMNI_STOP
   ```

   If you want to stop DEComni API each time you shut down the system, include the previous command line in the site-specific shutdown file SYS$MANAGER:SYSHUTDOWN.COM.

3.7 Getting Help and Reporting Problems

If an error occurs while you are using BASEstar Open Server, and you believe the error is the result of a problem associated with the product, take one of the following actions:

- If you have a basic or COMPAQ support Software Agreement, call your Customer Support Center (CSC). The CSC provides telephone support for high-level advisory and remedial assistance.
- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).
- If you purchased BASEstar Open Server within the last 90 days and you think the problem is caused by a software error, you can submit an SPR.

If you submit an SPR, please take the following steps:

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include the description and version number of BASEstar Open Server that you were using. Demonstrate the problem with specific examples.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, include files, relevant data files, and so forth.
4. Provide a listing of the program.
5. If the program is longer than 50 lines, submit a copy of it on machine-readable media (floppy diskette or magnetic tape). If necessary, also submit a copy of the program library used to build the application.
6. Report only one problem per SPR. This will facilitate a faster response.
7. Mail the SPR package to COMPAQ.

After Installing BASEstar Open Server
3.7 Getting Help and Reporting Problems

Experience shows that many SPRs do not contain enough information to duplicate or identify the problem. Concise, complete information helps COMPAQ give accurate and timely service to software problems.

If you find an error in the BASEstar Open Server documentation, send an electronic mail message to *manufacturing@compaq.com*.

4
Installing BASEstar CIMfast

This section describes how to install BASEstar CIMfast, an application development environment for creating real-time event-driven applications, on an OpenVMS system. Before attempting to install BASEstar CIMfast, make sure that you have:

- Installed BASEstar Open Server, as described in Section 2.1[1]
- Set up your BASEstar Open Node, using the *bstr_node_setup* function
- Run RDBVMS_SETVER (only necessary if you are running a multiversion of DEC Rdb).

Although the BASEstar CIMfast software is a separate kit, it resides on the same CDROM as the BASEstar Open Server software, in the directory assigned to the BASEstar Open Server kits. The BASEstar CIMfast kit is called BCFAXP022. It consists of a base module, together with a DEC Rdb-based SQL bridge module (minimum DEC Rdb version 6.0).

The following steps describe how to install BASEstar CIMfast on an OpenVMS Alpha system:

1. **Start the installation**

   To start the installation, invoke the SYS$UPDATE:VMSINSTAL.COM procedure from a privileged account (for example, SYSTEM):

```
# @SYS$UPDATE:VMSINSTAL

   OpenVMS AXP Software Product Installation Procedure V7.1

It is dd-mmm-yyyy at hh:mm.

Enter a question mark (?) at any time for help.
```

   If you do not supply either the product name or the device name, VMSINSTAL prompts you for this information later in the installation procedure.

2. **Confirm system backup**

```
%VMSINSTAL-W-ACTIVE, The following processes are still active:
       BATCH_288
       BATCH_289
* Do you want to continue anyway [NO]? y
* Are you satisfied with the backup of your system disk [YES]?
```

3. **Specify the drive on which you will mount the distribution volumes**

```
* Where will the distribution volumes be mounted:  sys$manager
```

[1] It is also possible to use BASEstar CIMfast with BASEstar Classic.

4. **Enter the name of the BASEstar CIMfast kit that you wish to install**

```
Enter the products to be processed from the first distribution volume set.
* Products: BCFAXP022

* Enter installation options you wish to use (none):
```

Specify any installation options to be used.

```
The following products will be processed:

  BCFAXP V2.2
```

5. **Begin the installation**

At this point, VMSINSTAL checks which license has been installed and continues with the installation as follows:

```
Beginning installation of BCFAXP V2.2 at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.

    BASEstar CIMfast for OpenVMS installation procedure.
    Copyright (c) 1999 Digital Equipment Corporation.  All Rights Reserved.

    Enter ? at any time for help.
```

For more information about the license requirement for the installation of BASEstar CIMfast, refer to Section 1.6.

6. **Specify the target disk/directory**

VMSINSTAL asks you where you wish to install BASEstar CIMfast (by default, it performs the installation in [SYS$SYSDEVICE:[BCF]) It also requests you to provide other information about how you want to perform the installation.

```
* In what directory do you wish to install CIMfast [SYS$SYSDEVICE:[BCF]]:
* Do you want to install CIMfast SQL Support for DEC RDB [YES]?
* Do you want to purge files replaced by this installation [YES]?
* Do you want to run the IVP after the installation [YES]?

 All questions regarding the installation of CIMfast have been
 asked. Depending upon system load and configuration the
 installation will take between 5 and 15 minutes.


 This installation provides a system startup file
     SYS$STARTUP:BCF$STARTUP.COM.  You must edit your
     system startup procedure(s) to invoke this file
 on each node where CIMfast will be run.


 The CIMfast per-user setup command procedure is in
 BCF$HOME:BCF$SETUP.COM. It should be executed at login
 time by each CIMfast user, by including a line such as
 the following in each user's LOGIN.COM procedure:

     $ @BCF$HOME:BCF$SETUP

 Alternatively, the above line may added to the system-wide
 login command procedure (usually SYS$MANAGER:SYLOGIN.COM).
```

```
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.EXE].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.LIB].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.BLD].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.BLD.BCF].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.BLD.BSTR].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.BLD.RDB].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.DEMO.POINTS].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.DEMO.CBCF_POINTS].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.DEMO.CALLABLE_BCF].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.DEMO.EXPRESSIONS].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.DEMO.MACROS].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  $1$DKB400:[BCF.DEMO.SQL].

 If this product is being installed in a OpenVMS cluster,
 several actions must be taken to enable other nodes in the
 cluster to be able to use this product.  These actions
 include:

  1 Define CIMfast related logical names
  2) Update DCL tables
  3) Load a valid license

 This can be done with the following commands:

  $ RUN SYS$SYSTEM:SYSMAN
  SYSMAN> SET ENVIRONMENT/CLUSTER
  SYSMAN> DO @SYS$STARTUP:BCF$STARTUP.COM
  SYSMAN> DO INSTALL REPLACE SYS$LIBRARY:DCLTABLES.EXE
  SYSMAN> DO LICENSE LOAD BASESTAR ! or any valid license
  SYSMAN> EXIT

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

          ************************************************************

               Remember to define the logical name BSTR_REALM
              to specify the BASEstar realm you want to use.

                              For instance,

                         DEFINE  BSTR_REALM  MY_REALM

          ************************************************************


Beginning CIMfast BASEstar Open Support IVP
-------------------------------------------

Logical name BSTR_REALM temporarily defined as IVP


BASEstar Open Server V3.2

Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.

Restricted Rights: Use, duplication, or disclosure by the U.S.
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.

This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.



BASEstar Open Server V3.2

Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.
```

```
Restricted Rights: Use, duplication, or disclosure by the U.S.
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.

This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.


BASEstar Open Server V3.2

Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.

Restricted Rights: Use, duplication, or disclosure by the U.S.
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.

This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.


The node VONALB has already been set up
Executing BSTR_NODE_START ...

BASEstar Open Server V3.2

Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.

Restricted Rights: Use, duplication, or disclosure by the U.S.
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.

This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.
```

```
Checking the environment
Calling BSTR$ROOT:[ETC.HOST_STARTUP]CHECK_COM.COM;1 ...
Cleanup
Calling BSTR$ROOT:[ETC.HOST_STARTUP]CLEANUP_COM.COM;1 ...
Cleaning the communication directory.
Cleaning the communication name table.
4 phases will be executed
Startup phase 0
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START00_LOG.COM;2 ...
Starting up the LOG Server.
%RUN-S-PROC_ID, identification of created process is 000004C2
Synchronizing after phase 0
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC00_LOG.COM;2 ...
Waiting for the LOG Server to start...
Startup phase 1
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START01_COM.COM;1 ...
Creating the node table.
Starting the Watchdog.
%RUN-S-PROC_ID, identification of created process is 000004C3
Synchronizing after phase 1
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC01_COM.COM;1 ...
Waiting for the Watchdog to start...
Startup phase 2
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START02_COM.COM;1 ...
Starting the LNS.
Starting LNS for name space BSTR
%RUN-S-PROC_ID, identification of created process is 000004C4
Synchronizing after phase 2
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC02_COM.COM;1 ...
Waiting for the LNS to start...
Startup phase 3
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START03_COM.COM;1 ...
Starting the LNS monitor for VMS
Starting LNS for name space BSTR
%RUN-S-PROC_ID, identification of created process is 000004C6
Synchronizing after phase 3
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC03_COM.COM;1 ...
Waiting for the LNS monitor for VMS.
All phases completed
Executing BSTR_REALM_SETUP_NODE ...
Executing BSTR_REALM_START_NODE ...
Configuring AMS objects ...
Executing IVP CEL file ...
Executing BSTR_REALM_SHUT_NODE ...
Executing BSTR_REALM_UNSET_NODE ...
Executing BSTR_NODE_SHUT ...
Environment correctly restored

IVP TERMINATED SUCCESSFULLY
 Installation of BCFAXP V2.2 completed at 10:32

    Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

    Creating installation data file: VMI$ROOT:[SYSUPD]BCFAXP022.VMI_DATA

Enter the products to be processed from the next distribution volume set.
* Products:
 VMSINSTAL procedure done at 10:33

$)
```

Part II

Installing the Device Connection Manager

This part consists of three chapters that explain how to install the Device Connection Manager:

- Before Installing the BASEstar Classic Device Manager
- Installing the BASEstar Classic Device Manager
- After Installing the BASEstar Classic Device Manager

5

Before Installing the BASEstar Classic Device Manager

This chapter is an overview of BASEstar software installation procedures and installation options. It also describes installation preparation, preliminary steps, and provides an overview of cluster installation.

5.1 Overview

The BASEstar installation procedures allow you to customize each BASEstar system by installing only those system components you want for the target system. You can install BASEstar software on a single node or on a cluster system, provided the cluster member nodes share the same system disk.

The main BASEstar installation procedure is mandatory for all BASEstar systems and installs basic application integration features, such as application synchronization and control and distributed name services. The main installation procedure has options, including the ability to install the BASEstar menu interface, BASEstar data management capabilities and BASEstar external data support.

5.2 Main Installation Options

Since the processing requirements for each node within your manufacturing system are distinct, the first step in installing BASEstar software is to determine which BASEstar components should be installed on each system.

5.3 Installation of the Device Connection Management Software

BASEstar device connection management software is installed after the main BASEstar installation procedure is complete. In order to communicate with plant equipment, BASEstar **device access software (DAS)** must also be installed for the industrial control devices used at each BASEstar site. Consult the manufacturing system designer or user group to understand which, if any, of this additional software must be installed.

The BASEstar device connection management installation procedure will conditionally install its menu interface component if the BASEstar menu system was installed during the main installation procedure. The device connection management installation procedure also includes installation of BASEstar device access software for RS-232 devices, DECnet and TCP/IP networks, and the DECdevice device communication module. Refer to the *BASEstar Open Guide to DCM-Modeled Device Connectivity* for information on how to configure and use these DAS modules.

5.4 Distribution Kit

The DCM software is available with the BASEstar Open kit, as the following software savesets:

- BCCM033 — BASEstar application integration software
- DCMM033 — BASEstar device connection management software

BASEstar technical documentation is available in either a development media kit or a run-time media kit. The run-time kit contains Volumes 1 and 2 of the documentation, and the development kit contains Volumes 1 to 4.

5.5 Pre-requisite Hardware

Refer to Section 1.4 for more information on the hardware required for the installation of DCM with BASEstar Open.

5.6 Pre-requisite and Optional Software

Refer to Section 1.5 for more information on the software required for the installation of DCM with BASEstar Open.

5.7 Pre-installation Steps

Perform these tasks before installing BASEstar software:

1. **Ensure OpenVMS, Version 5.3 or higher is installed on the system.**

   OpenVMS VAX Version, 5.3 or higher must be installed on each system planned for installation of BASEstar software.

2. **Ensure DECnet software is configured on your system.**

   DECnet software must be configured in order to install BASEstar software.

3. **Install the BASEstar Open product license.**

   Before you install and run the DCM software on a newly licensed node or cluster, you must first ensure that valid license for BASEstar Open has been installed.

   Table 5–1 shows the statuses of the licenses listed during the main BASEstar installation.

Table 5–1 BASEstar Licenses

License	Availability
BASESTAR	Available
BASESTAR-RT	Available
BASESTAR-CU	Available
BASESTAR-LTD	No longer available
BASESTAR-RT-LTD	No longer available
BASESTAR-G	Not yet available

(continued on next page)

Table 5–1 (Cont.) BASEstar Licenses

License	Availability
BASESTAR-RT-G	Not yet available
BASESTAR-USER	Not yet available

When performing a DCM installation, it is necessary to preload the licenses you want to use to authorize the use of DCM on the target node. A new license called BASESTAR-CU has been added for BASEstar Version 3.2. This license allows you to license a BASEstar system based on the maximum number of logical points residing on the target system.

At installation, the BASESTAR-CU license is checked for its presence and the installation procedure verifies that the number of logical points is less than the number licensed at startup. If the BASEstar parameter BCC$MAX_POINTS exceeds the number of units specified in the BASESTAR-CU license, the system cannot be started.

4. **Verify minimum run-time system parameters.**

   Table 5–2 describes the system parameter values recommended for the minimum run-time BASEstar configuration.

Table 5–2 BASEstar Minimum System Parameter Values

MODPARAMS.DAT Entries	Minimum Value
GBLSECTIONS	250
GBLPAGFIL	4,096
GBLPAGES	18,000
CTLPAGES	60
LOCKIDTBL	2,048
RESHASHTBL	512
MAXPROCESSCNT	35
BALSETCNT	33
VIRTUALPAGECNT	25,000
MAXBUF	4,096
SCSSYSTEMID	

Since these values represent minimum BASEstar run-time system parameter settings, you may need to adjust their values as required for systems larger than the minimum BASEstar hardware configuration or for systems running software in addition to BASEstar applications. For example, if you are using BASEstar library management and need to use large libraries, use the following steps to adjust the system parameter GBLPAGES:

5. **Stop interactive processes and disable logins.**

   Ensure that all BASEstar users have logged out and no other users can log in to the system during the installation. To disable users from logging in, enter:

```
$ SET LOGINS/INTERACTIVE=0
```

   Re-enable logins after the installation is complete. Refer to the *OpenVMS System Manager's Manual* for more information on logins.

6. **Stop batch queues.**

   To stop the batch queues, enter:

   ```
   $ STOP/QUEUE/NEXT  batch_queue_name
   ```

7. **Stop the BASEstar system.**

   If performing an upgrade of existing BASEstar software, ensure the current version of the system is not running. Stopping the BASEstar system also stops device connection management and all user-written BASEstar applications. To stop the BASEstar system, enter:

   ```
   $ BSTAR STOP SYSTEM
   ```

8. **Check VMSINSTAL requirements.**

   VMSINSTAL requires that the system account used to install BASEstar software have a minimum of the following quotas:

   - ASTLM=24
   - BIOLM=18
   - BYTLM=18,000
   - DIOLM=18
   - ENQLM=30
   - FILLM=20

   If VMSINSTAL detects any problems during the installation, it notifies you and asks if you want to continue the installation. In some instances, you can enter Y to continue. To stop the installation process and correct the situation, enter N or press Return. Then correct the problem and restart the installation.

9. **Back up the system disk.**

   Back up the system disk before installing the BASEstar software. This backup is a protection in the event of disk corruption or accidental changes or deletions during installation. The procedure for backing up the system disk is described in the *OpenVMS System Management Utilities Reference Manual*.

10. **Review release notes.**

    To read or print out the release notes included in the BASEstar distribution kit, invoke VMSINSTAL in the following manner:

    ```
    @SYS$UPDATE:VMSINSTAL BCCM033 mtcu: OPTIONS N

    Release notes included with this kit are always
    copied to SYS$HELP.

    Additional Release Notes Options:

    1.  Display release notes
    2.  Print release notes
    3.  Both 1 and 2
    4.  None of the above

    * Select option [2]:
    ```

    Select an option when the installation procedure prompts for release note options.

6

Installing the BASEstar Classic Device Manager

6.1 The Installation Procedure

Before you begin installing BASEstar Open, make sure that your system meets the requirements outlined in Chapter 1 and Chapter 5.

This chapter describes the step-by-step installation of the Device Connection Manager software.

BASEstar software components are installed using a combination of four procedures:

- Main BASEstar Open (single node or VAXcluster) — BCCM
- Device connection management (single node or VAXcluster) — DCMM

Read the instructions for installing BASEstar software. Follow the instructions for the installation of BASEstar device connection management, Refer to Appendix F for a sample installation procedures. All installation procedures will perform both new and upgrade installations.

Be prepared to provide the following information during the installation:

Step	√	Information to prepare before installation...
1.	_____	BCC$SPECIFIC directory location
2.	_____	BCC$COMMON directory location
3.	_____	Local name file estimated size

During the installation procedure, you are prompted to respond to questions. Type **Y** for yes or **N** for no and press Return or just press Return to select the default response. You can type a question mark and press Return after any prompt to receive on-line help.

To abort the installation procedure at any time, press Ctrl/Y. When you press Ctrl/Y, the installation procedure deletes all files it has created up to that point and exits. You can then start the installation again.

Note

DECnet software must be running during the installation. Do not shut down the network for installation. Even if you do not have a DECnet

license and your system is not networked, DECnet must be configured and running to install BASEstar software.

6.2 Installing BASEstar Software

This section describes how to install BASEstar software. Perform the following steps:

1. **Log in to the OpenVMS user account named SYSTEM.**

2. **Invoke VMSINSTAL.**

```
# @SYS$UPDATE:VMSINSTAL  BCCM033  ddcu:
```

   where **BCCM033** is the BASEstar product name and version and **ddcu:** is the name of the disk, tape drive or CD containing the software kit.

   The system displays the following:

```
VAX/VMS Software Product Installation Procedure V5.5-1

It is dd-mm-yyyy at hh:mm.

Enter a question mark (?) at any time for help.
```

3. **If you have active processes, VMSINSTAL lists them and asks if you want to continue.**

```
%VMSINSTAL-W-ACTIVE, The following processes are still active:
 DECW$WM_1
 DECW$TE_1
 _TWA2:
 _TWA3:
 _TWA4:
 _TWA9:
* Do you want to continue anyway [NO]?
```

4. **Confirm system backup.**

```
* Are you satisfied with the backup of your system disk [YES]?
```

   If you have not backed up your system, enter N and refer to *OpenVMS System Management Utilities Reference Manual*. If satisfied with your backup, press Return to continue.

5. **Mount the media.**

   If you are installing from a tape, you are prompted as follows:

```
Please mount the first volume of the save set on Nodename::

* Are you ready? Y
```

   The installation continues with the following:

```
The following products will be processed:

  BCCM V3.2

 Beginning installation of BCCM V3.2 at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been
  moved to SYS$HELP.
```

6. **Respond to license registration queries.**

```
BASEstar Reduced Kernel V3.2 Installation Procedure
---------------------------------------------------

   BASEstar requires one of these licenses to be registered
   and loaded on all nodes on which BASEstar will run:

            Product                    Producer    Version
            ---------------            --------    -------
        [1] BSTROPEN-DEVELOP           DEC         3.1
        [2] BSTROPEN-RUNTIME           DEC         3.1
        [3] BSTROPEN-DEVELOP-USER      DEC         3.1
        [4] BSTROPEN-RUNTIME-USER      DEC         3.1
        [5] BASESTAR-DCM               DEC         3.2

 * Which of the above licenses will be used? [1]:

        Product:      BASESTAR-DEVELOP
        Producer:     DEC
        Version:      3.1
        Release Date:

 * Does this product have an authorization key registered and loaded? y
```

The procedure prompts to ensure you have the correct BASEstar license installed on the system. Enter the appropriate number if the license is already installed. If not, abort the installation and install the BASEstar license.

7. **DECnet node information is verified and displayed:**

```
The DECnet node ID found for this node is:
  node-id
```

8. **Provide directory location.**

VMSINSTAL asks for a directory location for the BASEstar command and data files:

```
    A location must be chosen for the BCC$SPECIFIC directory.
    By default, BASEstar data and command files are installed in
    SYS$SPECIFIC:[BCC$SPECIFIC]. This specification may be
    overridden if desired.

 * Enter the BASEstar specific directory specification
   [SYS$SPECIFIC:[BCC$SPECIFIC]]:
```

If you do not specify a directory name, the BASEstar files are installed into the [BCC$SPECIFIC] directory under the system-specific root directory. If you enter a different directory, be sure it resides on a permanently mounted disk. (Since these files are data and command files, the system disk may not be an appropriate location.) If installing on a VAXcluster node and you do not use the system disk, specify a different directory name for each node within the VAXcluster system. Verify the directory name specified:

```
    The directory entered is:  SYS$SPECIFIC:[BCC$SPECIFIC]

 * Is this correct [YES]?
```

9. **Choose a location for the BCC$COMMON directory.**

By default, the BASEstar executable files are installed in SYS$COMMON: [BCC$COMMON]. This specification may be overridden.

```
      A location must be chosen for the BCC$COMMON directory.
      By default, BASEstar executable files are installed in
      SYS$COMMON:[BCC$COMMON].   This  specification  may be
      overridden if desired.

  * Enter the BASEstar common directory specification
    [SYS$COMMON:[BCC$COMMON]]:
    The directory entered is:  SYS$COMMON:[BCC$COMMON]
  * Is this correct [YES]?
```

10. **Read informational messages.**

```
_____________________________________

       The installation procedure has no further questions to ask and
      can  continue from  this point without user assistance.   The
      installation time  will  vary depending on the number of nodes
      selected  for installation and  the type of processor on which
      the installation is being done.

_____________________________________
1
      This installation creates a VMS rights identifier named BCC$USER,
      which is used to control access to BASEstar.

      After the installation is finished, the VMS Authorize Utility
      must be used to grant this identifier to all accounts that will
      use BASEstar.

      ------------------------------------------------------------
      A   R E M I N D E R   T O   T H E   I N S T A L L E R

      Each VMS user account that will issue the BSTAR commands must be
      granted the  BCC$USER rights identifier, as well as having
      certain Authorized privileges set.

      >> Remember to grant the BCC$USER identifier to the SYSTEM account
      >> (and/or any other accounts you will be using to start the BASEstar
      >> system) and to check the Authorized privileges of those accounts
      >> BEFORE YOU ATTEMPT TO START THE BASESTAR SYSTEM.
      ------------------------------------------------------------
```

The installation procedure restores the savesets, creates system directories, and displays the following:

```
  %VMSINSTAL-I-RESTORE, Restoring product save set B ...
  %VMSINSTAL-I-RESTORE, Restoring product save set C ...
  %VMSINSTAL-I-RESTORE, Restoring product save set D ...
  %VMSINSTAL-I-RESTORE, Restoring product save set E ...

  _____________________________________
  Installing BASEstar on node NODENAME:: ...
  _____________________________________

  %VMSINSTAL-I-SYSDIR, This product creates system disk directory
  VMI$ROOT:[BCC$COMMON].
  %VMSINSTAL-I-SYSDIR, This product creates system disk directory
  SYS$SYSDEVICE:[SYSn.BCC$SPECIFIC].

  %BCC-I-STARTCORE, BASEstar CORE installation beginning on NODENAME:: .

      This installation places  a file  named BCC$STARTUP.COM in the
      directory SYS$SPECIFIC:[SYS$STARTUP] in  each system  root  on
      which this installation is being performed. The SYSTARTUP_V5.COM
      file should be edited to invoke this startup file when the system
      boots.
```

```
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target
directories...
```

The BASEstar system disk directories are established, the startup command procedure is created, and the installation verification procedure is run.

```
BASEstar Reduced Kernel V3.2 Installation Verification Procedure (IVP)

$ DEFINE/NOLOG/PROCESS  BCC$SYSDATA  SYS$SPECIFIC:[BCC$SPECIFIC]
$ DEFINE/NOLOG/PROCESS  BCC$SYSTEM   VMI$ROOT:[BCC$COMMON]
$ set noverify
%BCCIVP-I-CORESTART, CORE verification starting on node NODENAME.

%BCCIVP-I-PASS, CORE data files check passed.
%BCCIVP-I-PASS, CORE command files check passed.
%BCCIVP-I-PASS, CORE executable files check passed.
%BCCIVP-I-PASS, CORE support files check passed.
%BCCIVP-I-PASS, CORE system files check passed.
%BCCIVP-I-PASS, CORE installed image check passed.
BASEstar Reduced Kernel V3.2 IVP completed successfully.
```

11. **End the installation procedure.**

```
Installation of BCCM V3.2 completed at hh:mm
VMSINSTAL procedure done at hh:mm
```

The BASEstar single node installation is complete. You can now log out of the privileged account if you do not want to install additional BASEstar software.

Continue with instructions later in this section to install additional BASEstar software or consult Chapter 7 for post-installation tasks.

6.3 Installing BASEstar Device Connection Management Software

This section describes how to install BASEstar device connection management software. If your main BASEstar installation did not include installing the menu interface, the device connection management installation cannot install or update the device connection management menus.

The BASEstar device connection management upgrade installation consists of the following steps:

1. **Log in to the OpenVMS user account named SYSTEM.**
2. **Invoke VMSINSTAL.**

```
# @SYS$UPDATE:VMSINSTAL  DCMM033  ddcu:
```

where **DCMM033** is the device connection management product name and version and **ddcu:** is the disk or tape drive containing the software kit. The system displays the following:

```
VAX/VMS Software Product Installation Procedure V5.4-3

It is dd-mmm-yyyy at hh:mm.

Enter a question mark (?) at any time for help.
```

3. **If you have active processes, VMSINSTAL lists them and asks if you want to continue.**

```
%VMSINSTAL-W-ACTIVE, The following processes are still active:
 DECW$WM_1
 DECW$TE_1
 _TWA3:
 _TWA4:
 _TWA5:
 _TWA6:
* Do you want to continue anyway [NO]?
```

4. **Confirm system backup.**

```
* Are you satisfied with the backup of your system disk [YES]?
```

If you have not backed up your system, enter N and refer to *OpenVMS System Management Utilities Reference Manual*. If satisfied with your backup, press Return to continue.

The installation continues with the following:

```
The following products will be processed:

  DCMM V3.2

 Beginning installation of DCMT V3.2 at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product save set A ...

   BASEstar Device Connection Management (DCMM) V3.2 Installation Procedure
   ------------------------------------------------------------------------
```

5. **Existing BASEstar Installation Message**

If BASEstar is already installed on your system, the following message is displayed:

```
BASEstar is installed on this system.
The file SYS$COMMON:[SYSLIB]BCC$SHR.EXE was found.
Its image file identification is:  BCC T3.n
```

6. **Respond to node information**

The procedure determines if the executing node is a member of a multinode VAXcluster. If the executing node is a standalone node or clustered with only one active member, the following message is displayed:

```
    This node is not a member of a multi-node VAXcluster.

    A single-node installation will be performed.

_____________________________________
Processing node NODENAME::
_____________________________________
```

If the executing node is a member of a multinode cluster, you will be prompted to indicate if you want a single-node installation:

```
This node is a member of a multi-node VAXcluster.

    A single-node installation or a multi-node installation
    can be chosen.

    For a multi-node installation, you will be prompted to
    enter the node names to be processed.  The nodes must
    share a cluster-common system disk.  Also, the current
    node (the node presently executing VMSINSTAL) will
    automatically be the first node processed.

* Do you want to install DCMM on just this node  [YES]?
```

If you answer N, the following text is displayed:

```
    ----------------------------------------
    A cluster installation has been selected.
    ----------------------------------------

The current node,  NODENAME,  will be processed first.

Checking node  NODENAME::  ...

    BASEstar is installed on this node.  The file
      SYS$SPECIFIC:[SYS$STARTUP]BCC$STARTUP.COM
    was found.

The node passed all checks.
```

7. **Enter additional nodes**

   If you are performing a cluster installation, you will be prompted to enter the additional nodes on which you want to install BASEstar software:

```
  You will now be prompted to enter the additional cluster nodes
    on which BASEstar DCM is to be installed.  End the list with a
    carriage return <CR>.

    (Enter Control-Y to abort the installation, if necessary.)

Current node name list is:  NODENAME

* Enter another node name (or <CR> to end the list): NODENAME2

Checking node  NODENAME2::  ...

   BASEstar is installed on this node.  The file
     SYS$SPECIFIC:[SYS$STARTUP]BCC$STARTUP.COM
    was found.

    ------------------------------------------------
    Wait -- using SYSMAN utility to check certain
    logical names on cluster member ...
      ... Done.
    ------------------------------------------------

The node passed all checks.

Current node name list is:  NODENAME1, NODENAME2

 Enter another node name (or <CR> to end the list):

2 valid node names were entered:
   NODENAME1, NODENAM2

* Is this list correct? y
```

   The installation continues:

```
_____________________________________
Processing node 1, NODENAM1::
_____________________________________

    The BASEstar Menu System is present.
    The DCM Menu System will be installed on this node.
```

8. **Choose size of file.**

```
  You must choose a initial size for the PHYSICAL POINT data file:

    ILAN$PHYPOINT_FILE
    ___________________

      Initial Allocation      Extend Allocation      File Size
```

```
      [1]    500 Phypoints         500 Phypoints       1,251 Disk Blocks
      [2]  2,000 Phypoints       1,000 Phypoints       3,333 Disk Blocks
      [3] 10,000 Phypoints       5,000 Phypoints      15,744 Disk Blocks
      [4] 20,000 Phypoints      10,000 Phypoints      32,604 Disk Blocks

* Enter ILAN$PHYPOINT_FILE size [1]:
```

Select a file size for the physical point data file. Enter the correct choice from the menu and press Return.

If you are performing a cluster installation, VMSINSTAL processes each node in the cluster and you select the size of the PHYSICAL POINT data file.

9. **Read informational messages.**

```
    Post-installation messages for the installer
    ------------------------------------------------------------

    ILAN$STARTUP.COM, a startup command procedure, is used
    to set up the environment for Device Connect Management.
    During the installation, this file will be placed in the
    SYS$SPECIFIC:[SYS$STARTUP]  directory of the nodes on
    which this installation is being performed.

    If this is a first-time DCM installation on this VAX system,
    the system startup command procedure

                SYS$MANAGER:SYSTARTUP_V5.COM

    needs to be manually edited (after this installation has
    successfully finished).   Locate the line

                $  @ SYS$STARTUP:BCC$STARTUP.COM

    and add this new line immediately after the BCC$STARTUP
    line:

Press RETURN to continue

                $  @ SYS$STARTUP:ILAN$STARTUP.COM

  ------------------------------------------------------------
    The installation procedure has no further questions to ask
    and can continue from this point without user assistance.
    The time needed to finish the installation will vary
    depending on the number of nodes selected for installation
    and the type of processors and disks involved.
    ------------------------------------------------------------


_____________________________________
Installing DCM on node NODENAME1:: ...
_____________________________________

    Creating DCM data files for this node ...

    Providing images to BCC$SYSTEM ...

  Providing DCM example PL/I programs to sub-directory
    of BCC$SYSTEM ...

%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DUA0:[SYS0.S YSCOMMON.BCC$COMMON.DCM_EXAMPLE].

    Providing RS232 DAS source files to sub-directory
    of BCC$SYSTEM ...

%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DUA0:[SYS0.S YSCOMMON.BCC$COMMON.RS232_EXAMPLE].

    Providing Network DAS NCT source files to sub-directory
    of BCC$SYSTEM ...
```

```
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
$1$DUA0:[SYS0.S YSCOMMON.BCC$COMMON.NCT_EXAMPLE].

   ... Done.

    Checking the BASEstar Parameter file for
    "ILAN$*"  (DCM)  parameters ...

    ... Done.

    Installing the DCM Menu System on this node ...
    ... Done.
```

If you are performing a cluster installation, the informational messages for each node in the cluster are displayed and the installation continues:

```
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target
directories...

IVP phase beginning on node NODENAME1 ...

%SYSMAN-I-ENV, current command environment:
        Individual nodes: NODENAME1
        Username SYSTEM       will be used on nonlocal nodes

%SYSMAN-I-TIMEVAL, timeout value is  00:11:00.00
%SYSMAN-I-OUTPUT, command execution on node NODENAME1

$  @SYS$STARTUP:ILAN$STARTUP.COM

$  @SYS$TEST:ILAN$IVP.COM

------------------------------------------------------------------------
BASEstar DCM V3.2 Installation Verification Procedure (IVP) starting.
------------------------------------------------------------------------

Checking logical name  BCC$SYSDATA  ...

Checking logical name  BCC$SYSTEM  ...

Checking ILAN files ...
Checking files in SYS$COMMON:[BCC$COMMON.DCM_EXAMPLE] ...
Checking files in SYS$COMMON:[BCC$COMMON.NCT_EXAMPLE] ...
Checking files in SYS$COMMON:[BCC$COMMON.RS232_EXAMPLE] ...
%ILAN$IVP-I-PASS, File check passed.

Checking ILAN installed images ...
%ILAN$IVP-I-PASS, Installed image check passed.

Checking  "ILAN$*"  BASEstar parameters ...
%ILAN$IVP-I-PASS, Parameter check passed.

------------------------------------------------------------------------
BASEstar DCMM V3.2 IVP completed successfully.
------------------------------------------------------------------------
```

If you are performing a cluster installation, the IVP messages for each node in the cluster are displayed.

10. **End the installation procedure.**

```
    Installation of DCMT V3.2 completed at hh:mm

        VMSINSTAL procedure done at hh:mm
```

The BASEstar device connection management installation is complete. You can now log out of the privileged account if you do not want to install additional BASEstar software.

Continue with instructions in this chapter for installing additional BASEstar software or consult Chapter 7 for post-installation tasks.

7

After Installing the BASEstar Classic Device Manager

This chapter describes the Device Connection Manager post-installation tasks.

7.1 The Device Connection Manager Post-installation Procedures

When you have completed the Device Connection Manager installation procedures, log in to the system manager's account and perform the following steps:

1. **Delete BASEstar Ada files if not needed (optional).**

   Certain Ada declarations files are rather large; if no Ada programming will be done at your site, you may want to delete the file SYS$COMMON:[SYSLIB]BCC_.ADA (~1400 blocks) and the ILAN*.ADA files in that same directory.

2. **Back up the system disk.**

   Back up the system disk after installing BASEstar software. The procedure for backing up the system disk is described in the *OpenVMS System Management Utilities Reference Manual*.

3. **Edit the system startup procedure.**

   If you want BASEstar software to automatically start when the system boots, add the following line to the SYS$MANAGER:SYSTARTUP_V5.COM file:

   ```
   $ @SYS$STARTUP:BCC$STARTUP.COM
   ```

   To automatically start BASEstar device connection management when the system boots, add the following line to the SYS$MANAGER:SYSTARTUP_V5.COM file:

   ```
   $ @SYS$STARTUP:ILAN$STARTUP.COM
   ```

   To automatically start BASEstar library management when the system boots, add the following line to the SYS$MANAGER:SYSTARTUP_V5.COM file:

   ```
   $ @SYS$STARTUP:BCCLIB$STARTUP.COM
   ```

   For a new installation, this procedure creates the library management database files and associated global sections. During an upgrade installation, this command procedure locates the existing database files and creates the library management database global sections.

4. **Set process quotas.**

   Ensure BASEstar user processes have quotas equal to or greater than those listed in Table 7–1.

Table 7–1 Process Resource Limits

Limit	Minimum Value	Description
ASTLM	100	AST queue limit
BIOLM	80	Buffered I/O count limit
BYTLM	20,000	I/O byte count limit
DIOLM	80	Direct I/O count limit
ENQLM	200	Enqueue quota
FILLM	100	Open file limit
PGFLQUO	20,000	Paging file limit
PRCLM	6	Subprocess creation limit
TQELM	20	Timer queue entry limit
WSDEF	500	Default working set size
WSEXTENT	1,500	Working set extent
WSQUO	800	Working set quota

5. **Select BASEstar users.**

   Grant the BCC$USER identifier to the SYSTEM account and/or any other accounts you will be using to start or stop the BASEstar system using the VMS Authorize Utility (AUTHORIZE) as follows:

   ```
   UAF> GRANT/IDENTIFIER  BCC$USER  username
   ```

   where **username** is the name of the user account.

6. **Set privileges.**

   Ensure that each BASEstar user has the following minimum OpenVMS privileges:

   - TMPMBX
   - NETMBX

7. **Edit SYS$STARTUP:BCC$STARTUP.COM if you installed the Monitor BASEstar utility.**

   Edit the file SYS$STARTUP:BCC$STARTUP.COM to include the following line:

   ```
   $ DEFINE/NOLOG/SYSTEM BCC$MONITOR_DATA -
     BCC$SYSDATA:BCC$MON_BSTAR_APP.DAT
   ```

8. **Edit BCC$SYSDATA:BCC$SYSTEM_STARTUP.COM File**

   You should modify the BCC$SYSDATA:BCC$SYSTEM_STARTUP.COM file on every node in the domain to include a single DCL command that executes a 20 second wait after starting the BCC$DISPATCHER process and before starting the BCC$LOGGER process.

   This modification temporarily fixes a problem found on some faster machines where processes were accessing BCC$DISPATCHER data before it was ready. This premature access caused BASEstar to crash on startup and threatened to crash the system. This problem will be fixed in a future release.

Modify the BCC$SYSDATA:BCC$SYSTEM_STARTUP.COM file as follows:

```
/MAILBOX='F$GETDVI("BCC$TERMINATION_MAILBOX","UNIT") -
/ERROR=BCC$SYSDATA:BCC$DISPATCHER.ERR -
/OUTPUT=BCC$SYSDATA:BCC$DISPATCHER.LOG -
/PROCESS=BCC$DISPATCHER -
/PRIORITY=8 -
BCC$SYSTEM:BCC$DISPATCHER.EXE
>>>$!
>>>$! wait to give the dispatcher a chance to initialize the
>>>$! message global section before other processes begin
>>>$! mapping to it.
>>>$!
>>>$ wait 0:00:20.00
$!
$! Start the BCC$LOGGER process.
$!      Note:  The default sizes here are for a MicroVAX II system.
$!
$       RUN/DETACHED/INPUT=NLA0: -
/AST_LIMIT=100 -
/BUFFER_LIMIT=38000 -
/ENQ=200 -
/FILE_LIMIT=100 -
/IO_BUFFERED=80 -
```

9. **Ensure DECnet is running.**

   BASEstar software will not run unless DECnet software is running. If DECnet is stopped, BASEstar software will shut down.

10. **Start BASEstar software.**

    BASEstar software can be started from the SYSTEM account after this account has been granted the BCC$USER identifier. It can also be started from any account having the privileges listed in Table 7–2. These privileges must be in the account's Authorized Privilege list and are added using AUTHORIZE.

Table 7–2 VMS Privileges Required to Start BASEstar Software

ALTPRI	DETACH	LOG_IO
NETMBX	PHY_IO	PSWAPM
SHARE	SYSGBL	SYSLCK
SYSNAM	TMPMBX	

Though accounts used to start BASEstar require these privileges in their Authorized Privilege list, they do not require them in their Default Privilege list for security reasons. Table 7–2 represents the minimum privileges needed by all BASEstar processes and applications. Some BASEstar processes may not need all privileges and can be started with only a subset. Whenever new applications need additional privileges not in this list, the additional privileges can be added to the account's Authorized Privilege list. The new privileges are in effect only after the user logs out and logs back in.

Note

The SYSTEM account must be granted the BCC$USER identifier regardless of whether it will be used to start BASEstar. Failure to grant

the BCC$USER identifier to the SYSTEM account will cause BASEstar startup attempts to fail.

To start BASEstar software, enter the following command at the DCL prompt:

```
$ BSTAR START SYSTEM
```

11. **Device Connection Manager Logicals Generated**

    All logical names created by the Device Connection Manager installation procedures are /SYSTEM logical names. The logical names do not exist until the appropriate SYS$SPECIFIC:[SYS$STARTUP]xxxxxx$STARTUP.COM procedure is invoked. Table D–6 lists the logical names created by the installations.

Part III

Environment Management

This section contains a series of chapters that describe the procedures and utilities that allow you to manage BASEstar Open Server on your OpenVMS system.

Note

The management procedures described in this section are also valid for BASEstar Open Client on OpenVMS platforms.

8

Introduction to Environment Management

An application consists of a set of application components; that is, user-written programs and BASEstar Open-supplied servers running on a distributed system configuration, including various types of computers, plant devices and cables. To provide users with transparent access to BASEstar Open objects, regardless of their physical location in the distributed system, you must prepare your BASEstar Open environment. For example, you must set up and start up Nodes and Realms.

8.1 Environment Management Overview

This section provides an overview of the BASEstar Open environment management operations.

Setting Up BASEstar Open

To set up BASEstar Open, you must set up Nodes and Realms. Setting up a Node involves creating the static resources (directories, files and environment-specific components) that enable a system of any supported platform to function as a BASEstar Open Node. Setting up a Realm involves creating directories and files that contain object definitions within a Realm.

Starting Up BASEstar Open

To make the BASEstar Open environment available to users, you must start up Nodes and Realms.

Shutting Down BASEstar Open

To make BASEstar Open unavailable to users, you must shut down Nodes and Realms in an orderly fashion.

Unsetting BASEstar Open

To unset BASEstar Open, you must delete Realm and Node directories and files, and the database containing the Realm object definitions.

Monitoring BASEstar Open

At any time, you can establish whether the BASEstar Open environment has been set up and started up successfully on Nodes and Realms. For example, you can display the list of the Realms available (that is, started up) on a given Node.

8.2 Environment and Application Components Relationship

Once you have completed the setup and startup operations for the BASEstar Open environment, you can request the Application Management Services to start up, monitor, and shut down the application components (that is, user-written programs and BASEstar Open-supplied servers).

When you shut down BASEstar Open, you must first shut down the application components and then the environment components. You can shut down the application components using, for example, the Application Management Services. The *BASEstar Open Command Language Interface* describes how to configure, start up and shut down the application components using Application Management Services features.

Chapter 9 describes the procedures that you must use to set up, start up, shut down and unset the environment components.

8.3 BASEstar Open Environment Components

The components of a BASEstar Open environment are:

- Nodes
- PODB Nodes
- Realms

In this context, Nodes and Realms are not considered as objects, but as a set of environment components that make Nodes and Realms available in the multi-Node, multi-platform, distributed BASEstar Open environment.

8.3.1 Nodes

A BASEstar Open Node is a computer of any of the supported platforms on which BASEstar Open has been installed and set up. To make a Node available in the BASEstar Open environment, you must start up its components by executing the *bstr_node_start* command.

The following are Node-specific components:

- Log Services
- Watchdog
- Name Service

Each of these components is described in the sections that follow. See also Appendix I for a list of the processes that implement components on each platform.

Log Services

On each Node, the Log Services server collects log information from Node-specific environment components and from the environment components of the Realms that are started up on each Node. The Log Services server writes the collected information to the appropriate log files. This information is valuable for monitoring the behavior of environment components and also facilitates troubleshooting activity when necessary.

For information about Log Services features and functions, and about how to display and manage log files, refer to Chapter 14 and Chapter 15.

Watchdog

The Watchdog is a background task that BASEstar Open starts when you execute the *bstr_node_start* procedure. The Watchdog component of a Node performs periodic checks on the availability of the other Nodes in a BASEstar Open environment. If a Node suddenly becomes unavailable, or if a BASEstar Open application running on one of the configured Nodes terminates abnormally, the Watchdog notifies the relevant components.

Name Service

The Name Service component of a Node makes internal symbols and their values unique and globally available to BASEstar Open. The Name Service consists of:

- A Name Service client process that must be active on each Node
- A Name Service server that must be active on one Node of the distributed BASEstar Open multi-Node environment.

The Name Service server is fundamental to the availability of the BASEstar Open environment. If the server fails, the BASEstar Open environment becomes unavailable. To avoid this, a redundancy mechanism is provided, based on a primary and a secondary copy of the Name Service server. When the Nodes are started up, the primary copy is activated on the specified Node and becomes the current Name Service server. At the same time, BASEstar Open activates the secondary copy in stand-by mode on another Node. If the current Name Service server fails or becomes unavailable for any reason, BASEstar Open activates the secondary copy, which thus becomes the current Name Service server.

The Node setup procedure allows you to specify on which Nodes BASEstar Open activates the primary and secondary copies of the Name Service server.

8.3.2 PODB Nodes

When you set up the BASEstar Open environment, you can choose which BASEstar Open Nodes you want to act as PODB Nodes (that is, contain the PODB object definitions). However, bear in mind the following considerations:

- You can only designate ONE Node to act as the PODB Node for a given Realm
- It is possible to designate a given Node as the PODB Node for different Realms.

  The example in Section 8.3.4 illustrates that NODE_DB has been set up as the PODB Node for both MY_REALM and TS_REALM. Alternatively, you could set up NODE_DB as the PODB Node for MY_REALM, and NODE1 (or NODE2) as the PODB Node for TS_REALM.

BASEstar Open supports various databases for the storage of PODB object definitions, depending to the native operating system. On OpenVMS Nodes, you can use an VAX Rdb/VMS database.

When you start up your environment, you must also start up the Database Services server on the PODB Node of each Realm that requires object definitions to be stored in the PODB.

8.3.3 Realms

User-written application processes and BASEstar Open-supplied servers are the application components of each Realm. These components are managed directly by the end-users through the Application Management Services or from the command interpreter.

Before starting up Realm application components, you must start up the Realm environment components on the appropriate Node(s). When you start up a Realm on a Node, BASEstar Open activates the following environment components:

- Communication Service
- Global Object Services server
- Application Management Services server

- Database Services server
- PC Communication server

BASEstar Open activates the per Node Realm-specific components when you execute the *bstr_realm_start* command. The sections that follow provide a brief description of each component. See also Appendix I for a list of the processes that implement components on each platform.

Communication Service

The Communication Service allows transparent communication between application and environment components of a Realm. This communication is independent of the number of Nodes on which the Realm has been started up.

For example, the Communication Service features allow you to access a Data_Point (through an API or a CLI service request) without needing to know the physical location of the Data & Device Services server that makes the Data_Point available.

Global Object Services Server

Global Object Services servers make global objects available in a Realm. Global objects are those that already exist before servers that use them can be started up. For example, a Data & Device Services server must already find the associated Domain in the VODB, otherwise it could not create objects in the VODB.

Global objects also include those objects that must be globally available in the Realm to all the other servers. For example, the same Datatype could be referred to by Event Services and Data & Device Services objects, and therefore it must be available in the Realm, regardless of whether any other server is available.

A Global Object Services server is started up on each Node where a Realm is started up. This allows distributed management of the global objects; all the Global Object Services servers of a distributed Realm work together to make available to any user or to other servers all the same global objects independently of their location.

Application Management Services Server

When you start up a Realm on a given Node, the Application Management Services server is also activated and makes the Application Management Services objects and features available on that Node. This enables users at any Realm Node to request services such as executing, suspending, resuming and terminating Activities.

As soon as you start up the Realm on other Nodes, an Application Management Services server is activated on each of them, thus allowing distributed management of the Realm application components.

For example, you can execute an Activity on a Realm Node from any Node of the Realm. As another example, if a Realm Node fails, the Application Management Services servers running on the other Nodes of the Realm cooperate to restart the Realm application components on the Node(s) specified by the user in the configuration.

To provide such dynamic and distributed services, the Application Management Services servers of a Realm continuously notify each other of their respective availability and operations in progress.

PC Communication Server

One PC Communication server is activated on a Node when a Realm is started up on that Node. BASEstar Open applications running on client Nodes can operate in transparent mode on the objects of a Realm through any PC Communication server active for that Realm. This is independent of the Realm Node on which the servers that make these objects available have been started.

See the *BASEstar Open Client Inst. & Management Guide* for details of how to install, configure and manage BASEstar Open Client on PC platforms.

Database Services Server

It is recommended that you activate a Database Services server on the Realm PODB Node. You only need to start up a Database Services server for Realms that require object definitions to be stored in the Permanent Object Database.

The Database Services server is activated when you execute the *bstr_realm_start_db* command.

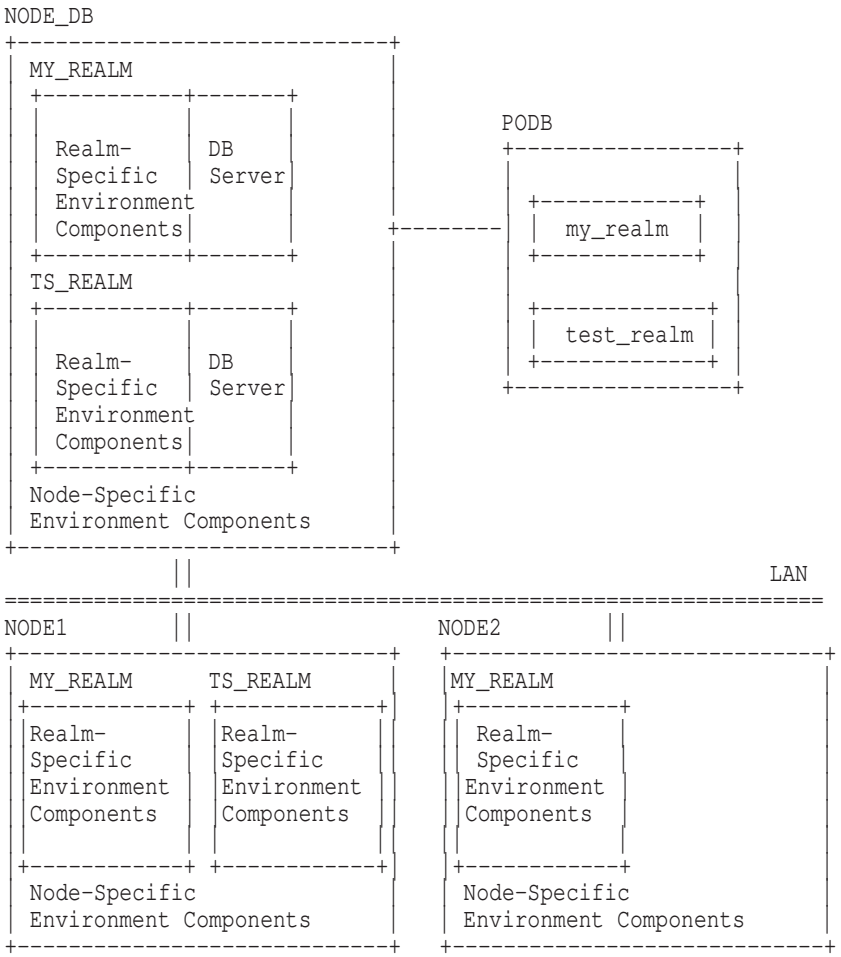
8.3.4 Sample Environment Configuration

Figure 8–1 is an example of a BASEstar Open environment that consists of three systems interconnected through a LAN. To make this sample environment available, you must:

1. Set up NODE_DB, NODE1 and NODE2. In the example, NODE_DB acts as the PODB Node for both MY_REALM and TS_REALM.
2. Start up each Node.
3. Set up MY_REALM on each Node.
4. Set up TS_REALM on each Node.
5. Start up the environment components of MY_REALM on each Node where MY_REALM must be available (that is, on all the Nodes).
6. Do the same for TS_REALM, but on NODE_DB and NODE1 only.
7. On the PODB Node, start up a Database Services server for each Realm.

In Figure 8–1, it is assumed that NODE_DB is a Compaq Tru64 UNIX system capable of acting as the PODB Node (ORACLE is the selected database). It is also assumed that NODE1 is a OpenVMS system and NODE2 is a Compaq Tru64 UNIX system. However, there are numerous alternatives to this configuration; for example, you could also have MS Windows client Nodes.

Figure 8–1 Sample BASEstar Open Environment



9

Environment Management Procedures

This chapter describes the procedures that allow you to manage the BASEstar Open environment. You must perform the operations in the order prescribed:

1. Set up Nodes, Realms and databases on PODB Nodes. A setup operation creates files and directories for the corresponding component.
2. Start up Nodes, Realms and Realm PODB servers. A startup operation runs the system processes that constitute the active part of the given component.
3. Shut down Nodes, Realms and Realm PODB servers. A shutdown operation terminates the system processes in an orderly fashion.
4. Unset Realms, Nodes and Realm PODB servers. An unset operation deletes files and directories for a specified component.

You must set up a Realm before you can start it up. However, it is not necessary to set up all the components in order to start up just one of them.

The BASEstar Open environment management procedures described in this chapter refer to the BASEstar Open environment commands documented in Chapter 10. Before issuing any of these commands, read the command description given in Chapter 10, which also highlights pre-requisites and constraints.

9.1 Setup Procedures

This section describes the procedures that you must use to set up Nodes and Realms. Execute the procedure described in Section 9.1.1 for each Node that you wish to set up, and the procedure described in Section 9.1.2 for all the Realms in your environment.

For information regarding the parameters used by the procedures described in this chapter, refer to the relative man pages.

9.1.1 Setting Up a Node

To create Node-specific directories and configuration files, use the *bstr_node_setup* command. This command allows you to select the Node on which to activate the Name Service server. If you use the advanced *bstr_node_setup* option, it allows you to select the BASEstar Open working directory root database, and the communication configuration parameters.

When you set up a Node, BASEstar Open usually asks you to specify just the names of the Nodes on which you want it to start the primary and secondary copies of the Name Service servers.

The *bstr_node_setup* command also creates the *bstrusers.** command files, for setting up installation-dependent global variables (see Appendix G).

9.1.2 Setting Up a Realm

You must set up a Realm for each Node on which the Realm is to be available, as explained in Section 9.1.2.1. If you wish to store Realm object definitions in the PODB, create a database for the Realm on the PODB Node (see Section 9.1.2.2).

9.1.2.1 Setting Up a Realm on a Node

The *bstr_realm_setup_node* command allows you to create the directories and files that are required at run-time by the internal Realm components started on the Node.

9.1.2.2 Setting Up the Realm Database on the PODB Node

To create a database for a Realm, run the *bstr_realm_setup_db* command on the PODB Node.

9.1.3 Setting Up the Sample Environment

The following command sequence demonstrates how to set up the BASEstar Open environment for the sample configuration illustrated in Figure 8–1. Note that these instructions cause BASEstar Open to activate the primary copy of the Name Service server on NODE2 and the secondary copy on NODE1. You can set up the Nodes shown in this example in any order.

For details of the rules governing the execution of environment commands, refer to Chapter 10.

Setting Up the Nodes

Log in as SYSTEM on NODE1 and issue the following command:

```
# bstr_node_setup  -l WORK1:[BASESTAROPEN032]  -y
  BSTR.Spool_Work_Dir: DUA1:[BASESTAROPEN032.WRK]
  BSTR.Spool_Snap_Dir: DUA1:[BASESTAROPEN032.SNP]
  LNS.Server_Node [NODE1]: NODE2
  LNS.Server_Node_2[]: NODE1
  ...
```

Execute the following command on NODE2:

```
# bstr_node_setup  -l WORK1:[BASESTAROPEN032]  -y
   ...
   LNS.Server_Node [NODE2]: NODE2
   LNS.Server_Node_2[]: NODE1
   ...
```

Execute the following command on NODE_DB:

```
# bstr_node_setup  -l WORK1:[BASESTAROPEN032]  -y
  ...
  LNS.Server_Node [NODE_DB]: NODE2
  LNS.Server_Node_2[]: NODE1
  ...
```

Setting Up the MY_REALM Realm

To set up the MY_REALM Realm on NODE1, NODE2 and NODE_DB, execute the appropriate setup command file on each Node:

```
# @BSTR$KIT:[ETC]BSTRUSERS.COM
```

Then issue the following command on NODE1 (you can set up the Realms on the Nodes in any order):

```
$ bstr_realm_setup_node MY_REALM
```

Then issue the following command on NODE2:

```
$ bstr_realm_setup_node MY_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_setup_node MY_REALM
$ bstr_realm_setup_db  MY_REALM
```

Setting Up the TS_REALM Realm

To set up the TS_REALM Realm on NODE1 and NODE_DB, issue the following command on NODE1:

```
$ bstr_realm_setup_node TS_REALM
```

The setup operations for TS_REALM are similar to those performed for MY_REALM. Note that TS_REALM has not been set up on NODE2. This is because there was no request for the Realm to be available on this Node.

Execute the following commands on NODE_DB:

```
$ bstr_realm_setup_node TS_REALM
$ bstr_realm_setup_db  TS_REALM
```

9.2 Startup Procedures

This section describes the procedures that allow you to start up a Node or a Realm in the BASEstar Open environment. You must execute the procedure described in Section 9.2.1 for each Node that you wish to start up.

Execute the procedure described in Section 9.2.2 for each Realm that you wish to make available in your environment.

9.2.1 Starting Up a Node

The *bstr_node_start* command activates the node-specific components, and checks for license and environment integrity on the Node. Note that the started Nodes only become available when you start up the Node on which the Name Service server is activated.

9.2.2 Starting Up a Realm

Start up the Realm on each Node on which you want the Realm to be available (see Section 9.2.2.1). You must also start up the Database Services server process (allowing access to the Realm database) on the PODB Node (see Section 9.2.2.2).

9.2.2.1 Starting Up a Realm on a Node

The *bstr_realm_start_node* command starts up Realm-specific components on the Node on which it is executed. It also checks for license and environment integrity on the Node.

Note

BASEstar Open components that are active on a Node cannot access components that are active on other Nodes until you start up the Node where the Name Service server resides.

9.2.2.2 Starting Up the Database Server for a Realm on the PODB Node

You must only use the *bstr_realm_start_db* command on the PODB Node. This command activates the Database Services server for the specified Realm.

9.2.3 Starting Up the Sample Environment

The command sequence in this section demonstrates how to start up the BASEstar Open environment for the sample configuration illustrated in Figure 8–1.

Starting Up the Nodes

You can start up the sample Nodes in any order. Execute the following command on NODE1:

```
$ bstr_node_start
```

Execute the following command on NODE2:

```
$ bstr_node_start
```

Execute the following command on NODE_DB:

```
$ bstr_node_start
```

Starting Up the MY_REALM Realm

To start up MY_REALM on NODE1, NODE2 and NODE_DB, execute the command sequence illustrated below (you can start up the Realm on the Nodes in any order).

First, you must set up the MY_REALM Realm on NODE1, NODE2 and NODE_DB. Execute the appropriate setup command file on each Node:

```
# @BSTR$KIT:[ETC]BSTRUSERS.COM
```

Then issue the following command on NODE1:

```
$ bstr_realm_start_node MY_REALM
```

Execute the following command on NODE2:

```
$ bstr_realm_start_node MY_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_start_node MY_REALM
$ bstr_realm_start_db MY_REALM
```

If you execute the above command sequence correctly, BASEstar Open makes the MY_REALM Realm available. At this point, you can use, for example, the Application Management Services to start up the Realm application components as explained in the *BASEstar Open Command Language Interface*.

Starting Up the TS_REALM Realm

To start up TS_REALM on NODE1 and NODE_DB, execute the following command on NODE1:

```
$ bstr_realm_start_node TS_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_start_node TS_REALM
$ bstr_realm_start_db TS_REALM
```

The startup operations for TS_REALM are similar to those performed for MY_REALM. Note that TS_REALM has not been started up on NODE2, since there was no request for the Realm to be available on this Node.

9.3 Shutdown Procedures

This section describes the procedures that allow you to shut down a Node and a Realm in the BASEstar Open environment. You must execute the procedure described in Section 9.3.1 for each Realm that you wish to be make unavailable in your environment.

You must also execute the procedure described in Section 9.3.2 for each Node to be made unavailable in your environment.

9.3.1 Shutting Down a Realm

To shut down a Realm in an orderly fashion, you must first shut down the application components. To do so, use either the command interpreter on your platform or the Application Management Services.

You must then shut down the active Database Services server on the Realm PODB Node (see Section 9.3.1.1). Finally, you must shut down the Realm on each Node where the Realm was started, as explained in Section 9.3.1.2.

Note

It is possible to shut down a Realm on one Node and leave it available on the remaining Nodes. However, it is recommended that you shut down all the Realm application components in an orderly fashion on all Realm Nodes before shutting down the Realm environment on the Node.

9.3.1.1 Shutting Down the Database Server for a Realm on the PODB Node

The *bstr_realm_shut_db* command stops the Realm Database Services server. You can only perform this operation on the PODB Node.

9.3.1.2 Shutting Down a Realm on a Node

The *bstr_realm_shut_node* command makes a Realm unavailable on a Node. It also shuts down any active Realm-specific components on the Node.

9.3.2 Shutting Down a Node

The *bstr_node_shut* command deactivates the node-specific environment components. Note that once you have completed the shutdown operation for the Node on which the Name Service server is running, BASEstar Open becomes unavailable on all other Nodes unless you activate a secondary copy of the server on a Node that is still active.

9.3.3 Shutting Down the Sample Environment

The command sequence in this section demonstrates how to shut down the BASEstar Open environment for the sample configuration illustrated in Figure 8–1.

Shutting Down the MY_REALM Realm

You can shut down MY_REALM on NODE1, NODE2 and NODE_DB in any order. Before shutting down the Realm on the PODB Node, you must first issue the *bstr_realm_shut_db* command on that Node.

Execute the appropriate setup command file on each Node:

```
# @BSTR$KIT:[ETC]BSTRUSERS.COM
```

Execute the following command on NODE1:

```
$ bstr_realm_shut_node MY_REALM
```

Execute the following command on NODE2:

```
$ bstr_realm_shut_node MY_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_shut_db  MY_REALM
$ bstr_realm_shut_node MY_REALM
```

Shutting Down the TS_REALM Realm

To shut down TS_REALM on NODE1 and NODE_DB, issue the following command on NODE1:

```
$ bstr_realm_shut_node TS_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_shut_db  TS_REALM
$ bstr_realm_shut_node TS_REALM
```

The shutdown operations for TS_REALM are similar to those performed for MY_REALM.

Note

You do not need to be a superuser (root) to shut down a Realm. However, the user who shuts down the Realm must be the same user that started up the Realm.

Shutting Down the Nodes

Once you have shut down MY_REALM and TS_REALM in an orderly fashion, you can shut down the Nodes in any order.

Execute the following command on NODE1:

```
$ bstr_node_shut
```

Execute the following command on NODE2:

```
$ bstr_node_shut
```

Execute the following command on NODE_DB:

```
$ bstr_node_shut
```

9.4 Unset Procedures

This section describes the procedures that allow you to unset Realms and Nodes. Execute the procedure described in Section 9.4.1 for all the Realms in your environment, and the procedure described in Section 9.4.2 for all the Nodes to be unset.

9.4.1 Unsetting a Realm

Unsetting a Realm implies unsetting it for all the Nodes on which it was set, as described in Section 9.4.1.1. In addition, you must delete the PODB database created for that Realm, as explained in Section 9.4.1.2.

9.4.1.1 Unsetting a Realm from a Node

To unset a Realm from a Node, execute the *bstr_realm_unset_node* command on the Node in question. This command deletes all Realm-specific directories and files previously created by the *bstr_realm_setup_node* command from the file system of the Node.

The PODB Node also performs the actions described in Section 9.4.1.2.

9.4.1.2 Unsetting a Realm Database on the PODB Node

To unset a Realm database on the PODB Node, execute the *bstr_realm_unset_db* command on the PODB Node. This command deletes the database for the specified Realm.

Note

If you delete the database for a specified Realm, BASEstar Open also deletes all PODB object definitions.

9.4.2 Unsetting a Node

The *bstr_node_unset* command deletes the directories and node-specific files previously created by the *bstr_node_setup* command from the file system of the Node.

9.4.3 Unsetting the Sample Environment

The command sequence in this section demonstrates how to unset the BASEstar Open environment for the sample configuration illustrated in Figure 8–1.

Unsetting the MY_REALM Realm

You can unset the MY_REALM Realm from NODE1, NODE2 and NODE_DB in any order. Execute the following command on NODE1:

```
$ bstr_realm_unset_node MY_REALM
```

Execute the following command on NODE2:

```
$ bstr_realm_unset_node MY_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_unset_db  MY_REALM
$ bstr_realm_unset_node MY_REALM
```

Unsetting the TS_REALM Realm

To unset the TS_REALM Realm on NODE1 and NODE_DB, issue the following command on NODE1:

```
$ bstr_realm_unset_node TS_REALM
```

Execute the following commands on NODE_DB:

```
$ bstr_realm_unset_db  TS_REALM
$ bstr_realm_unset_node TS_REALM
```

The operations required to unset TS_REALM are similar to those performed for MY_REALM. Note that TS_REALM has not been unset from NODE2. This is because it was not set on this Node.

Unsetting the Nodes

You can unset the Nodes shown in the example in any order. Execute the following command on NODE1:

```
$ bstr_node_unset
```

Execute the following command on NODE2:

```
$ bstr_node_unset
```

Execute the following command on NODE_DB:

```
$ bstr_node_unset
```

9.5 Monitoring BASEstar Open

In addition to the general management procedures, you can use the *bstr_mon* tool (or the *bstr_env_show* command) to monitor the active components of the BASEstar Open environment. It displays:

- All available Realms
- For each Realm, the Nodes on which the Realm is available, and the available Domains and VMDs
- A list of available Nodes and Realms available on each Node.

For a complete description of the *bstr_mon* tool, refer to Chapter 11. Refer to Chapter 10 for a description of the *bstr_env_show* command.

You can also use the *bstr_realm_check_env* command to check whether the BASEstar Open environment has been set up and started successfully on a given Node.

10

Environment Management Command Reference

10.1 Executing Environment Commands

This section outlines the general requirements for executing environment commands successfully. The description of each individual command includes specific command requirements.

If you execute a command successfully, BASEstar Open returns the following exit status:

```
1 (one)
```

If a command fails, it returns a different value in the command exit status. It also displays one or more messages that provide information about the error(s) that occurred during execution of the command. The log file often contains additional information about the components used to execute the command (see Chapter 14).

10.1.1 Requirements and Constraints

This section explains which users can execute specific environment management commands, and specifies the rules that you must follow when executing a command.

10.1.1.1 Users and Commands

Not all OpenVMS users can execute all the environment management commands. Table 10–1 identifies which user can execute a given command, where

- *Root* indicates the SYSTEM user or any user with all privileges.
- *Spool user* indicates the user responsible for managing a given Node. For a given Node, the *spool user* is the SYSTEM user (default) or the user whose name was specified when the *bstr_node_setup* command was executed for that Node.

  The *spool user* must have all the privileges.
- *Realm user* indicates the user responsible for managing a given Realm. For a given Realm, the *Realm user* is the OpenVMS user who executed the *bstr_realm_setup_node* command for that Realm.

  As a minimum requirement, the *Realm user* needs the BYPASS, TMPMBX, DETACH, GRPNAM and NETMBX privileges, and the following quotas:

```
FILLM = 20
BYTLM = 80,000
ASTLM = 20
BIOLM = 18
DIOLM = 18
ENQLM = 20
JTQUOTA = 200
PGFLQUOTA = 80000
PRCLM = 10
TQELM = 20
WSDEFAULT = 256
WSQUOTA = 512
WSEXTENT = 8192
```

When you select the *Realm user*, you must remember that the Application Management Services server has the same UIC of the *Realm user*. This means that the Application Management Services server sees also the logical names of the group to which the *Realm user* belongs.

In addition, as the Application Management Services server has the same privileges as the *Realm user*, you must remember to assign to the *Realm user* all the privileges that must also be assigned to the Program executions that you want it to run.

Note

If *root*, *spool user* or *Realm user* is specified for a command, it means that only that particular user can execute the command.

Table 10–1 OpenVMS Relationship Between Environment Management Commands and Users

Command	Type of User
Monitoring Environment Components	
bstr_env_show	Realm user
bstr_realm_check_env	Realm user
Managing Nodes	
bstr_node_setup	Root
bstr_node_start	Spool user
bstr_node_shut	Spool user
bstr_node_unset	Root

(continued on next page)

Table 10–1 (Cont.) OpenVMS Relationship Between Environment Management Commands and Users

Command	Type of User
	Managing Realms
bstr_realm_setup_node	Realm user
bstr_realm_start_node	Realm user
bstr_realm_shut_node	Realm user
bstr_realm_unset_node	Realm user
bstr_realm_setup_db	Realm user
bstr_realm_start_db	Realm user
bstr_realm_shut_db	Realm user
bstr_realm_unset_db	Realm user
	Invoking the CLI
cli_bstr	Realm user

10.1.1.2 Executing a Command

You must follow the rules listed below when executing an environment command (except for *bstr_node_setup*, for which refer to the instructions given in the command description):

- Log in as the appropriate user, as specified in Table 10–1, on the selected Node.
- Ensure that the installation-dependent global variables have been correctly set by executing the *bstrusers.com* command procedure (see Appendix G for details).

For specific information on command requirements and constraints, refer to the description of each individual command.

10.2 Privileges for Running Environment Management Procedures

Table 10–2 lists the privileges required to run the various BASEstar Open Server environment management commands.

Table 10–2 Environment Management Privileges

Phase	Procedure	Privileges Required for Execution
Set up BASEstar Open	*bstr_node_setup*	All
Start up BASEstar Open	*bstr_node_start*	All
Shut down BASEstar Open	*bstr_node_shut*	CHKRNL
Unset BASEstar Open	*bstr_node_unset*	All
Set up a Realm	*bstr_realm_setup_node*	None
Start up a Realm	*bstr_realm_start_node*	DETACH

(continued on next page)

Table 10–2 (Cont.) Environment Management Privileges

Phase	Procedure	Privileges Required for Execution
Shut down a Realm	*bstr_realm_shut_node*	None
Unset a Realm	*bstr_realm_unset_node*	None
Set up a database	*bstr_realm_setup_db*	None
Start up a database	*bstr_realm_start_db*	DETACH
Shut down a database	*bstr_realm_shut_db*	None
Unset a database	*bstr_realm_unset_db*	None
Invoke the CLI	*cli_bstr*	None

cli_bstr

Invokes the BASEstar Open CLI.

Syntax

cli_bstr [-c] [-e *"command1" "command2"..."command10"***] [-f** *command_file***] [-i** *startup_file***] [-v]**

Description

The *cli_bstr* command allows you to invoke the BASEstar Open CLI. You can execute CLI commands either interactively, or in batch mode by specifying the name of a command file from the command interpreter prompt.

Options

-c

This option allows you to check the syntax and semantics of a CLI command file without executing the file.

-e "*command1*" "*command2*"..."*command10*"

This option allows you to execute up to ten CLI commands without leaving the platform-dependent command interpreter. After executing the specified commands, the command interpreter redisplays the CLI prompt. Enclose each CLI command between double quotes, and make sure that the commands are separated by at least one blank.

-f *command_file*

Pathname of a CLI command file that BASEstar Open executes at the shell prompt.

-i *startup_file*

An optional startup command file that you can request the CLI to execute. If you fail to specify the -i option, BASEstar Open does not execute a startup command file unless the following CLI command file exists in your home directory:

```
cli_init.ini
```

-v

This option displays additional information on execution of the command.

Examples

1. `$ cli_bstr` Return

   ```
   BSTR>
   ```

   This command invokes the CLI so that you can execute commands in interactive mode.

2. $ **cli_bstr -e "SET SCOPE VOLATILE" "SHOW DATA_POINT /DOM1/DP3"** Return
   $

   This command sequence sets the VODB scope and displays the /DOM1/DP3 Data_Point.

3. $ **cli_bstr -f exec2.com** Return
   $

   This command executes the exec2.com CLI command file at the shell prompt.

See Also

None

bstr_env_show

This command displays information about the environment components (Nodes and Realms) and services (server activities, PODB and Application Management Services servers currently started up in the BASEstar Open environment.

Syntax

bstr_env_show -q *query* **[-s]** | **-d**

Description

The *bstr_env_show* command allows you to display information about the components active in your BASEstar Open environment. It displays information on Nodes, Realms, or services, depending on what is specified in the *query* argument of the **–q** qualifier.

The *bstr_env_show* command displays the components that are active at the time the command is executed (it does not provide information on components that have been set up but not started).

To execute the *bstr_env_show* command:

- Log in as a generic user (see Section 10.1 for details).
- Check that the BASEstar Open environment has been started up successfully on the Node on which you want to execute the *bstr_env_show* command (see Chapter 9 for details). Note that this command accesses the Node where the Name Service server is active.
- Ensure that the BASEstar Open installation-dependent global variables have been appropriately set, by executing the `bstrusers.sh` or `bstrusers.csh` script file (see Appendix G for details).

If the command completes successfully, a zero value is returned in the command exit status, otherwise a value other than zero is returned.

Lower case qualifiers (with the associated argument) must be enclosed between double quotes (").

Options

-q *query*
The *query* expression specifies the kind of information displayed by the *bstr_env_show* command. The following expressions are valid:

- **N**

  Displays a list of available BASEstar Open Nodes, that is, the Nodes for which the *bstr_node_start* command has been executed, and whose components have been started up successfully.

- **R**

  Displays, for each Realm, a list of the BASEstar Open Nodes on which the Realm is available. (In this context, a Realm is considered to be available on a given Node if the Communication Service is found active and running on that Node.)

- **n[:***node_name***]**

Displays a list of the Realms currently available on the *node_name* Node. The services currently available for each Realm are also shown, together with the resources made available by each type of service. (In this context, a Realm is considered to be available on a given Node if the Communication Service component is found active and running on that Node.)

Several lines may be displayed, each containing the name of a Realm, the identifier of a service currently available on that Realm (ams, dtm, dvm, evm, or podb), and the associated resource (if significant).

If you do not specify the *node_name* argument, BASEstar Open assumes by default the name of the Node on which you are executing the command.

- **r[:***realm_name***]**

  Displays, for the *realm_name* Realm, a list of the services available on the Nodes where *realm_name* is available, together with the list of resources that are made available by each service. (In this context, a Realm is considered to be available on a given Node if the Communication Service component is found active and running on that Node.)

  Several lines may be displayed, each containing the identifier of an available service (ams, dtm, dvm, evm, or podb), the name of the Node on which the corresponding server is started up, and the associated resource (if significant).

  If you do not specify the *realm_name* argument, the name of the Realm set in the BSTR_REALM global variable is assumed by default.

- **s:[***realm_name* **]:ams|dtm|dvm|evm|mgm|podb**

  Displays, for the *realm_name* Realm, a list of the available services of a specified type (ams, dtm, dvm, evm, or podb), and informs you of the Node on which they are available. The resource (Domain or VMD) made available by each service is also displayed, where applicable.

  Several lines may be displayed, each containing the name of the Node on which the corresponding server is started up, and the associated resource (if significant).

  If you do not specify the *realm_name* argument, the name of the Realm set in the BSTR_REALM global variable is assumed by default.

- **d:[***realm_name***]:***domain_name*

  Displays, for the *domain_name* Domain belonging to the *realm_name* Realm, which Event Services server and Data Services server Activities (evm or dtm) are available and on which Node the corresponding server has been started up. You must supply the full name of a Domain in the *domain_name* argument.

  Several lines may be displayed, each containing the identifier of the available service and the name of the Node where the service is available.

  If you do not specify the *realm_name* argument, the name of the Realm set in the BSTR_REALM global variable is assumed by default.

-s
Not allowed.

-d
Displays a set of formatted records, which contain all the information that can be returned by the *bstr_env_show* command.

Examples

1. $ **bstr_env_show -q N** Return

```
***     BASEstar Nodes Started     ***

 NODE
 ****
 sabato
 dennix
```

This command displays the names of all the Nodes that are currently available in the BASEstar Open environment.

2. $ **bstr_env_show -q R** Return

```
***     BASEstar Realms Started     ***

 REALM    NODE
 *****    ****
 PEASANT sabato
 TEST     sabato
 TEST     dennix
```

This command displays the names of all the Realms that are currently available in your BASEstar Open environment. The name of the Node (or Nodes) on which the Realms are started is also displayed. The TEST Realm is started on both sabato and dennix Nodes, while the PEASANT Realm is started on the sabato Node only.

3. $ **bstr_env_show -q n:sabato** Return

```
***     BASEstar Realms and Services for Node sabato     ***

 REALM    SERVICE DOMAIN VMD
 *****    ******* ****** ***
 PEASANT AMS
 PEASANT PODB
 TEST     AMS
 TEST     DTM     /DTM_1
 TEST     DTM     /DTM_2
 TEST     DTM     /DTM_3
 TEST     EVM     /EVM_1
 TEST     EVM     /EVM_2
 TEST     MGM     /MGM_1
```

This command displays the list of services that are available on Node sabato, sorted by Realm.

4. $ **bstr_env_show -q r:TEST** Return

```
***     BASEstar Services for Realm TEST     ***

   SERVICE NODE   DOMAIN VMD
 ******* ****   ****** ***
 AMS     sabato
 DTM     sabato /DOMAIN_1
 DTM     sabato /DOMAIN_2
 DTM     dennix /DOMAIN_3
 EVM     sabato /DOMAIN_1
 EVM     dennix  /DOMAIN_2
```

This command displays the names of all the services that are started up in the TEST Realm. For each service, the command also displays the name of

the Node where the service has been started up, and the name of the resource (or resources) made available by that service. In the example, both the Data Services server and the Event Services server have been started for the /DOMAIN_1 domain, and both on Node sabato.

5. $ **bstr_env_show -q s:TEST:evm** Return

```
***     BASEstar Service EVM for Realm TEST     ***

 NODE   DOMAIN
 ****   ******
 sabato /DOMAIN_1
 dennix /DOMAIN_2
```

This command shows that there are two active Event Services in the TEST Realm, for /DOMAIN_1 and /DOMAIN_2, respectively. It also shows the names of the Nodes where the corresponding Event Services servers have been started up (sabato and dennix Nodes, respectively).

6. $ **bstr_env_show -q d:TEST:/DOMAIN_2** Return

```
***     BASEstar Services for Domain /DOMAIN_2 of Realm TEST     ***

 SERVICE NODE
 ******* ****
 EVM     sabato
```

This command shows that the server for the DOMAIN_2 Domain is currently available on Node sabato.

See Also

None

bstr_node_setup

This command creates a series of directories and configuration files on the Node on which you execute it. It allows you to define configuration parameters relative to the Name Services, BASEstar Open working directories, database and communication parameters, and miscellaneous run-time parameters.

Syntax

bstr_node_setup -l *location* **[-y | -adv] [-v]**

Description

You must execute the *bstr_node_setup* command for each Node on which you want to run BASEstar Open. You can only execute this command on a Node where BASEstar Open is installed.

Before running the *bstr_node_setup* command, you must log in as SYSTEM. The *bstr_node_setup* command is in the `ETC` directory, under the BASEstar Open installation directory.

You cannot execute the *bstr_node_setup* command if there are BASEstar Open processes currently active (all environment and application components must be shut down first).

The *bstr_node_setup* command performs the following operations:

- Creates the configuration files for the Node-specific environment components.
- Creates the command procedures that can be used to set the values of the installation-dependent BASEstar Open global variables (see Appendix G for details).
- Creates the subtree that stores the BASEstar Open temporary files created at run-time.

To display some or all of the configuration parameters, select either **–y** (default), or **–adv**.

Options

-l *location*
The pathname of the directory where the BASEstar Open kit is installed. It must be an absolute OpenVMS file specification.

-y
Simplified Node setup. The only value you can supply is the name of the BASEstar Open Nodes where the primary and secondary copies of the Name Services server were activated.

-adv
Advanced Node setup. Same as **-y** option. In addition, you can change the name and characteristics of the subtree under which the BASEstar Open temporary and snapshot files are stored at run-time.

-v
This option displays additional information on execution of the command.

Usage Notes

While executing the *bstr_node_setup* command, press Return to accept the default value displayed between square brackets. Press Ctrl/C at any time to abort the *bstr_node_setup* command.

The lines displayed by the *bstr_node_setup* command start with different character strings, depending on the operation being performed. The following character strings can be displayed:

- The `@@@` string identifies a message that describes the operation in progress.
- The `&&&` string identifies an error message (error or fatal error).

Lines that are displayed without any of the above character strings are input requests.

Note

The name of the less extensive option that displays a given parameter is specified within brackets. In other words, if the **–y** option is specified, it means that the parameter is also displayed by the **-adv** option.

Changing Configuration Parameters

The *bstr_node_setup* command allows you to display and change the values of some configuration parameters. All the *bstr_node_setup* command prompts and displays are listed below.

```
BSTR.Spool_Work_Dir []:
```

(**–y** option). Enter a file specification for the Node-specific BASEstar Open working directory (for example `disk3:[work_naxos]`). The directory in question must not already exist. You must specify a dedicated working directory for each Node. You cannot specify a snapshot directory (see next prompt) or a descendant directory of a snapshot directory.

This parameter is mandatory.

```
BSTR.Spool_Snap_Dir []:
```

(**-y** option). Enter a file specification for the directory that contains the snapshot files for the Node that you are setting set up. If no snapshot directory exists, BASEstar Open creates one. The specified directory must not be a working directory (see previous question) or a descendant directory of a working directory. It is possible for several Nodes to share the same snapshot directory.

This parameter is mandatory.

```
BSTR.Spool_Owner [current_user]:
```

(**-adv** option). Name of the user who owns the working and snapshot directory trees. The specified user must exist.

```
BSTR.Spool_Protection_Mode "(S:rwed,O:rwed,G:rwed,O:rwed)
```

(**-adv** option). Protection mask applied to the working and snapshot directory trees. The default value restricts directories access to the owner (BSTR.Spool_Owner) and the group (BSTR.Spool_Group).

```
LNS.Server_Node [current_node]:
```

(**-y** option). Name of the Node on which BASEstar Open is to activate the primary copy of the LNS name server. This parameter must have the same value on all the configured BASEstar Open Nodes.

```
LNS.Server_Node_2 []:
```

(**-y** option). This parameter is optional and specifies the name of the Node on which BASEstar Open is to activate the secondary copy of the LNS name server. If a value is entered for this parameter, it must be the same on all the Nodes.

Note

Note that the secondary copy of the LNS name server cannot be activated on the same Node specified for the primary copy.

Complete or Abort Command Execution

The following prompt allows you to interrupt execution of the command:

```
Do you want to continue or not? [y]:
```

(**-y** option). Enter **y** to terminate command execution by leaving the system unaltered, enter **n** to continue.

Creating Configuration and Command Procedures

After displaying the values of the configuration parameters (and allowing you to modify them if necessary), the *bstr_node_setup* command displays a series of messages informing you that configuration and script files are being created or edited. The files created by *bstr_node_setup* are:

- The `installation_file.h_`*`node_name`*, which contains all the configuration parameters entered in response to the *bstr_node_setup* command prompts, together with a few reserved configuration parameters.
- The `lns.conf.h_`*`host_name`* file, which contains all the configuration parameters entered in response to the *bstr_node_setup* command prompts regarding the LNS network service.
- The `SYS$STARTUP:BSTR$STARTUP.COM` command procedure, which performs the following operations:
  - Defines the BSTR$KIT logical name with the file specification of the installation directory.
  - Executes the `bstrusers.com` command file.
  - Executes the *bstr_node_start* command.
- The `bstrusers_<`*`node_name`*`>.com` command procedure, which allows users to set installation-dependent global variables (see Appendix G for details).

Creating BASEstar Open Work Directories

Finally, the *bstr_node_setup* command creates the basic BASEstar Open work directories using the specified values.

On completion, the *bstr_node_setup* command returns you to the DCL prompt.

Examples

1.
```
$ @DISK3:[BSTROSVV320.ETC]bstr_node_setup -l DISK3:[BSTROSVV320] -y Return
Name of the work directory. You must introduce the path name
    (according to  the rules of VAX/VMS - .e.g. "DISK:[SUBD1.SUBD2]" )
    of an UNEXISTING directory :

BSTR.Spool_Work_Dir []: DISK3:[BSTROSVV320.work] Return

Name of the snapshot directory. You must introduce the path name
    (e.g. "DISK:[SUBD1.SUBD2]") of a directory. If it doesn't exists,
    it will be created by the procedure, otherwise you'll share it
    with some  other installations.
 WARNING : Snapshot directory cannot be the same as work directory
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

BSTR.Spool_Snap_Dir []: DISK3:[BSTROSVV320.snapshot] Return

Name of the host where the primary LNS Server will run:

LNS.Server_Node [NODE1]: NODE2 Return

 Name of the host where the secondary LNS Server will run:
     --- Note that the primary and secondary LNS Server Hosts
     cannot coincide.

LNS.Server_Node_2 []: NODE1 Return

If you are not satisfied with the settings of installation
parameters you can terminate the procedure and run it again:

Do you want to continue or not ? [Y] : Return

@@@ creating SYS$STARTUP:BSTR$STARTUP.COM;1 file

@@@ creating BSTR_ETC:bstrusers_NODE1.com file

@@@ creating BSTR_ETC:bstradmin_NODE1.com file

@@@ creating BSTR_ETC:lns_NODE1.conf file

$
```

If you are logged in to NODE1, the *bstr_node_setup* command creates the BASEstar Open static structures on NODE1: BASEstar Open expects the primary copy of the Name Services server to be available on NODE2 and the secondary copy on NODE1.

The other parameters are set to their default values.

bstr_node_shut

Shuts down the Node-specific environment components.

Syntax

bstr_node_shut [-h] [-v]

Description

The *bstr_node_shut* command shuts down the Node-specific components.

Before issuing this command, you are advised to shut down all the Realms that are available on the Node (issue the *bstr_realm_shut_node* command for each Realm, or specify the **–h** option).

Options

-h

If you specify this option, the *bstr_node_shut* command also shuts down all the Realms that are available on the Node. In other words, the *bstr_realm_shut_node* command is automatically issued for each available Realm.

If you do not specify this option (default), the *bstr_node_shut* command checks whether there are any Realms started up on the Node. If the command finds one or more Realms still active, it lists them and terminates without performing any actions.

-v

This option displays additional information on the execution of the command.

Examples

1. $ **bstr_node_shut** Return

   If you are logged in to NODE1, the *bstr_node_shut* command shuts down the node-specific environment components on that Node.

See Also

bstr_node_start

bstr_node_start

Starts up the Node-specific environment components.

Syntax

bstr_node_start [-v] [-c]

Description

The *bstr_node_start* command starts up the Node-specific environment components and checks for license and environment integrity on the Node.

Execute the *bstr_node_start* command for each Node on which you want to start up BASEstar Open. Before running the *bstr_node_start* command, execute the *bstr_node_setup* command to set up the Node. Execute this command before you run the *bstr_realm_start_node* and *bstr_realm_start_db* commands.

Starting up a Node does not affect operations on other Nodes, where one or more Realms may already have been started. However, remember that any Nodes you start up become available when you start up the Node where the Name Service server resides.

On completion, the *bstr_node_start* command outputs messages notifying you of its successful completion (all the Node-specific components have been started up), or unsuccessful completion (no components have been started up).

Options

-v

This option displays additional information on execution of the command.

-c

If you specify this option, the *bstr_node_start* command performs a cleanup of all the temporary files for the current Node and all the Realms that have been set up on it. For example, it deletes working files and purges the Log Services files.

Examples

1. $ **bstr_node_start** Return

   If you are logged in to NODE1, the *bstr_node_start* command starts up the Node-specific components on that Node.

See Also

bstr_node_setup

bstr_node_shut

bstr_node_unset

Deletes the directories and configuration files previously created using the *bstr_node_setup* command from the Node where the command is executed.

Syntax

bstr_node_unset [-l *location* **] [-v]**

Description

You cannot issue the *bstr_node_unset* command if there are BASEstar Open processes active (all environment and application components must be shut down first).

Before issuing the *bstr_node_unset* command, shut down all the Realms that are currently available on the Node and the Node itself. The *bstr_node_unset* command performs the following actions:

- Deletes the Node-specific BASEstar Open working directory.
- Deletes the snapshot directory specified for the Node. Because a snapshot directory can be shared among many Nodes, the shared snapshot directory (which resides on only one Node) is physically deleted only when all the Nodes that share it have been unset.

When executed on a PODB Node, this command also deletes all the DBMS structures that were created using the *bstr_realm_setup_db* command. The DBMS configuration files are also deleted.

Options

-l *location*
Pathname of the directory where the BASEstar Open kit is installed. It must be the same pathname as that specified while setting up BASEstar Open.

-v

This option displays additional information on the execution of the command.

Examples

1. $ **bstr_node_unset** Return

   If you are logged in to NODE1, this command deletes the BASEstar Open static structures on that Node.

See Also

bstr_node_setup

bstr_realm_check_env

Checks whether the BASEstar Open environment has been set up and started up successfully.

Syntax

bstr_realm_check_env[*realm_name***]**

Parameters

realm_name
Specifies the local name of the Realm that you want to check. If you specify this parameter, *bstr_realm_check_env* checks that *realm_name* matches the value of the BSTR_REALM global variable.

Description

The check is performed on the Node on which the command is issued. Once invoked, the *bstr_realm_check_env* command performs the following:

- If the *realm_name* parameter is specified, it checks whether the BSTR_REALM global variable has been set and whether its value is equal to *realm_name*.
- Checks whether the installation-dependent global variables have been set (that is, if the appropriate `bstrusers` script has been executed successfully).
- Checks whether the directories specified in the installation-dependent global variables exist and have appropriate contents (that is, if the *bstr_realm_setup_node* command has been executed successfully on the Node).
- Checks whether the account you are using is authorized for performing operations on the BASEstar Open environment (see Section 10.1 for details).

If any of the above checks fails, the *bstr_realm_check_env* command displays an error message and returns control to the command interpreter. Otherwise, a series of messages are displayed to inform you that the *realm_name* Realm has been set up successfully on the Node. In addition, the *bstr_realm_check_env* command tells you whether or not the Realm has been started up successfully on the Node.

To issue the *bstr_realm_check_env* command:

- Log in as a generic user on the Node for which you wish to check the BASEstar Open environment (see Section 10.1 for details).
- Ensure that the BASEstar Open installation-dependent global variables have been set successfully, by executing the `bstrusers.sh` or `bstrusers.csh` script file (see Appendix G for details).
- Set BSTR_REALM with the name of the Realm to be checked.

If the command completes successfully, a zero value is returned in the command exit status, otherwise a value other than zero is returned.

Examples

1. $ **setenv BSTR_REALM MY_REALM** Return
   bstr_realm_check_env Return

```
          BASEstar Open

Realm Environment Check for the Realm: MY_REALM

Global Environment Variables: Correctly Set.

Working Environment Existence: Correctly built.

Working Environment access: Accessible.

          REALM : MY_REALM

Currently Active
```

   If you are logged in to NODE2, this command checks whether the MY_REALM Realm has been set up successfully.

   All preliminary checks were successful and MY_REALM has been started up on NODE2.

See Also

None

bstr_realm_setup_db

Creates a DBMS-dependent database where you can store PODB object definitions for the specified Realm.

Syntax

bstr_realm_setup_db *realm_name* **[-c] [-l** *location***] [-RDB** *version***]**

Parameters

realm_name
Specifies the local name of the Realm for which the database is to be created.

The database takes its name from the specified Realm.

Description

Issue this command on the Realm PODB Node only, after you have set up the PODB Node, using the *bstr_node_setup* command.

If you do not specify the **–l** qualifier, the following message is displayed:

```
Please enter the database location [BSTR_WORK_ROOT:[REALM_realm_name]]:
```

Enter the specification of the directory where the VAX Rdb/VMS database is to be created.

The *bstr_realm_setup_db* command first checks whether any snapshots still exist for *realm_name*. If there are, it displays an error message and does not create the PODB database. To force the command to delete the snapshots, you must specify the **-c** qualifier.

The *bstr_realm_setup_db* command writes additional information to the `<realm_name>_setup_db.log` file created under the BSTR_WORK_ROOT:[REALM_*realm_name*.TMP] directory.

Options

-c

Forces the command to delete any existing snapshots for *realm_name*.

-l *location*
Pathname of the directory where the VAX Rdb/VMS database is created. It must be an OpenVMS file specification. If you do not specify this qualifier, the *bstr_realm_setup_db* command prompts you to enter the name of the location directory.

-RDB *version*
The version of DEC Rdb to be used (if you are running a multi-version).

See Also

bstr_node_setup

bstr_realm_setup_node

Creates directories and files to be used at run-time by the Realm-specific environment components started on the Node.

Syntax

bstr_realm_setup_node *realm_name* **[-v]**

Parameters

realm_name
Specifies the local name of the Realm for which the structures (files and directory) are to be created.

The *realm_name* string can have a maximum of 8 uppercase characters and must follow the syntax rules of a BASEstar Open local name.

Description

Before issuing the *bstr_realm_setup_node* command, issue the *bstr_node_setup* command to set up the Node.

Options

-v

This option displays additional information on the execution of the command.

Examples

1. $ **bstr_realm_setup_node MY_REALM** Return

   If you are logged in to NODE1, the *bstr_realm_setup_node* command creates the BASEstar Open static structures for MY_REALM.

See Also

bstr_node_setup
bstr_realm_unset_node

bstr_realm_shut_db

Shuts down the Database Services server for the specified Realm.

Syntax

bstr_realm_shut_db *realm_name*

Parameters

realm_name
Specifies the local name of the Realm for which the Database Services server is to be shut down.

Description

Execute the *bstr_realm_shut_db* command on the PODB Node only. This must be the first operation you perform in the *realm_name* shutdown process. Before issuing the command, you must ensure that there are no transactions in progress on the database. In other words, check that all the application components have been shut down in an orderly fashion, and that no CLI or Graphic Configuration utility operators are working on the PODB definitions for *realm_name*.

Examples

1. $ **bstr_realm_shut_db MY_REALM** Return

   If you are logged in to the PODB Node (NODE_DB), the *bstr_realm_shut_db* command shuts down the Database Service server for MY_REALM.

See Also

bstr_realm_start_db

bstr_realm_shut_node

Shuts down Realm-specific environment components for the Node on which it is executed.

Syntax

bstr_realm_shut_node *realm_name* **[-h] [-v]**

Parameters

realm_name
Specifies the local name of the Realm whose Realm-specific environment components you wish to shut down.

Description

Execute the *bstr_realm_shut_node* command on a given Node for a given Realm to make the specified Realm unavailable on the Node.

The *bstr_realm_shut_node* command performs the following steps:

- Shuts down the PC Communication server.
- Shuts down the Application Management Services server and therefore terminates the Activities and Programs under its control.
- Shuts down the Global Object Services server.
- Shuts down the Communication Service. The Communication Service processes are only stopped after termination of the application components.

Note that the **–h** option causes the Communication Service to terminate unconditionally, without waiting for the Application Components to terminate.

Before issuing this command, shut down the Realm application components as appropriate. If you are addressing a PODB Node, execute the *bstr_realm_shut_db* command before executing the *bstr_realm_shut_node* command.

Options

-h

Not allowed.

-v

This option displays additional information on the execution of the command.

Examples

1. $ **bstr_realm_shut_node MY_REALM** Return

   If you are logged in to NODE1, this command shuts down the internal components on the specified Node for MY_REALM.

See Also

bstr_realm_start_node

bstr_realm_start_db

Starts the Database Service server for the specified Realm.

Syntax

bstr_realm_start_db *realm_name*

Parameters

realm_name
Specifies the local name of the Realm for which the Database Service server is to be started.

Description

Execute the *bstr_realm_start_db* command on the Realm PODB Node only.

This command starts up the Database Service server, which allows BASEstar Open users to access the Realm PODB in order to create, update and delete object definitions for *realm_name*.

Before executing the *bstr_realm_start_db* command, use the *bstr_realm_setup_db* command to create the appropriate database, and start up the Realm on the PODB Node using the *bstr_realm_start_node* command.

See Also

bstr_realm_setup_db
bstr_realm_shut_db

bstr_realm_start_node

Starts up the realm-specific environment components on the Node.

Syntax

bstr_realm_start_node *realm_name* **[-v] -w [-nocs]** *work_dir*

Parameters

realm_name
Specifies the local name of the Realm for which the environment components are to be started up.

Description

Execute the *bstr_realm_start_node* command on a given Node for a given Realm to make the specified Realm available on the Node.

Before executing the *bstr_realm_start_node* command, start up the Node using the *bstr_node_start* command, and set up the Realm on the same Node using the *bstr_realm_setup_node* command.

Note that the user-defined global variables that you set before executing this command are visible to the Application Management Services server. Therefore, they are also visible to all processes (that is, Programs) started by the server.

See Section 12.3.1 for further details.

In addition, if you wish to specify a different port number for the PC Communication server, set the new value in the BSTR_PC_PORT global variable.

The *bstr_realm_start_node* command performs the following steps:

1. Starts up the Communication Service.
2. Starts up the Global Object Services server.
3. Starts up the Application Management Services server.
4. Starts up the PC Communication server.
5. Checks for license and environment integrity on the Node.

Options

-w *work_dir*

This option specifies the pathname of the directory where BASEstar Open stores Realm-specific work files. The specified directory must exist.

-nocs

This option allows you to disenable the mechanism that automatically starts up the Communication Server for the Realm. If you do not need to connect a client, *-nocs* allows you to save an environment process for each Realm.

-v

Displays additional information on the execution of the command.

bstr_realm_start_node

Examples

1. $ **bstr_realm_start_node MY_REALM** Return

   If you are logged in to NODE1, the *bstr_realm_start_node* command starts up MY_REALM on that Node.

See Also

bstr_realm_setup_node

bstr_realm_shut_node

bstr_realm_unset_db

Deletes the database that contains the PODB definitions for the specified Realm.

Syntax

bstr_realm_unset_db *realm_name* **[-s]**

Parameters

realm_name
Specifies the name of the Realm whose database you wish to delete.

Description

Execute the *bstr_realm_unset_db* command to delete the database associated with *realm_name*. BASEstar Open deletes the specified database, together with its contents.

Note

Deleting the database for a specified Realm means deleting all its PODB object definitions.

The *bstr_realm_unset_db* command performs the physical deletion of the PODB database for the specified Realm only if there are no other Nodes still set for that Realm.

Note that the *bstr_realm_unset_db* command is executed automatically by the *bstr_realm_unset_node* command.

If the *-s* option is specified, the *bstr_realm_unset_db* deletes all the object definitions for *realm_name* but not the snapshot configurations.

Options

-s

If you specify this option, the *bstr_realm_unset_db* deletes all the object definitions for *realm_name*, but leaves the snapshot configurations.

See Also

bstr_realm_setup_db

bstr_run

The *bstr_run* procedure is the BASEstar Open Running Procedure (BRP). It allows you to activate a Realm without having to set up or start a Node.

syntax

bstr_run *realm_name* **[–I] [–L** *location***] [–NOPODB] [–NOVODB] [–RDB** *db_version***]**

Parameters

realm_name

The name of the Realm that you want *bstr_run* to activate. BASEstar Open creates the Realm with the following default characteristics:

- Single Node
- One Domain (root "/")
- One AMS Actor called *realm_name*_SERVERS
- Two Activities, allowing use of the VOLATILE database (VODB), as follows:
  - EVM, (event manager) named /*realm_name*_SERVERS/EVM, which runs the Program /*realm_name*_EVM_SERVER
  - DTM, (data manager) named /*realm_name*_SERVERS/DTM, which runs the program /*realm_name*_DTM_SERVER.

If RDB is available, your system can support PERMANENT database (PODB) and DB server.

You can use the appropriate parameters and options to override *bstr_run* procedure defaults.

Options

–I

Enables the interactive execution of the *bstr_run* procedure. If this flag is specified, all the BASEstar Open environment procedures are executed interactively, and the *bstr_run* default values can be overridden.

The BASEstar Open environment procedures contain questions during their execution (for example, requiring confirmation, parameters, values, and so on). By default, the *bstr_run* procedure executes these procedures automatically, suppressing any questions and using internal default values.

–L *location*

The pointer to the directory specification containing the BASEstar Open kit or where BASEstar Open was installed. This directory specification must be provided only if the Node is not currently set up.

–NOPODB

Causes *bstr_run* to skip the PERMANENT database setup-start phase. As a result, you can only configure objects in the VOLATILE database.

By default, *bstr_run* sets up the Realm environment in order to create the PODB and all structures required to use the PERMANENT database. Furthermore, it will also start the database server to allow immediate access to PODB facilities.

The **–NOPODB** flag inhibits the execution of this phase, thereby preventing access to the PODB. However, even if the **–NOPODB** flag is not specified, the *bstr_run* procedure can determine whether the Node is elegible to be a "PODB Node" (that is, if a supported DBMS is installed). If this is notthe case, it skips the PODB setup-start phase automatically.

–NOVODB

Causes *bstr_run* to skip the VOLATILE database setup-start phase.

By default, *bstr_run* sets up an AMS Actor containing two Activities related to EVM and DTM servers, where:

- The name of the Actor is *realm_name*_SERVERS.
- The full name of the Activity related to EVM server is "/*realm_name*_SERVERS/EVM" and the program executed by it is "/*realm_name*_EVM_SERVER".
- The full name of the Activity related to DTM server is "/*realm_name*_SERVERS/DTM and the program executed by it is "/*realm_name*_EVM_SERVER".
- The Node is the current one and the Domain used is the root (/). This object is configured by using a CLI file (script) created by *bstr_run* called SYS$SCRATCH:AMS_SETUP_FILE.CLI. This file can also be used as an example of a simple AMS configuration.

If you only specify the **–NOVODB** flag, BASEstar Open skips this phase; you must therefore create any AMS objects separately if you want to access the VODB. However, even if you do not specify the **–NOVODB** flag, BASEstar Open only executes this phase if the Realm has been started by *bstr_run*. This is to avoid interference with any user configurations.

–RDB *db_version*

Specifies that the RDB version *db_version* is to be used to set the PERMANENT database facilities. The RDB version must be provided only if more than one RDB version is present on the system.

This parameter has no significance if the **–NOPODB** flag is set.

Description

The *bstr_run* procedure analyses at the user's environment and performs all the operations needed to run BASEstar Open: using *bstr_run*, the user no longer needs to call all the DCL command procedures needed to set up and start BASEstar Open. For example, if the Node has been already set up and started, *bstr_run* starts its activity from the execution of the *bstr_realm_setup_node* procedure.

The *bstr_run* procedure runs the following BASEstar Open procedures to create a minimum BASEstar Open environment:

1. *bstr_node_setup*[1]

[1] Requires all privileges

2. *bstr_node_start*[1]
3. *bstr_realm_setup_node*
4. *bstr_realm_start_node*[2]
5. *bstr_realm_setup_db*
6. *bstr_realm_start_db*[2]

The account from which you run the *bstr_run* procedure must have the requisite privileges for each of the constituent environment procedures. For more information on the privileges required to run the environment management procedures, refer to Section 10.2.

The *bstr_run* procedure performs a ''rollback'' mechanism if any of the operations fail during execution. The rollback mechanism involves only the operations performed previously by *bstr_run*; it has no effect on operations already performed before *bstr_run* commenced execution. For example, if you run *bstr_run* on a Node that has already been set up and started, the rollback only affects any subsequent steps, and leaves the Node in the ''set and started'' state.

In specific circumstances, *bstr_run* sets up and starts the PODB and an Actor containing a couple of Activities. This enables you to work with Event and Data Services on both PERMANENT and VOLATILE databases.

During execution, the *bstr_run* procedure displays only the basic messages relevant to its execution. Detailed information of each step of the execution is instead collected the log file SYS$SCRATCH:BRP_LOG_FILE.OUT.

Examples

1. $ **bstr_run SAMPLE**

   In this example, the *bstr_run* procedure performs the steps required to set up the realm SAMPLE, and then performs the ''rollback'' mechanism on failure:

   a. Sets up the Node

   b. Executes the *bstr_run* procedure for the Realm SAMPLE. The *bstr_run* procedure:

      - Starts the Node
      - Sets up the Realm SAMPLE
      - Starts the Realm SAMPLE

      At this point, the start operation on Realm SAMPLE fails (either partially or completely). The *bstr_run* PROCEDURE continues as follows with the ''rollback'':

      - Shuts Realm SAMPLE
      - Unsets Realm SAMPLE
      - Shuts Node

   c. The *bstr_run* procedure terminates.

[2] Requires DETACH privilege

11

BASEstar Open Environment Monitor

11.1 Introduction

The BASEstar Open Environment Monitor is an interactive tool capable of displaying the Realms and Nodes that have been started up, together with their relationships in the BASEstar Open operating environment. It also displays the Domain server resources that are available within a Realm. Domain Services servers can be data servers, device servers, or event servers.

The *bstr_mon* tool is used to monitor the active components of the BASEstar Open environment; for example, started Nodes, Realms that are started on these Nodes and the services available for a given Realm. The *bstr_mon* tool can be used to display the following:

- The available Realms
- For each Realm, the Nodes on which the Realm is available, the available Domains and the VMDs
- A list of the available Nodes and a list of Realms available on each Node.

You can run the BASEstar Open Environment Monitor from any VTxxx terminal. The Monitor is user-friendly, with simple command keystrokes and consistent screen menus. It displays Realms (with associated services) and Nodes on two screens. The two views are:

- Active Realm View - displays the active Realms. If you select a Realm, it displays the Nodes which made the Realm available to it, together with the Realm's associated services.
- Active Node View - displays the active Nodes. If you select a Node, it displays the Realms made available for use on the Node.

When the BASEstar Open Environment Monitor command is invoked, the default view is displayed (see Figure 11–1). The default view is the Active Realm View.

11.2 Running the BASEstar Open Environment Monitor

To run the BASEstar Open Environment Monitor perform the following steps:

1. Ensure you have the quotas and privileges of a Realm user (see Section 10.1.1.1 for details).
2. Check that the BASEstar Open environment has been started up successfully on the Node that is to run the Monitor (see Chapter 9 for details).
3. Make sure that the installation-dependent global variables have been initialized for the login user. To initialize the installation dependent global variables, execute the *bstrusers* script. See Appendix G for further information about global variables.
4. Enter the following from the command interpreter prompt:

   ```
   $ bstr_mon
   ```

   to display the view of the active Realms shown in Figure 11–1.

11.3 General Characteristics

The general characteristics of the BASEstar Open Environment Monitor are described below.

Refreshing Displayed Information

The BASEstar Open Environment Monitor application retains information in its memory and you must therefore use the Update command to review the current environment. The screen is refreshed automatically every five minutes to display the current environment. The Update command can be executed in both views (mnemonic "U").

Quitting the Environment Monitor

To exit from either view of the BASEstar Open Environment Monitor, execute the Quit Command (mnemonic "Q").

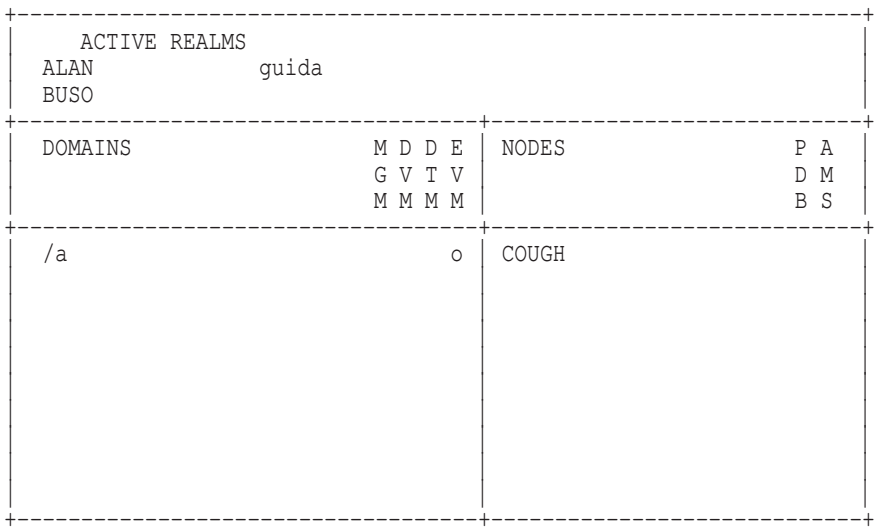
11.4 Views

The sections below explain how to perform operations on the BASEstar Open Environment Monitor Views.

11.4.1 Active Realm View

The default view that is displayed when the BASEstar Open Environment Monitor is invoked is shown in Figure 11–1. The screen is divided into three windows.

Figure 11–1 Active Realm View



The Active Realm window (upper window) displays a list of available Realms, sorted by name. If there are no names available, the screen displays ‘‘No active realms’’. The cursor is initially positioned in this window and the first Realm is highlighted. The cursor can be moved through the list of Realms by using the up and down cursor movement keys.

When the cursor is positioned on a Realm, the list of resources (Domains and VMDs) for this Realm is displayed in the bottom left-hand window (the Domain window) and all the Nodes on which the Realm has been started up are displayed in the bottom right-hand window (the Node window).

The Domain window displays the list of resources available in the Realm, sorted by name. If there are no resources available, the screen displays ‘‘No active domains’’. Each resource is displayed on a separate line. To the right of each line, a bullet is displayed indicating the type of server activity that makes the resource available, as follows:

Identifier	Server Activity
EVM	Event Services server
MGM	Packet Services server
DTM	Data Services server
DVM	Device Services server

The following also applies:

- If the bullet is positioned under EVM, MGM or DTM, the string to its left contains the full name of a Domain.
- If the bullet is positioned under DVM, the string to its left contains the name of a VMD.

The Node window displays a list of the Nodes that have been started up for the selected Realm. If no Nodes have been started up, the screen displays ‘‘No active nodes’’. This window displays each Node on a separate line. Each line contains the following information:

- The name of a Node
- A bullet positioned under the PDB column, if the Database Services server is active on that Node
- A bullet positioned under the AMS column, if the Application Management Services server is active on that Node.

If it is not possible to view all of the information in the Domain window or Node window, you can select these windows by pressing the carriage return (ENTER) key (press the carriage return key again to select the Node window, and repeat this action to select the Active Realm window). You can then move the cursor through the list of Domains or Nodes using the up and down cursor movement keys.

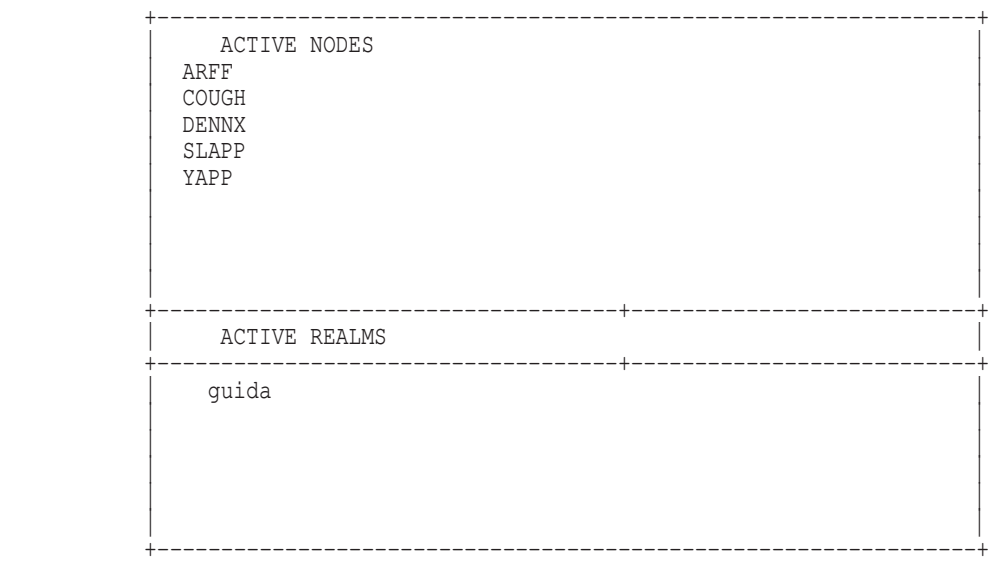
The command line at the bottom of the screen displays the available options. In the case of the Active Realm View, menu options include jumping to the Active Node View, Update, quitting the BASEstar Open Environment Monitor, or switching between windows, as follows:

Option	Command	Function
[ENTER]-toggle	Carriage return [ENTER] key	To move between windows
Update	U	To refresh screen to display current BASEstar Open Environment
Nodes	N	To invoke Node View
Quit	Q	To exit BASEstar Open Environment Monitor

11.4.2 Active Node View

The screen is divided into two windows (see Figure 11–2).

Figure 11–2 Active Node View



The example in Figure 11–2 displays a list of the active Nodes in the BASEstar Open environment. The cursor is positioned on the Node "arff" and the bottom window therefore displays a list of the Realms that have been made available on that Node.

The Active Node window (top window) displays a list of Nodes, sorted by name. If there are no Nodes, the screen displays ‘‘No active nodes’’. If it is not possible to view all of the Nodes together, you can move the cursor through the list of Nodes using the up and down cursor movement keys. When you position the cursor on a Node, it is highlighted and automatically becomes the selected Node.

The Active Realm window (bottom window) displays the list of Realms (sorted by name) that have been started up for the selected Node. If there are no Realms, the screen displays ‘‘No active realms’’. If it is not possible to view all of the Realms together, you can select this window by pressing the carriage return (ENTER) key and then move the cursor through the list of Realms using the up and down cursor movement keys.

The command line at the bottom of the screen displays the available options. In the case of the Active Node View, menu options include jumping to the Active Realm View, Update, quitting the BASEstar Open Environment Monitor, or switching between windows, as follows:

Option	Command	Function
[ENTER]-toggle	Carriage return [ENTER] key	To move between windows
Update	U	To refresh screen to display current BASEstar Open Environment
Realms	R	To invoke Active Realm View

Option	Command	Function
Quit	Q	To exit BASEstar Open Environment Monitor

12

BASEstar Open-Supplied Servers

A BASEstar Open-supplied server is a system process that makes a set of BASEstar Open objects available at run-time. For each Realm, the following BASEstar Open servers make VODB objects available:

- Event Services servers, which make the Event Services objects of one or more Domains available at run-time.
- Packet Services servers, which make the Packet Services objects of one or more Domains available at run-time.
- Data & Device Services servers, which make the following objects available at run-time:
  - Data Services objects of one or more Domains
  - One or more VMDs and associated Device Services server objects.
- Global Object Services servers make the Realm global objects, such as Domains and VMDs, available at run-time.
- The Application Management Services server, which makes the Application Management Services objects available at run-time and therefore provides the main functions for the activation, monitoring and control of the Realm application components (user-written applications, Data Services servers, Event Services servers, and Device Services servers).

A Realm can contain additional optional servers that perform some specialized tasks for the Realm, such as:

- A Database Services server, which provides access to the PODB object definitions.
- A PC Communication server, which enables applications running on client Nodes to perform operations on Realm objects and object definitions like other BASEstar Open applications running on server Nodes.

The rest of this chapter describes the characteristics of the different types of servers, and explains how to manage them.

12.1 Servers, Activities and Owned Resources

The Event Services servers, Packet Services servers, Data Services servers and Device Services servers are implemented as Application Management Services applications. This means that, at run-time, a server acts as a Program execution capable of executing several Activities. Depending on the kind of server, each Activity *owns* and thus makes available a specific set of resources (Domain) to BASEstar Open users.

BASEstar Open servers (apart from Application Management Services servers), make VODB objects available once they have been started up. A BASEstar Open server is started up using the command interpreter features or the Application Management Services features described in the *BASEstar Open Command Language Interface* manual. The Realm Application Management Services servers are activated automatically as part of the Realm startup process (see Chapter 9).

In order to activate a server using Application Management Services, you must configure it as a set of Activity objects associated with an appropriate Program object.

12.1.1 Snapshots

When a server Activity is started up, it looks for the definitions of the default objects in a specific snapshot configuration. (Snapshots are created using CLI configuration commands.) The use of snapshots speeds up the operation that each Activity performs when loading default objects in the VODB. Of course, if the specified snapshot is not found, no default objects are created.

For example, when a Data Services server Activity is started up, it makes objects such as Data_Points and Triggers available, whose definitions are contained in the snapshot for the associated Domain. At run-time, the same Activity allows for additional objects that are visible in that Domain to be created in the VODB.

12.1.2 Logged Information

Like many other BASEstar Open components, BASEstar Open servers write information to a log file. To find out how this information is logged, see Chapter 14.

12.2 Global Object Services

A Global Object Services server is the Realm-specific environment component that makes the global objects available on that Realm. It enables users to request services such as creating and getting attributes of global objects.

The objects made available by Global Object Services belong to the following classes:

- Datatype
- Domains
- VMD
- Protocol_Profile
- VMD_Extension_Parameter

A Global Object Services server is started up on each Node where the Realm is started. Once started up, a Global Object Services server checks to see if another Global Object Services server has been started up on another Node for the same Realm, and then:

- If it is the first to start up, looks for its snapshot in the most recently generated snapshot configuration and then creates its default objects in the VODB.
- If it is not the first to start up, it does not load the configuration because it makes available objects that have been already loaded by the other Global Object Services servers in the Realm from the snapshot or that have been already created when fulfilling user service requests.

To provide such dynamic and distributed services, the Global Object Services servers of a Realm continuously notify each other of their respective availability and of operations in progress. Once you have started up the Realm, you can use the LOAD CONFIGURATION command to force the active Global Object Services servers to load an alternative snapshot configuration.

A Global Object Services server supports up to 40 client application components. This limit is determined by the value of the BSTR_GOM_MAX_CLIENT variable. To increase the number of clients, follow these steps:

1. Locate the following section of code in the `$BSTR_ETC/realm_startup/start05_gom.sh` file:

```
#
#        Define default GOM trace utility
#
BSTR_GOM_TRACE=GOM; export BSTR_GOM_TRACE
```

2. To set a maximum of 100 clients, enter the following code immediately below the above section:

```
#
# Define new GOM clients maximum limit
#
# BSTR_GOM_MAX_CLIENT=100; export BSTR_GOM_MAX_CLIENT
```

12.3 Application Management Services

The Application Management Services servers for a given Realm make the Application Management Services objects of that Realm available to clients. An Application Management Services server is started up on each Node where the Realm is started.

The objects made available by Application Management Services belong to the following classes:

- Activity
- Actor
- Node
- Program

Once started up, an Application Management Services server looks for a snapshot in the most recently generated snapshot configuration and then creates its default objects in the VODB. Once you have started up the Realm, you can use the LOAD CONFIGURATION command to request the active Application Management Services servers to load an alternative snapshot configuration.

12.3.1 Logical Names - Inheritance and Usage

The Application Management Services server has the same UIC of the *Realm user*. This means that it sees also the logical names of the group to which the *Realm user* belongs.

As the Application Management Services server has the same privileges of the *Realm user*, you must remember to assign the *Realm user* all the privileges you want to be assigned to the Program executions the Application Management Services server will run.

New logical names (or modifications to existing logical names), can be set in the Node environment file and/or parameter file of each Program. (For information regarding the record format of such files, refer to Appendix G.)

The values set to these logical names are made available and can be used in different contexts. Table 12–1 specifies when a user-defined logical name can be set, and the context in which it can be used.

Table 12–1 Logical Names - Validity and Usage

	Logical Name Usage		
If Set...	**OBJS_PATH**	**PARS_PATH**	**RET_VAR**
In the LNM$GROUP table	Yes	Yes	Yes
In the environment Node file	Yes Cannot be used to express the Node environment file pathname	No	Yes
In the Program parameter file	No	No	Yes

Logical Name Usage

OBJS_PATH — Used to express the pathnames in the following Node and Program attributes: Node environment file, Program parameter file and Program image file.
PARS_PATH — Used to form the pathnames specified in the following Program parameter file items: input, output and error.
RET_VAR — Available to a Program execution that can retrieve their value using the getenv( ) run-time C Language function.

12.3.1.1 Application Management Services Server Privileges

The Application Management Services server can create processes having the following privileges: GRPNAM, DETACH, GROUP, WORLD, TMPMBX, NETMBX, SYSPRV and SHARE.

If the Application Management Services server must have additional privileges other than those listed above, the *Realm user* must also have the SETPRV privilege.

The additional privileges can be specified to the Application Management Services server in the following ways, depending on whether or not you set the value of the *flag* item to NOUAF in the Program parameter file:

- The *flag* item is set to NOUAF.
  The additional privileges can be specified using the *prv* item of same Program parameter file.
- The *flag* item has not been specified.
  The additional privileges are all the DEAFULT PRIVILEGES defined in UAF for the *Realm user*, plus any additional privilege set in the *login.com* associated with the *Realm user*, using the SET PROCESS/PRIVILEGES DCL command.

12.4 Event Services

One Event Services server can manage a maximum of 32 Activity executions. Each Activity execution makes the Event Services objects visible in the associated Domain available to users.

The objects made available by Event Services belong to the following classes:

- Enbox
- Event
- Event_Set

Once started up, an Activity execution looks for the corresponding snapshot to create the default objects in the VODB. If no snapshot is found, no default VODB objects are created for the associated Domain.

The executable name for the Program is `evm_server`. This executable is stored in the $BSTR_BIN directory (see Appendix G).

12.4.1 Using Application Management Services to Manage an Event Services Server

An Event Services server must be configured as a Program, with a number of associated Activities that is equal to the number of Domains that it has to make available. Application Management Services start up the server when an execute message is sent to any of its Activities.

For further information about how to configure and manage an Event Services server using Application Management Services, refer to the *BASEstar Open Command Language Interface*.

12.4.2 Starting Up and Shutting Down an Event Services Server from the Command Interpreter

To start up an Event Services server from the command interpreter, run its executable image from the command interpreter prompt after having set the appropriate global variables. Remember that, in this case, an Event Services server can only execute one Activity; that is, it can only make one Domain available.

Before running the server, you must perform the following actions:

- Set the name of the Activity in the BSTR_ACTIVITY global variable. This name bears no relation to any existing Application Management Services Activity object, and is simply a label used to identify messages in the LOG file.
- Set the BSTR_DOMAIN global variable with the full name of the Domain that the server Activity is to make available. This name is also used to identify any corresponding snapshot.
- Initialize the installation-dependent global variables by executing the *bstrusers* command file (see Appendix G).
- Set the BSTR_REALM global variable with the name of the Realm on which the server is to run.
- Set the BSTR_DBVERSION and BSTR_DBACCESS_KEY global variables (optional).

For further information about global variables and how to set them, refer to Appendix G.

The example below shows how to start the Event Services serverfrom DCL. The server makes the /PLANT/PRODUCTION/LINE2 Domain available. The server creates the Event Services default objects whose definitions are contained in the snapshot configuration identified by a major identifier of 2.

```
# @DISK3:[BSTROSVA032.ETC]bstrusers
# DEFINE BSTR_REALM MY_Realm
# DEFINE BSTR_ACTIVITY "DATA_ACT_PRODUCTION_LINE2"
# DEFINE BSTR_DOMAIN "/PLANT/PRODUCTION/LINE2"
# DEFINE BSTR_DBVERSION 2
# SPAWN/NOWAIT RUN BSTR_BIN:evm_server
```

In the example, note that the server has been started as a background process; this is advisable, but not mandatory.

To stop a Event Services server use the STOP/ID="*PID*" OpenVMS/DCL command. Stopping a server using the STOP command causes the server to exit without performing the cleanup phase (some files may be left in the BASEstar Open working directory).

12.5 Packet Services

One Packet Services server can manage a maximum of 32 Activity executions. Each Activity execution makes the Packet Services objects visible in the associated Domain available to users.

The objects made available by Packet Services belong to the following classes:

- Packet
- Port

Once started up, an Activity execution looks for the corresponding snapshot to create the default objects in the VODB. If the snapshot is not found, no default VODB objects are created for the associated Domain.

The executable name for the Program is `mgm_server`. This executable is stored in the $BSTR_BIN directory (see Appendix G).

12.5.1 Using Application Management Services to Manage a Packet Services Server

A Packet Services server must be configured as a Program, with a number of associated Activities that corresponds to the number of Domains to be made available. Application Management Services start up the server when an execute message is sent to any of its Activities.

For further information about how to configure and manage a Packet Services server using the Application Management Services, refer to the *BASEstar Open Command Language Interface*.

12.5.2 Starting Up and Shutting Down a Packet Services Server from the Command Interpreter

To start up a Packet Services server from the command interpreter, run its executable image from the command interpreter prompt after having set the appropriate global variables. Remember that, in this case, a Packet Services server can only execute one Activity: that is, it can only make one Domain available.

Before running the server, you must perform the following operations:

- Set the name of the Activity in the BSTR_ACTIVITY global variable. This name bears no relation to any existing Application Management Services Activity object, and is simply a label used to identify messages in the LOG file.
- Set the BSTR_DOMAIN global variable with the full name of the Domain that the server Activity is to make available. This name is also used to identify any corresponding snapshot.
- Initialize the installation-dependent global variables by executing the *bstrusers* command file (see Appendix G).
- Set the BSTR_REALM global variable with the name of the Realm on which the server is to run.
- Set the BSTR_DBVERSION and BSTR_DBACCESS_KEY global variables (optional).

For further information about global variables and how to set them, refer to Appendix G.

The following example shows how to start the Packet Services server from DCL. The server makes the /PLANT/PRODUCTION/LINE2 Domain available. The server creates the Event Services default objects whose definitions are contained in the snapshot configuration identified by a major identifier equal to 2.

```
# @DISK3:[BSTROSVA032.ETC]bstrusers
# DEFINE BSTR_REALM MY_Realm
# DEFINE BSTR_ACTIVITY "PACKET_ACT_PRODUCTION_LINE1"
# DEFINE BSTR_DOMAIN "/PLANT/PRODUCTION/LINE2"
# DEFINE BSTR_DBVERSION 2
# SPAWN/NOWAIT RUN BSTR_BIN:mgm_server
```

In the example, note that the server has been started as a background process; this is advisable, but not mandatory.

To stop a Packet Services server use the STOP/ID="*PID*" OpenVMS/DCL command. Stopping a server using the STOP command causes the server to exit without performing the cleanup phase (some files may be left in the BASEstar Open working directory).

12.6 Data Services

A Data Services server can manage a maximum of 32 Activity executions. Each Activity execution makes the Data Services server objects visible in the associated Domain available to users. At startup, an Activity execution looks for the corresponding snapshot to create the default objects in the VODB.

The executable name for the Program is `ddm_server`. This executable is placed in the $BSTR_BIN directory, located in the BASEstar Open installation directory (see Appendix G).

12.6.1 Environmental Variables

This section describes the Data Services server environmental variables.

For a full summary of the Data Services server environmental variables, refer to Table 12–2.

Table 12–2 Data Management—Environmental Variables

Name	Description	Default	Range
BSTR_ACTIVITY	Activity name (starting without Application Management Services)	None	Name of Activity
BSTR_DADE_STACT_TH_SIZE	Activity control thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_BVMS_TH_SIZE	BASEstar VMS thread stack size	196608 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_CLIENT_TH_SIZE	Client servicing thread stack size	262144 bytes	21504 bytes –> MAXINT[1] bytes
BSTR_DATA_CONNH_TH_SIZE	Connection handling thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_FCONN_TH_SIZE	DATA <-> DEVICE comm thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_I_SIZE	DATA management initial memory pool	250 Kbytes	250 –> MAXINT[1] Kbytes
BSTR_DOMAIN	Domain name (starting without Application Management Services)	None	Name of Domain
BSTR_REALM	Realm to use	None	Name of Realm
BSTR_SERVERS_PFACTOR	Parallel/Sequential execution of client connect/disconnect and Activity execute/terminate operations	1	1 –> 32

[1]Maximum positive value in a signed 32-bit integer, expressed in decimal format, no mantissa & exp notation.

12.6.2 Using Application Management Services to Manage a Data Services Server

A Data Services server is configured as a Program, with a number of associated Activities that is equal to the number of Domains that it needs to make available. The Application Management Services start up the server when an execute message is sent to any of its Activities.

For further information about how to configure and manage a Data Services server using the Application Management Services, refer to the *BASEstar Open Command Language Interface*. If you want to set one or more of the Data Services-specific global variables listed in Section 12.6.1, you must set them in the parameter file of the associated Program object.

The example that follows shows a valid parameter file for a Data Services server. Refer to Appendix H for details of the general structure of parameter files. However, remember that for a Data Services server only the global variables listed in Section 12.6.1 must be specified.

```
/PROG_DATA_SERVER.env: BSTR_DATA_I_SIZE=1000
```

12.6.3 Starting Up and Shutting Down a Data Services Server from the Command Interpreter

To start up a Data Services server from the command interpreter, run its executable image from the command interpreter prompt after having set the appropriate global variables. Remember that a Data Services server can only execute one Activity; that is, it can make only one Domain available.

Before running the server, you must perform the following operations:

- Set the name of the Activity in the BSTR_ACTIVITY global variable. This name bears no relation to any existing Application Management Services Activity object, but is simply a label used to identify messages in the LOG file.
- Set the BSTR_DOMAIN global variable with the full name of the Domain that the server Activity is to make available. This name is also used to identify any corresponding snapshot.
- Initialize the installation-dependent global variables by executing the *bstrusers* command file (see Appendix G).
- Set the BSTR_REALM global variable with the name of the Realm on which the server is to run.
- Set the BSTR_DBVERSION and BSTR_DBACCESS_KEY global variables (optional).

For further information about global variables and how to set them, refer to Appendix G.

The example below shows how to start the Data Services server from DCL. The server makes the /PLANT/PRODUCTION/LINE2 Domain available. The server creates the Data Services server default objects whose definitions are contained in the snapshot configuration identified by a major identifier of 2.

```
# DISK3:[BSTROSVA032.ETC]bstrusers
# DEFINE BSTR_REALM MY_Realm
# DEFINE BSTR_ACTIVITY "DATA_ACT_PRODUCTION_LINE1"
# DEFINE BSTR_DOMAIN "/PLANT/PRODUCTION/LINE2"
# DEFINE BSTR_DBVERSION 2
# SPAWN/NOWAIT RUN BSTR_BIN:ddm_server
```

Note that in this example the server has been started as a background process. This is advisable, but not mandatory.

To stop an Data Services server use the STOP/ID="*PID*" OpenVMS DCL command. Stopping a server with the STOP command causes it to exit without performing the cleanup phase (some files may be left in the BASEstar Open working directory).

12.7 Device Services

One Device Services server can manage a maximum of 32 Activities, each capable of making a VMD and all its associated Device Services objects available. At run-time, each Activity looks for its VMD snapshot in the current snapshot configuration (specified during execution of the LOAD CONFIGURATION command).

The objects made available by Device Services belong to the following classes:

- Named_Variable
- Polling_Set
- Protocol_Profile
- Unnamed_Variable
- VMD
- VMD_Extension_Parameter

The executable name for the Program is `ddm_server`. This executable is placed in the $BSTR_BIN directory, under the BASEstar Open installation directory (see Appendix G).

12.7.1 Global Variables

This section describes the global variables for a Device Services server.

12.7.1.1 Specifying the Calling VMD

Some protocols require you to specify the addresses of both the calling and the called partner VMDs, so that the Device Services server can establish an association with the plant device. Before starting up the Device Services server from the command interpreter or the Application Management Services, specify the name of the calling VMD in the BSTR_LOCAL_VMD_NAME global variable.

12.7.1.2 Specifying the Connection Retry Timeout to a VMD

The value assigned to the BSTR_RETRY_CONN_TIME variable specifies how many seconds the Device Services server waits before trying to restart the association with the VMD after a failure. The default value is 30 seconds.

For a full summary of the Device Services server environmental variables, refer to Table 12–3.

Table 12–3 Device Management—Environmental Variables

Name	Description	Default	Range
BSTR_ACTIVITY	Activity name (starting without Application Management Services)	None	Name of Activity
BSTR_DADE_STACT_TH_SIZE	Activity control thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_BVMS_TH_SIZE	BASEstar VMS thread stack size	196608 bytes	21504 –> MAXINT[1] bytes

[1]Maximum positive value in a signed 32-bit integer, expressed in decimal format, no mantissa & exp notation.

(continued on next page)

Table 12–3 (Cont.) Device Management—Environmental Variables

Name	Description	Default	Range
BSTR_DEVICE_GEN_TH_SIZE	DEVICE general thread stack size	131072 bytes	21504 –> MAXINT[1] bytes
BSTR_DEVICE_I_SIZE	DEVICE management initial memory pool	50 Kbytes	50 –> MAXINT[1] Kbytes
BSTR_LOCAL_VMD_NAME	Local VMD name	None	Name of Local VMD
BSTR_POLLING_FACTOR	Time scale factor for Polling Sets	1	1 –> 100
BSTR_REALM	Realm to use	None	Name of Realm
BSTR_RETRY_CONN_TIME	Connection retry time for VMDs	30 seconds	1 –> MAXINT[1] seconds
BSTR_SERVERS_PFACTOR	Parallel/Sequential execution of client connect/disconnect and Activity execute/terminate operations	1	1 –> 32
BSTR_VMD	VMD name (starting without Application Management Services)	None	Name of VMD

[1]Maximum positive value in a signed 32-bit integer, expressed in decimal format, no mantissa & exp notation.

12.7.2 Using Application Management Services to Manage a Device Services Server

Configure a Device Services server as a Program, with a number of associated Activities that is equal to the number of VMDs that it has to make available. Application Management Services start up the server when an execute message is sent to any of its Activities. For further information about how to configure and manage a Device Services server using the Application Management Services, refer to the *BASEstar Open Command Language Interface*.

If you want to set one or more of the Device Services global variables listed in Section 12.7.1, you must set them in the parameter file of the associated Program object. The example that follows shows a valid parameter file for a Device Services server. Refer to Appendix H for details of the general structure of parameter files. However, remember that for a Data Services server only the global variables listed in Section 12.7.1 must be specified.

```
/PROG_DEVICE_SERVER_1.env: BSTR_LOCAL_VMD_NAME=DV_CALLING_VMD_1
/PROG_DEVICE_SERVER_1.env: BSTR_RETRY_CONN_TIME=15
```

12.7.3 Starting Up and Shutting Down a Device Services Server from the Command Interpreter

To start up a Device Services server from the command interpreter, run its executable image from the prompt after having set the appropriate global variables.

Before running the server, you must perform the following actions:

- Set the names of the Activities in the BSTR_ACTIVITY global variable. These names bear no relation to any existing Application Management Services Activity objects, but are simply labels used to identify messages in the LOG file. Names must be separated by blanks.

- Set the BSTR_VMD global variable with the name of the VMD that the server Activity is to make available. These names are also used to identify the corresponding snapshot.
- Initialize the installation-dependent global variables by executing the *bstrusers* command file (see Appendix G).
- Set the BSTR_REALM global variable with the name of the Realm on which the server is to run.
- Set the BSTR_DBVERSION and BSTR_DBACCESS_KEY global variables (optional).

For further information about global variables and how to set them, refer to the Appendix G.

The example that follows shows how to start a Device Services server from DCL. The server makes the PLC1 and PLC2 VMDs available. The server creates the default object definitions in the snapshot configuration identified by a major identifier equal to 2. (BASEstar Open has been installed under the DISK3:[BSTROSVA032.ETC] directory.)

```
# @DISK3:[BSTR0320.ETC]bstrusers
# DEFINE BSTR_REALM MY_Realm
# DEFINE BSTR_ACTIVITY DVM_SERVER_PLC1
# DEFINE BSTR_VMD PLC1
# DEFINE BSTR_DBVERSION 2
# SPAWN/NOWAIT RUN BSTR_BIN:ddm_server
```

Note that in this example the server has been started as a background process. This is advisable, but not mandatory.

To stop a Device Services server, use the OpenVMS DCL command STOP /ID="*PID*". Stopping a server with the STOP command causes it to exit without performing the cleanup phase (some files may be left in the BASEstar Open working directory).

12.8 Combined Data & Device Services

The BASEstar Open Data & Device Services combine under a single server the functionality provided by the individual Data Services and Device Services components. This improves performance by eliminating the need for inter-process communication between the Data Manager (DTM) and the Device Manager (DVM).

12.8.1 Environmental Variables

Table 12–4 contains a full summary of the Data & Device Services server environmental variables.

Table 12–4 Data and Device Management—Environmental Variables

Name	Description	Default	Range
BSTR_ACTIVITY	Activity name (starting without Application Management Services)	None	Name of Activity
BSTR_CONN_MODE	Local VMD connection behavior	ACTIVE	ACTIVE, PASSIVE
BSTR_DADE_STACT_TH_SIZE	Activity control thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_BVMS_TH_SIZE	BASEstar VMS thread stack size	196608 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_CLIENT_TH_SIZE	Client servicing thread stack size	98304 bytes	21504 bytes –> MAXINT[1] bytes
BSTR_DATA_CONNH_TH_SIZE	Connection handling thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_FCONN_TH_SIZE	DATA <-> DEVICE comm thread stack size	49152 bytes	21504 –> MAXINT[1] bytes
BSTR_DATA_I_SIZE	DATA management initial memory pool	250 Kbytes	250 –> MAXINT[1] Kbytes
BSTR_DEVICE_GEN_TH_SIZE	DEVICE general thread stack size	131072 bytes	21504 –> MAXINT[1] bytes
BSTR_DEVICE_I_SIZE	DEVICE management initial memory pool	50 Kbytes	50 –> MAXINT[1] Kbytes
BSTR_DOMAIN	Domain name (starting without Application Management Services)	None	Name of Domain
BSTR_LOCAL_VMD_NAME	Local VMD name	None	Name of Local VMD
BSTR_POLLING_FACTOR	Time scale factor for Polling Sets	1	1 –> 100
BSTR_REALM	Realm to use	None	Name of Realm
BSTR_RETRY_CONN_TIME	Connection retry time for VMDs	30 seconds	1 –> MAXINT[1] seconds

[1]Maximum positive value in a signed 32-bit integer, expressed in decimal format, no mantissa & exp notation.

(continued on next page)

Table 12–4 (Cont.) Data and Device Management—Environmental Variables

Name	Description	Default	Range
BSTR_SERVERS_PFACTOR	Parallel/Sequential execution of client connect/disconnect and Activity execute/terminate operations	1	1 –> 32
BSTR_VMD	VMD name (starting without Application Management Services)	None	Name of VMD

12.8.2 Using the Data & Device Services to Support Passive Connections

As an alternative to establishing a connection to a physical device, you can instruct the Data & Device Services server to wait passively for connection requests from the device. To configure the Data & Device Services server to do this, follow these steps:

1. Define the BSTR_LOCAL_VMD_NAME environment variable as the local VMD name.
2. Define the BSTR_CONN_MODE environment variable as "PASSIVE"[1].

The example below establishes a passive connection via the local VMD name DATADEV_CALLING_VMD_1.

```
/PROG_DEVICE_SERVER_1.env: BSTR_LOCAL_VMD_NAME=DATADEV_CALLING_VMD_1
/PROG_DEVICE_SERVER_1.env: BSTR_CONN_NODE=PASSIVE
```

[1] If you do not define BSTR_CONN_MODE variable, or if you define it as "ACTIVE", the Data & Device Services server establishes the connection actively

12.9 PC Communication Servers

Using a PC Communication server, it is possible for BASEstar Open applications running on client Nodes to manage objects of a Realm that has been started up on the server Nodes of a distributed BASEstar Open environment.

A PC Communication server accepts TCP/IP service requests from BASEstar Open MS Windows client Nodes and dispatches them to the appropriate Realm servers. BASEstar Open starts a PC Communication server on all the Nodes where the Realm is started. It is important that you remember this so that you can balance the load of each PC Communication server (and, therefore, of each Node) by configuring the client Nodes appropriately. Refer to the *BASEstar Open Client Inst. & Management Guide* for details of how to install, configure and manage BASEstar Open on client PCs.

13

Application Management Services Monitor

13.1 Introduction

The Application Management Services Monitor is capable of detecting and displaying changes in the attributes of the Application Management Services objects of a given Realm as they occur. Attribute changes are generally caused by the following events:

- Messages sent to the objects in question from users that issue CLI commands, or applications that invoke API procedures; for example, an Actor executed via the ACTOR EXECUTE CLI command.
- Asynchronous occurrences that take place within the system resources; for instance, an Activity execution that terminates autonomously, or a system process in which a Program execution is run, that terminates abnormally.

You can run the Application Management Services Monitor from any VTxxx terminal; the information displayed always reflects the real time situation. The Monitor is user-friendly, with simple command keystrokes and consistent screen menus.

13.2 Running Application Management Services Monitor

To run the Application Management Services Monitor, perform the following steps:

1. Ensure you have the quotas and privileges of a *Realm user* (see Section 10.1.1.1 for details).
2. Check that the BASEstar Open environment has been started up successfully on the Node that is to run the Monitor (see Chapter 9 for details).
3. Make sure that the installation-dependent global variables have been initialized for the login user. To initialize the installation-dependent global variables, execute the *bstrusers* script. You must also define the BSTR_REALM global variable as the name of the Realm that you want to address.

   For further information about global variables, refer to Appendix G.
4. Enter the following from the command interpreter prompt:

   ```
   $ ams_mon
   ```

The Monitor displays several screens of objects, together with their associated attributes. These screens are called views. There is one view for each Application Management Services object:

- Actor view
- Activity view
- Program view

- Process view
- Node view

The default view, that appears when you invoke the Application Management Services Monitor, is the Actor view (see Figure 13–2).

13.3 General Characteristics

The general characteristics of the Application Management Services Monitor are described below.

Attribute Windows

The Application Management Services Monitor displays the attributes of an Application Management Services object by means of Attribute windows. Each attribute is displayed on a separate line, in two columns (one column contains the name of the attribute field and the other contains its value).

The names of the attribute fields do not always strictly correspond to the names of the attributes in the *BASEstar Open Reference Guide*, as they are often abbreviated. However, they always call the corresponding attribute, unless otherwise specified.

Command Line

The command line at the bottom of the screen displays the available options for the current view. These options include moving from one view to another, switching to another window on a selected object, changing the cursor mode or quitting the Application Management Services Monitor. The options are accessed by selecting the highlighted mnemonic or by pressing the Enter key to switch between windows (see Figure 13–2).

Selected Object

When the cursor is positioned on an object, the object is highlighted and automatically becomes the selected object.

Truncated Fields

Fields that do not fit into a window are truncated and the # symbol is displayed to the left and/or right of the field, depending on where it was truncated. Truncated fields may be viewed in their entirety by positioning the cursor on the required object and then scrolling to the left or right using the following keys:

Key	Function
SPACE	Scroll left
DELETE	Scroll right

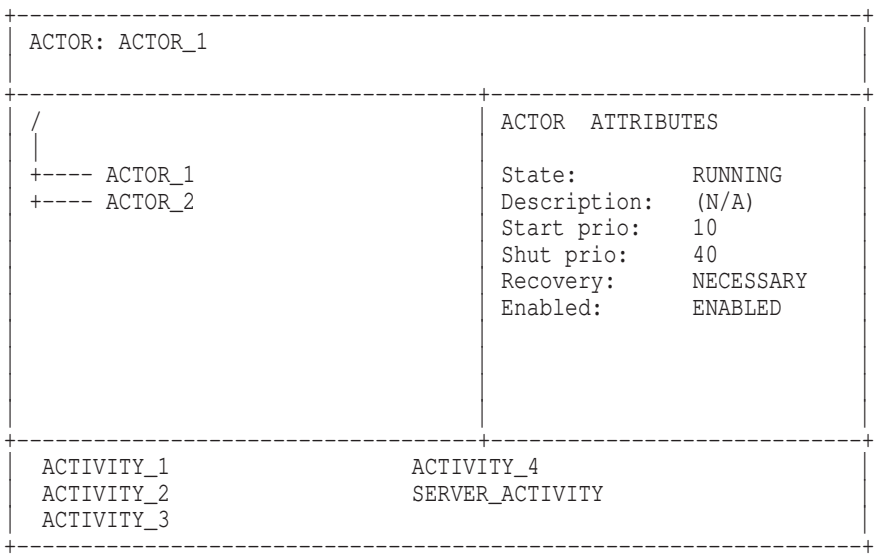
Cursor Mode

The default mode is user-mode, where the cursor is user-driven; that is, it only moves when instigated by the user.

In the follow-updates mode, the cursor follows the Application Management Services objects as they are updated. In this mode, the cursor is automatically positioned on the most recently modified object in the current view.

To invoke Cursor Mode, enter the mnemonic command "M". The Application Management Services Monitor remains within the same view and the command line at the bottom of the screen changes (see Figure 13–1).

Figure 13–1 Cursor Mode



To select the required option, enter the mnemonic command "U" (User-mode), or "F" (Follow-updates). Once the required command has been invoked, the view returns to its previous state.

Quit

To exit from the Application Management Services Monitor within any view, select the Quit option (mnemonic "Q").

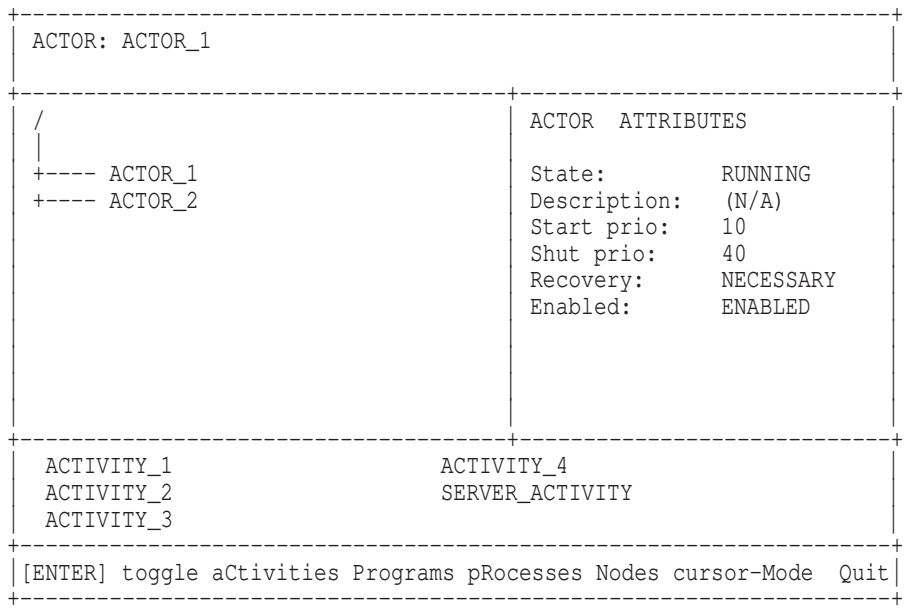
13.4 Views

The following sections explain how to perform operations on the Application Management Services views.

13.4.1 Actor View

Figure 13–2 shows the default view when the Application Management Services Monitor is invoked and displays the Actor objects defined in the Realm. The screen is divided into three windows.

Figure 13–2 Actor View with Actor Attributes



The example in Figure 13–2 shows two Actor objects. The cursor is positioned on the Actor object ACTOR_1 and therefore displays the component Activities of that Actor in the bottom window. The attribute window displays the attributes of that Actor.

The upper left window is the Actor Tree window, which displays the Realm Actor tree. The highlighted cursor is initially positioned in this window. You can use the cursor movement keys to move around the tree of Actor objects by means of the up and down cursor movement keys.

When the cursor is positioned on an Actor, the full name of the Actor is displayed on the top line of the screen. The component Activities of the selected Actor are listed in the bottom window (the Activity window). The attributes of the selected Actor object are displayed in the upper right window (the attribute window).

The Actor tree is displayed in order of name. The Actor names have different terminal attributes, depending on their current Actor state, as follows:

Terminal Attribute	Actor State
Normal	IDLE
Blink	STARTING
Highlight	RUNNING
Blink + Highlight	SHUTTING
Normal	PARTIALLY_RUNNING
Normal	RECOVERY_IN_PROGRESS

The Activity window at the bottom of the screen displays the list of Activities that are components of the selected Actor. In order to display the attributes of an Activity, select the Activity window by pressing Return.

If it is not possible to view all the Activities at the same time, you can use the up and down cursor movement keys to move the cursor through the list of Activity objects and then select the required Activity object. The attribute window displays the attributes of the selected Activity. The top line of the screen displays the full name of the Activity.

Figure 13–3 Actor View with Activity Attributes

```
+-------------------------------------------------------------------+
| ACTIVITY: /ACTOR_1/SERVER_ACTIVITY                                |
|                                                                   |
+-----------------------------------+-------------------------------+
| /                                 | ACTOR  ATTRIBUTES             |
| |                                 |                               |
| +---- ACTOR_1 <<                  | Description:   (N/A)          |
| +---- ACTOR_2                     | Program: /program_example_2   |
|                                   | Profile:       (N/A)          |
|                                   | Running on:    current_node   |
|                                   | State:         RUNNING        |
|                                   | Breakpoints:                  |
|                                   | Start prio:    10             |
|                                   | Shut prio:     230            |
|                                   | Start tim.:    30             |
|                                   | Shut tim.:    30              |
|                                   | Recovery:      NECESSARY      |
|                                   |                               |
+-----------------------------------+-------------------------------+
|  ACTIVITY_1                    ACTIVITY_4                         |
|  ACTIVITY_2                    SERVER_ACTIVITY                    |
|  ACTIVITY_3                                                       |
+-------------------------------------------------------------------+
```

The example in Figure 13–3 illustrates the Activity components of the Actor ACTOR_1. The cursor is positioned on the Activity SERVER_ACTIVITY and the attribute window displays its attributes. Press the Return key to select the Actor window once again.

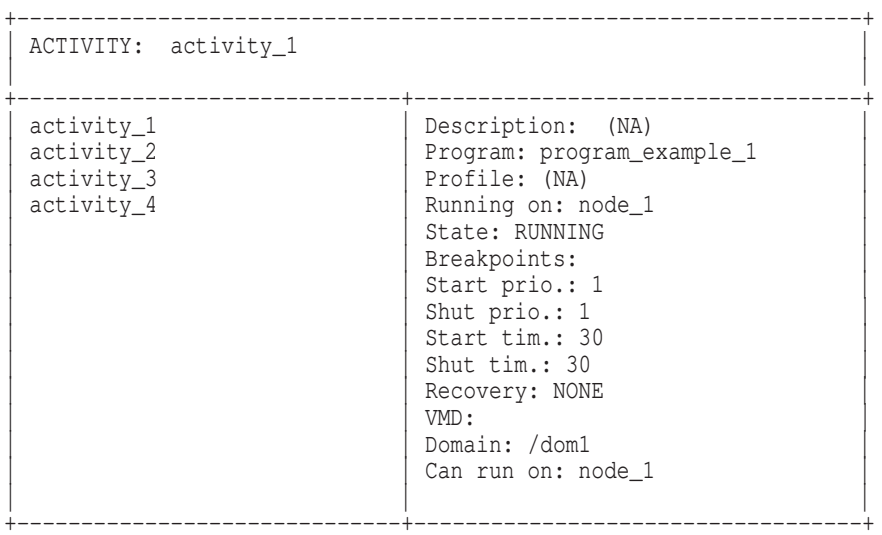
The command line at the bottom of the screen displays the available options. In the case of the Actors view, menu options include jumping to another view, changing the cursor mode, quitting the Application Management Services Monitor, or switching between the Actor window and the Activity window as follows:

Option	Command	Function
[ENTER] toggle	Return key	To move between windows
aCtivities	C	To invoke Activity view
Programs	P	To invoke Program view
Processes	R	To invoke Process view
Nodes	N	To invoke Node view
cursor-Mode	M	User Mode or Follow Updates
Quit	Q	To exit Application Management Services Monitor

13.4.2 Activity View

Figure 13–4 illustrates how the Activity view displays all the Activity objects in the Realm. The screen is divided into two windows.

Figure 13–4 Activity View



Activities are displayed in the Activity window. They are grouped according to Actor and sorted by full name. The top line of the screen displays the full name of the selected Activity and the Attribute window displays its attributes.

The "Running On" attribute in the Attribute window is a combination of the *start_enabled*, *master_state* and *current_node* attributes; that is, it can acquire values of all of these attributes. It can therefore acquire the following values:

- DISABLED - same value as the *start_enabled* attribute
- Not Running - a combination of the *start_enabled* attribute with the ENABLE value and the *master_state* attribute with the IDLE value
- Name of the current Node - same value as the *current_node* attribute

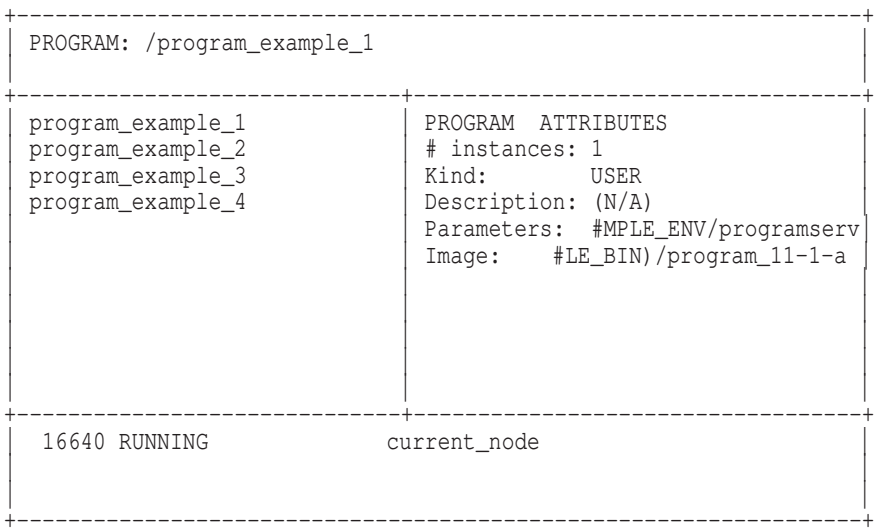
The command line at the bottom of the screen displays the available options. In the case of the Activity view, menu options include jumping to another view, changing the cursor mode, or quitting the Application Management Services Monitor, as follows:

Option	Command	Function
Actors	A	To invoke Actor view
Programs	P	To invoke Program view
pRocesses	R	To invoke Process view
Nodes	N	To invoke Node view
cursor-Mode	M	User Mode or Follow Updates
Quit	Q	To exit Application Management Services Monitor

13.4.3 Program View

Figure 13–5 shows how all Program objects defined in the Realm are displayed in the Program view. The screen is divided into three windows.

Figure 13–5 Program View



The Program window is the upper left window, which displays the Programs sorted by full name. The highlighted cursor is initially positioned in this window. You can use the up and down cursor movement keys to move it through the list of Program objects. The top line of the screen displays the full name of the selected Program.

The Attributes window (upper right window) displays all the attributes of the selected Program. The Process window (bottom window) displays the list of processes that are instances of the selected Program; that is, its associated Program executions. Alternatively, if this Program has not been executed, the message ''No processes'' is displayed.

Should it not be possible to view all of the processes together, select this window by pressing Return, and use the up and down cursor movement keys to move the cursor through the list of processes. This window displays each process on a separate line, in three columns (the first column contains the process identifier, the second contains its state and the third contains the node on which the process is running). Press Return again to select the Program window.

The command line at the bottom of the screen displays the available options. In the case of the Program view, menu options include jumping to another view, changing the cursor mode, quitting the Application Management Services Monitor, or switching between the Program window and the Process window, as follows:

Option	Command	Function
[ENTER] toggle	Return key	To move between windows
Actors	A	To invoke Actor view
aCtivities	C	To invoke Activity view

Option	Command	Function
pRocesses	R	To invoke Process view
Nodes	N	To invoke Node view
cursor-Mode	M	User Mode or Follow Updates
Quit	Q	To exit Application Management Services Monitor

13.4.4 Process View

This view has only one window and displays all existing processes, which are the Program executions associated to the Program objects defined in the Realm. When this view is called, it displays each process on a separate line, together with its process identifier, state and the name of the BASEstar Open Node on which the process is running. The top line of the screen displays the full name of the Program associated with the selected process (see Figure 13–6).

Figure 13–6 Process View

```
+-----------------------------------------------------------------+
| PROGRAM: /program_example_1                                     |
|                                                                 |
+-----------------------------------------------------------------+
| 19606  RUNNING             current_node                         |
| 19593  RUNNING             current_node                         |
| 19642  RUNNING             current_node                         |
| 19632  RUNNING             current_node                         |
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
|                                                                 |
+-----------------------------------------------------------------+
```

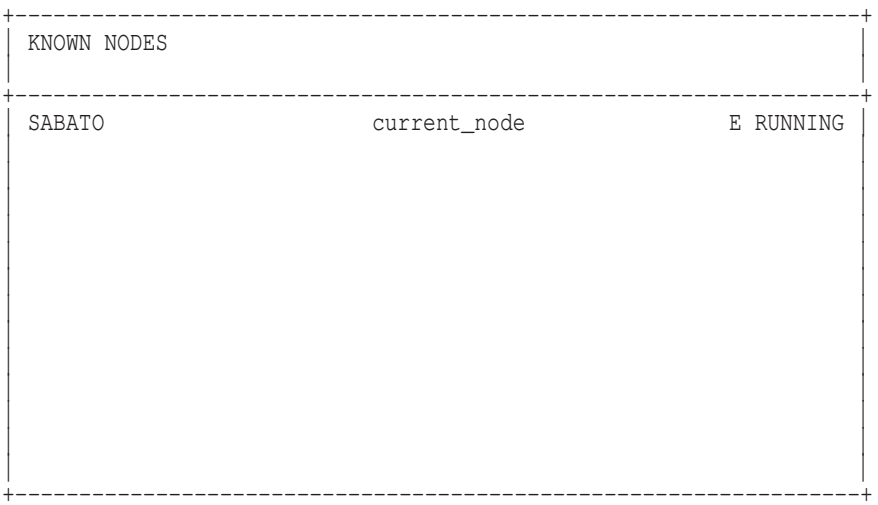
The command line at the bottom of the screen displays the available options. In the case of the process view, menu options include jumping to another view, changing the cursor mode, or quitting the Application Management Services Monitor, as follows:

Option	Command	Function
Actors	A	To invoke Actor view
aCtivities	C	To invoke Activity view
Programs	P	To invoke Program view
Nodes	N	To invoke Node view
cursor-Mode	M	User Mode or Follow Updates
Quit	Q	To exit Application Management Services Monitor

13.4.5 Node View

Figure 13–7 shows how the Node view displays all network Nodes known to Application Management Services.

Figure 13–7 Node View



This view has only one window and displays Node information in four columns. It provides the following information:

- The name of the Node (the physical name attribute value of the associated BASEstar Open Node).
- The name of the BASEstar Open Node.
- The value of the Node state attribute. The letter "E" represents Enabled and the letter "D" represents Disabled.
- The implementation status of the Application Management Services server currently active on the node. There are four possible values:
  - IDLE
    Server not yet activated
  - STARTING
    Server starting up
  - RUNNING
    Server up
  - ABORTED
    Server was up, but went down under abnormal conditions

The command line at the bottom of the screen displays the available options. In the case of the Node view, menu options include jumping to another view, changing the cursor mode, or quitting the Application Management Services Monitor, as follows:

Option	Command	Function
Actors	A	To invoke Actor view

Option	Command	Function
aCtivities	C	To invoke Activity view
Programs	P	To invoke Program view
pRocesses	R	To invoke Process view
cursor-Mode	M	User Mode or Follow Updates
Quit	Q	To exit Application Management Services Monitor

14
Log Services Features

The Log Services collect and record information regarding general events and error conditions detected by the Node and Realm-specific components. BASEstar Open activates a Log Services server process on each Node. The Log Services server process:

- Collects log records sent from the other Node-specific components and writes them to the Node log file
- Creates a log file for each Realm that has been started up
- Collects log records from the components specific to the Realms that have been started up on the Node and writes them to the Realm log file.

The information collected in the log files is intended primarily to help COMPAQ support personnel. However, you can also use it to monitor the behavior of actions performed by your environment components.

14.1 Saving Copies of Log Files

The Log Services operator interface consists of the commands shown in Table 14–1 (see Chapter 15 for a complete description).

Table 14–1 Log Services CLI Commands

Command Name	Description
OPEN LOG	Swaps the contents of a log file working copy into a save copy.
DISPLAY LOG	Displays the contents of a log file working copy.

You can swap the contents of a working copy log file into a save copy by using the OPEN LOG command which also empties up the log file working copy. The OPEN LOG command appends a user-specified string at the end of the just created save copy, and inserts the same string at the top of the emptied log file working copy.

Working and save copies of the log files are assigned the following names, where *N* is a number that is assigned sequentially:

- The name of the log file working copy for the current Node is BASESTARNODE.log, whereas the name of the save copies is BASESTARNODE_*N*.log.
- The name of the working copy log file for a Realm log file is *realm_name*.log, whereas the name of the save copies is *realm_name_N*.log.

BASEstar Open stores the log files in the following directory on each Node started up:

```
BSTR_WORK_ROOT:[node.log]
```

14.1.1 Purging Log Files

With the exception of the working copies, you can delete all the log files that have been created on a Node. To do so, specify the -c option when executing the *bstr_node_start* command for the Node.

14.2 Log File Record Format

Log files are ASCII files containing log records whose format is defined by BASEstar Open and consists of:

- The sequence number associated with a log message generated by the Log Services server. This number is reset to 0 when the Log Services server is started up (that is, when you execute the *bstr_node_start* command), and is incremented by 1 for each message generated.

  The Log Services server writes each log message to a log file. Each message therefore has a sequence number that does not depend on the log file to which it is written. The message number is unique for each message logged on that Node.

- The date and time when the log record was written to the log file
- The process identifier of the component that generated the log record
- An identifier (see Table 14–2) that indicates the environment component responsible for generating the log message.
- The text of the log record message.

Table 14–2 Log File Component Identifier

BASEstar Open Component	Identifier
Node-Specific Environment Components (Started by Environment Commands)	
Name Service	LNSHM LNSSR
Realm-Specific Environment and Application Components (Started by Environment Commands)	
Communication Service	CE CNS RCMTCP
Application Management Services servers	AMS_SERVER
Global Object Services servers	GOM_SERVER
Database Services servers	DB Server
PC Communication server	PC

(continued on next page)

Table 14–2 (Cont.) Log File Component Identifier

BASEstar Open Component	Identifier
Realm-Specific Environment and Application Components (Started from Application Management Services or Command Interpreter)	
Event Services servers	EVM_SERVER
Packet Services servers	MGM_SERVER
Data & Device Services servers	DDM_SERVER
Miscellaneous Components	
CLI instances	CLI_BSTR

Example 14–1 illustrates a sample Node log file.

Example 14–1 Node Log File Example

```
0000000 1999/03/23 12:22:54   20447@ayrton.mln.dec.com WATCHD      : S-BSTR_S_COM_START: Start (BASEstar Open
0000001 1999/03/23 12:22:54   20447@ayrton.mln.dec.com WATCHD      : S-BSTR_S_COM_START: Start (BASEstar Open
0000002 1999/03/23 12:22:54   20447@ayrton.mln.dec.com WATCHD      : S-BSTR_S_COM_WD_LOADEDCONF_OPT: Configur
0000003 1999/03/23 12:23:00   19886@ayrton.mln.dec.com LNSHM       : S-BSTR_S_COM_START: Start (BASEstar Open
0000004 1999/03/23 12:23:00   19886@ayrton.mln.dec.com LNSHM       : S-BSTR_S_COM_LNS_LOADEDCONF: Configurati
0000005 1999/03/23 12:23:00   19886@ayrton.mln.dec.com LNSHM       : S-BSTR_S_COM_LNS_LOADEDCONF_OPT: Configu
0000006 1999/03/23 12:23:00   19886@ayrton.mln.dec.com LNSHM       : I-BSTR_S_COM_LNS_HM_TRY_CONNECT: Host Ma
0000007 1999/03/23 12:23:00   20687@ayrton.mln.dec.com LNSSR       : S-BSTR_S_COM_START: Start (BASEstar Open
0000008 1999/03/23 12:23:00   20687@ayrton.mln.dec.com LNSSR       : S-BSTR_S_COM_LNS_LOADEDCONF_OPT: Configu
0000009 1999/03/23 12:23:00   20687@ayrton.mln.dec.com LNSSR       : S-BSTR_S_COM_LNS_LOADEDCONF: Configurati
0000010 1999/03/23 12:23:01   20687@ayrton.mln.dec.com LNSSR       : S-BSTR_S_COM_LNS_I_AM_MASTER: I am MASTE
0000011 1999/03/23 12:23:01   20687@ayrton.mln.dec.com LNSSR       : S-BSTR_S_COM_LNS_GRANTED_LIC: Granted 0
0000012 1999/03/23 12:23:08   20687@ayrton.mln.dec.com LNSSR       : S-BSTR_S_COM_LNS_HM_CONNECTED: Host Mana
0000013 1999/03/23 12:23:08   19886@ayrton.mln.dec.com LNSHM       : S-BSTR_S_COM_LNS_HM_CONNECTED: Host Mana
0000014 1999/03/23 12:23:08   20447@ayrton.mln.dec.com WATCHD      : I-BSTR_S_COM_LNS_HS_CHANGE: Watchdog sta
0000015 1999/03/23 12:23:08   20687@ayrton.mln.dec.com LNSSR       : I-BSTR_S_COM_LNS_HS_CHANGE: Watchdog sta
```

Example 14–2 shows an example of a Realm log file:

Example 14–2 Realm Log File Example

```
0000051 1999/03/23 15:53:48   21221@ayrton.mln.dec.com BSTR_CLI    :
0000052 1999/03/23 15:53:51   21275@ayrton.mln.dec.com CE          : S-BSTR_S_COM_START: Start (BASEstar Open Serv
0000053 1999/03/23 15:53:51   21290@ayrton.mln.dec.com CNS         : S-BSTR_S_COM_START: Start (BASEstar Open Serv
0000054 1999/03/23 15:53:53   21295@ayrton.mln.dec.com GOM_SERVER  : I-BSTR_S_GOM_SERVER_START: GOM Server startin
0000055 1999/03/23 15:53:53   21295@ayrton.mln.dec.com GOM_SERVER  : Initialize BASEstar environment for BASEstarA
0000056 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : I-BSTR_S_GOM_SNP_LOAD: Load snapshot file: ob
0000057 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : W-BSTR_S_GOM_SNP_NOT_FOUND: Snapshot file not
0000058 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : I-BSTR_S_GOM_SNP_LOAD: Load snapshot file: ob
0000059 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : W-BSTR_S_GOM_SNP_NOT_FOUND: Snapshot file not
0000060 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : I-BSTR_S_GOM_SNP_LOAD: Load snapshot file: ob
0000061 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : W-BSTR_S_GOM_SNP_NOT_FOUND: Snapshot file not
0000062 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : I-BSTR_S_GOM_SNP_LOAD: Load snapshot file: ob
0000063 1999/03/23 15:53:54   21295@ayrton.mln.dec.com GOM_SERVER  : W-BSTR_S_GOM_SNP_NOT_FOUND: Snapshot file not
0000064 1999/03/23 15:53:54   21154@ayrton.mln.dec.com TCP_RCM     : S-BSTR_S_COM_START: Start (BASEstar Open Serv
0000065 1999/03/23 15:53:56   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_AMSSTART: AMS SERVER starting fo
0000066 1999/03/23 15:53:56   21301@ayrton.mln.dec.com AMS_SERVER  : Initialize BASEstar environment for BASEstarA
0000067 1999/03/23 15:53:57   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_CLILOAD: LOAD CONFIGURATION requ
0000068 1999/03/23 15:53:57   21301@ayrton.mln.dec.com AMS_SERVER  : W-BSTR_S_AMS_SNP_NOT_FOUND: Snapshot file not
0000069 1999/03/23 15:53:57   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_READY: AMS Server ready to work
0000070 1999/03/23 15:53:58   21323@ayrton.mln.dec.com COM_SERVER  : th 1; Communication Server (BASEstar Open Ser
0000071 1999/03/23 15:53:58   21323@ayrton.mln.dec.com COM_SERVER  : th 1; realm MLC, port 0, protocol V4.0, lns_n
0000072 1999/03/23 15:53:59   21323@ayrton.mln.dec.com COM_SERVER  : th 1; Listening at 0.0.0.0, port 2443
0000073 1999/03/23 15:53:59   21323@ayrton.mln.dec.com COM_SERVER  : th 1; GOM Server SAP is GOM_SERVER-16.192.64.
0000074 1999/03/23 15:53:59   21323@ayrton.mln.dec.com COM_SERVER  : th 4; Started LNS manager thread
0000075 1999/03/23 15:54:54   21326@ayrton.mln.dec.com BSTR_CLI    : Initialize BASEstar environment for BASEstarA
0000076 1999/03/23 15:56:10   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_ACTSTARTING: Starting actor '/'
0000077 1999/03/23 15:56:10   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_PRCCREA: Process 21335 created o
0000078 1999/03/23 15:56:10   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_PRCCREA: Process 21333 created o
0000079 1999/03/23 15:56:10   21333@ayrton.mln.dec.com EVM_SERVER  : Events Management Services Server (BASEstar O
0000080 1999/03/23 15:56:10   21333@ayrton.mln.dec.com EVM_SERVER  : Initialize BASEstar environment for BASEstarA
0000081 1999/03/23 15:56:10   21335@ayrton.mln.dec.com DTM_SERVER  : Initialize BASEstar environment for BASEstarA
0000082 1999/03/23 15:56:10   21333@ayrton.mln.dec.com EVM_SERVER  : Program '/EVM' running with AMS
0000083 1999/03/23 15:56:10   21333@ayrton.mln.dec.com EVM_SERVER  : Program '/EVM' starting
0000084 1999/03/23 15:56:10   21335@ayrton.mln.dec.com DTM_SERVER  : Program '/DTM' starting
0000085 1999/03/23 15:56:10   21335@ayrton.mln.dec.com DTM_SERVER  : DataDevice Management Services Server (BASEst
0000086 1999/03/23 15:56:10   21335@ayrton.mln.dec.com DTM_SERVER  : Program '/DTM' running with AMS
0000087 1999/03/23 15:56:11   21335@ayrton.mln.dec.com DTM_SERVER  : Starting activity /DTM
0000088 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Start Activity '/EVM'
0000089 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Can't open snapshot file 'domain' - Ignored
0000090 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' beginning
0000091 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' ready to init
0000092 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' enter init phase
0000093 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' ready to start
0000094 1999/03/23 15:56:11   21335@ayrton.mln.dec.com DTM_SERVER  : Can't open snapshot file '/' - Ignored
0000095 1999/03/23 15:56:11   21335@ayrton.mln.dec.com DTM_SERVER  : Domain '/' partially/not loaded
0000096 1999/03/23 15:56:11   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' enter start phase
0000097 1999/03/23 15:56:15   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_ACTSHUTTING: Terminating actor '
0000098 1999/03/23 15:56:15   21335@ayrton.mln.dec.com DTM_SERVER  : Shutting activity /DTM
0000099 1999/03/23 15:56:15   21333@ayrton.mln.dec.com EVM_SERVER  : Shut Activity '/EVM'
0000100 1999/03/23 15:56:15   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' ready to shutdown
0000101 1999/03/23 15:56:15   21335@ayrton.mln.dec.com DTM_SERVER  : Activity '/' ending
0000102 1999/03/23 15:56:15   21333@ayrton.mln.dec.com EVM_SERVER  : Activity '/EVM' enter shutdown phase
0000103 1999/03/23 15:56:15   21333@ayrton.mln.dec.com EVM_SERVER  : Program '/EVM' exiting
0000104 1999/03/23 15:56:15   21333@ayrton.mln.dec.com EVM_SERVER  : Events Management Services Server exited
0000105 1999/03/23 15:56:15   21335@ayrton.mln.dec.com DTM_SERVER  : Starting to terminate the Program...
0000106 1999/03/23 15:56:15   21335@ayrton.mln.dec.com DTM_SERVER  : DataDevice Management Services Server shuttin
0000107 1999/03/23 15:56:15   21335@ayrton.mln.dec.com DTM_SERVER  :  Release BASEstar environment
0000108 1999/03/23 15:56:15   21333@ayrton.mln.dec.com EVM_SERVER  :  Release BASEstar environment
0000109 1999/03/23 15:56:16   21301@ayrton.mln.dec.com AMS_SERVER  : W-BSTR_S_AMS_CHLDEXIT: Child 21333 terminated
Error 0 occurred.
```

(continued on next page)

Example 14–2 (Cont.) Realm Log File Example

```
0000110 1999/03/23 15:56:16   21301@ayrton.mln.dec.com AMS_SERVER  : W-BSTR_S_AMS_CHLDEXIT: Child 21335 termi
Error 0 occurred.
0000111 1999/03/23 15:56:18   21326@ayrton.mln.dec.com BSTR_CLI    :  Release BASEstar environment
0000112 1999/03/23 15:56:31   21301@ayrton.mln.dec.com AMS_SERVER  : W-BSTR_S_AMS_SHUT_REQUEST: Shutdown requ
0000113 1999/03/23 15:56:31   21301@ayrton.mln.dec.com AMS_SERVER  : I-BSTR_S_AMS_AMSSHUT: AMS SERVER shuttin
0000114 1999/03/23 15:56:35   21301@ayrton.mln.dec.com AMS_SERVER  :  Release BASEstar environment
0000115 1999/03/23 15:56:40   21295@ayrton.mln.dec.com GOM_SERVER  : I-BSTR_S_GOM_SERVER_SHUT: GOM Server shu
0000116 1999/03/23 15:56:40   21295@ayrton.mln.dec.com GOM_SERVER  :  Release BASEstar environment
0000117 1999/03/23 15:56:44   21154@ayrton.mln.dec.com TCP_RCM     : S-BSTR_S_COM_END: End
0000118 1999/03/23 15:56:44   21290@ayrton.mln.dec.com CNS         : S-BSTR_S_COM_END: End
0000119 1999/03/23 15:56:44   21275@ayrton.mln.dec.com CE          : S-BSTR_S_COM_END: End
```

14.2.1 Displaying Log Files

The log files are ASCII files that can be displayed by using any platform-dependent tool or editor. You must perform different operations depending on whether you want to display a working copy or save copy of a log file.

See Section 14.1 to know about where log files are stored and about log file naming conventions.

14.2.1.1 Displaying a Save Copy

To display a save copy of a log file, use any command or utility that can display an ASCII file.

14.2.1.2 Displaying a Working Copy

The working copy log files are opened by the Log Services server and thus they cannot be displayed by using the standard tools when the Node is started up. Thus, the DISPLAY LOG command has been provided to allow the display of log file working copies. This command also can display the last lines of a log file automatically refreshing them as soon as a new record is inserted in the displayed log file. This allows you to monitor record insertion in the selected log file.

14.3 Log Services Server Activities

The Log Services server process is a Node-specific component which is started up when you execute the *bstr_node_start* command. In fact, it is the first process to be started up, so it is already available for the other Nodes.

In turn, the Log Services server logs some record information regarding general events and unrecoverable errors detected during execution in the platform-dependent system log. For example, as soon as the Log Services server is started up, it logs the following message to the system log file:

```
Jul 14 11:35:12 localhost: 21124 BASESTAR LOG: Server started,
output file: ./log_server.out, log directory: ./
```

15

Log Services Command Reference

DISPLAY LOG

Displays the contents of a log file working copy.

Format

DISPLAY LOG

Command Qualifier	Default
[-REALM realm_name]	Current Realm
[-NODE]	Current Node
[-HEAD head_lines]	10
[-TAIL tail_lines]	10
[-CONTINUE]	

Description

The DISPLAY LOG command allows you to display records from both Realm and Node log file working copies.

The **–REALM** and **–NODE** qualifiers are mutually exclusive:

- If you specify the **–REALM qualifier**, the *realm_name* argument indicates the name of the Realm.
- If you specify the **–NODE** qualifier, BASEstar Open displays the log file for the Node that you are addressing.
- If you do not specify either the **–REALM** or the **–NODE** qualifier, BASEstar Open assumes the value of the **–NODE** qualifier by default.

If you provide a value for the *head_lines* (*tail_lines*) argument, the DISPLAY LOG command displays the requested lines starting from the top (bottom) of the selected log file.

If you specify both the -CONTINUE and -TAIL qualifiers, the DISPLAY LOG command displays the last *tail_lines* and refresh them automatically as soon as a new record is inserted in the displayed log file. This allows you to monitor record insertion in the selected log file. To terminate execution of a command, press Return

Qualifiers

-REALM realm_name

The *realm_name* argument specifies the local name of the Realm whose log file you want to display. The Realm in question already be started on the current Node.

This qualifier and the -NODE qualifier are mutually exclusive.

-NODE

Directs the DISPLAY LOG command to display the log file of the current Node.

This qualifier and the -NODE qualifier are mutually exclusive. If you do not specify either the **–REALM** or the **–NODE** qualifier, BASEstar Open assumes the value of the -NODE qualifier by default.

-HEAD head_lines

Directs the command to display the number of lines specified by the *head_lines* argument starting from the top of the selected log file. The -TAIL and -HEAD qualifiers are mutually exclusive.

-TAIL tail_lines

Directs the command to display the number of lines specified by the *tail_lines* argument starting from the bottom of the selected log file. The -TAIL and -HEAD qualifiers are mutually exclusive.

-CONTINUE

Directs the DISPLAY LOG command to display the last *tail_lines* and to refresh them as soon as a new record is inserted in the displayed log file. This allows you to monitor record insertion in the selected log file. When specified, the -CONTINUE qualifier must be specified together with the -TAIL qualifier.

OPEN LOG

Swaps the contents of a log file working copy into a save copy.

Format

OPEN LOG

Command Qualifier	Default
[-REALM realm_name]	Current Realm
[-NODE]	Current Node
[-DESCRIPTION string]	No Description

Description

The OPEN LOG command performs the following actions for the current Node or the specified Realm:

1. Closes the working copy log file and renames it into a save copy.
2. Appends the string specified by the -DESCRIPTION qualifier at the end of the just created save copy.
3. Creates and opens a new empty working copy log file and inserts the same string specified by the -DESCRIPTION qualifier.

The -REALM and -NODE qualifiers are mutually exclusive. If neither the -REALM nor the -NODE qualifier is specified, the -NODE qualifier is assumed by default. Assuming that *N* is a progressively assigned number:

- The name of the log file for the current Node is BASESTARNODE.log, while save copies have name BASESTARNODE_*N*.log.
- The name of the log file for a Realm log file is *realm_name*.log, while save copies have name BASESTARNODE-*N*.log.

Log files are stored under the following directory:

```
BSTR_WORK_ROOT:[node.log]
```

Qualifiers

-REALM realm_name

Tells the OPEN LOG command to address the working copy of the log file for the *realm_name*. If you do not specify this qualifier, BASEstar Open assumes by default the Realm specified by the BSTR_REALM global variable. The Realm in question must already be started on the current Node.

This qualifier and the -NODE qualifier are mutually exclusive.

-NODE

Tells the OPEN LOG command to address the working copy of the log file for the current Node.

This qualifier and the **–NODE** qualifier are mutually exclusive. If you do not specify either the **–REALM** or the **–NODE** qualifier, BASEstar Open assumes the **–NODE** qualifier by default.

-DESCRIPTION string

User-defined information that appears at the end of a save copy and at the top of the working copy log file.

The description need not conform to any format and must be enclosed between quotation marks (" "). The *string* argument is a string of maximum 128 characters.

Part IV

Installation and System Management Appendices

This section contains a series of appendices pertaining to the installation and management of BASEstar Open Server on OpenVMS:

- Files and Logical Names Installed by BASEstar Open Server
- BASEstar Open Installation Messages
- Sample Installation Log Files for BASEstar Open Server on OpenVMS
- Additional Files Created During Installation
- Device Connection Manager Installation Messages
- Sample Installation Log File of the Device Connection Manager
- BASEstar Open Global Variables
- Environment and Parameter Files
- Environment Component Processes
- Managing Snapshot Files in a Distributed Environment

A

Files and Logical Names Installed by BASEstar Open Server

The BASEstar Open Server installation procedure creates some directories, installs a number of files, and defines some logical names, as listed in the following sections.

A.1 Directories Created by BASEstar Open Server

Table A–1 lists the directories and files that are created on your system when you install BASEstar Open Server.

Table A–1 BASEstar Open Server Directories

Directory	Description
BIN	Executable programs distributed with BASEstar Open (for example, Server and CLI)
DOC	Files constituting the product documentation addendum (for example, the Release Notes)
ETC	Configuration files and programs or command procedures for BASEstar Open management
INCLUDE	Include files[1]
LIB	Libraries and message catalogues
IVP	Installation Verification Procedure (IVP) files

[1]These files are only installed if either of the following license are installed:

- BSTROPEN-DEVELOP
- BSTROPEN-DEVELOP-USER

A.2 Files Created by BASEstar Open Server

Table A–2 lists the files installed by BASEstar Open Server.

Table A–2 Files Installed by BASEstar Open Server

Directory	File
[BIN]	AMS_MON.EXE
	AMS_SERVER.EXE

(continued on next page)

Table A–2 (Cont.) Files Installed by BASEstar Open Server

Directory	File
	BSTR$EDIT.COM
	BSTR_ENV_SHOW.COM
	BSTR_FORCEX.EXE
	BSTR_LICENCE_CHECK.EXE
	BSTR_MON.EXE
	BSTR_REALM_CHECK_ENV.COM
	BSTR_REALM_SETUP_DB.COM
	BSTR_REALM_SETUP_NODE.COM
	BSTR_REALM_SHUT_DB.COM
	BSTR_REALM_SHUT_NODE.COM
	BSTR_REALM_START_DB.COM
	BSTR_REALM_START_NODE.COM
	BSTR_REALM_UNSET_DB.COM
	BSTR_REALM_UNSET_NODE.COM
	BSTR_RUN.COM
	BSTR_SERVER_START.COM
	BSTR_UTL_TGEN.EXE
	CE.EXE
	CLI_BSTR.EXE
	CLI_TRC.EXE
	CNF_DBSERVER.EXE
	CNF_DB_INIT.EXE
	CNS.EXE
	COATTMSG.EXE
	COBRFDMP.EXE
	COEDIPAR.EXE
	COGETLNM.EXE
	COGETPAR.EXE
	COMALIVE.EXE
	COM_CHOOSE_CECONF.COM
	COM_CHOOSE_INSTF.COM
	COM_CONTROL.COM
	COM_START_LNS.COM
	COM_START_LNSMON.COM
	COM_START_WATCHDOG.COM
	COM_SYNC_LNS.COM
	COM_SYNC_WATCHDOG.COM
	DDM_SERVER.EXE

(continued on next page)

Table A–2 (Cont.) Files Installed by BASEstar Open Server

Directory	File
	ENV_SHOW.EXE
	EVM_SERVER.EXE
	EVM_TEST.EXE
	GET_DBINFO.COM
	GET_PATH.EXE
	GOM_SERVER.EXE
	HMCTRL.EXE
	INTERNAL.DIR
	LCPISRV.EXE
	LNSHM.EXE
	LNSSR.EXE
	LNS_VMSMON.EXE
	LOGCNTRL.EXE
	LOGTOOL.EXE
	LOG_SERVER.EXE
	MGM_SERVER.EXE
	RCMTCP.EXE
	TAIL.EXE
	TRS_PATH.EXE
	WATCHD.EXE
	WDCTRL.EXE
[DOC]	BSTROSVA032.RELEASE_NOTES
[ETC]	BSTRADMIN.COM
	BSTRUSERS.COM
	BSTR_IVP.COM
	BSTR_NODE_SETUP.COM
	BSTR_NODE_SHUT.COM
	BSTR_NODE_START.COM
	BSTR_NODE_UNSET.COM
	BSTR_VER_SHOW.COM
	HOST_SHUTDOWN.DIR
	HOST_STARTUP.DIR
	INSTALLATION_FILE.TEMPLATE
	IVP_INSTALLATION_FILE.TEMPLATE
	REALM_SHUTDOWN.DIR
	REALM_STARTUP.DIR

(continued on next page)

Table A–2 (Cont.) Files Installed by BASEstar Open Server

Directory	File
[ETC.HOST_SHUTDOWN]	SHUT00_LOG.COM
	SHUT01_COM.COM
	SHUT02_COM.COM
	SHUT03_COM.COM
[ETC.HOST_STARTUP]	CHECK_COM.COM
	CLEAN00_LOG.COM
	CLEAN01_COM.COM
	CLEAN02_COM.COM
	CLEANUP_COM.COM
	START00_LOG.COM
	START01_COM.COM
	START02_COM.COM
	START03_COM.COM
	SYNC00_LOG.COM
	SYNC01_COM.COM
	SYNC02_COM.COM
	SYNC03_COM.COM
[ETC.REALM_SHUTDOWN]	SHUT00_COM.COM
	SHUT01_GOM.COM
	SHUT02_AMS.COM
	SHUT03_COM.COM
[ETC.REALM_STARTUP]	CHECK_COM.COM
	CLEAN00_COM.COM
	CLEAN01_GOM.COM
	CLEAN02_AMS.COM
	CLEANUP_AMS.COM
	CLEANUP_GOM.COM
	START00_COM.COM
	START01_GOM.COM
	START02_AMS.COM
	START03_COM.COM
	SYNC00_COM.COM
	SYNC01_GOM.COM
	SYNC02_AMS.COM

(continued on next page)

Table A–2 (Cont.) Files Installed by BASEstar Open Server

Directory	File
	SYNC03_COM.COM
[INCLUDE][1]	BSTR.H
	BSTRMSG.H
	BSTRMSGL.H
	BSTROSD.H
	BSTRPPB.H
	BSTRPUB.H
	BSTRSTAT.H
	BSTRVMS.H
	TRLIB.H
[IVP]	
	BSTR_KIT.LIS
	IVP_OBJECTS_FILE.CLI
	IVP_SETUP_FILE.CLI
	IVP_SHOW_FILE.CLI
	IVP_SHUT_FILE.CLI
	IVP_START_FILE.CLI
	IVP_TYPES_FILE.CLI
[LIB]	BSTR$CLI_HELP.HLB
	BSTR$TRC_HELP.HLB
	BSTR.OPT
	BSTR_CLI_CAT.CAT
	BSTR_CMA.OPT
	BSTR_CMN_SHR.EXE
	BSTR_CMN_SHR_CMA.EXE
	BSTR_DDM_SHR.EXE
	BSTR_DDM_SHR_CMA.EXE
	BSTR_FDB_SHR.EXE
	BSTR_FDB_SHR_CMA.EXE
	BSTR_LCPI_SHR.EXE
	BSTR_LCPI_SHR_CMA.EXE
	BSTR_MSG.CAT
	BSTR_MSG.EXE
	BSTR_MSGPTR.OBJ

[1]These files are only installed if the development license is installed.

(continued on next page)

Table A–2 (Cont.) Files Installed by BASEstar Open Server

Directory	File
	BSTR_PODB.RBF
	BSTR_SHR.EXE
	BSTR_SHR_CMA.EXE
	BSTR_TRC_SHR.EXE
	BSTR_TRC_SHR_CMA.EXE
	CLI_VMS.SYNTAX
	TRC_VMS.SYNTAX

A.3 Systemwide Logical Names Created by BASEstar Open Server

The installation procedure does not define any logical names. BASEstar Open Server process logical names are defined when the `bstrusers.com` command procedure is executed. For further information, refer to Appendix G.

A.4 Files Created by DEComni API

When you install DEComni API, a number of files are created. This section lists the files according to their location.

A.4.1 DEComni API Files for Application Development

Table A–3 lists the files that application development files that DEComni API places in *omni_lib_loc* and *ods_dest*.

Table A–3 Files Installed by DEComni API

Directory	File
omni_lib_loc[2]	LINK_BASICSHR.OPT
	LINK_OMNISHR.OPT
	LINK_OMNISRV.OPT
	OMNI_IDENT.OPT
	OMNI_UI_SQL_STUBS.OBJ
	RDB.OPT
ods_dest[1] [OMNI.ODS.INCLUDE]	ODS_MSG.H
	ODS_DEFS.H

[1] By default, these files are placed on the disk SYS$SYSDEVICE, but the disk location can be chosen during installation.

[2] By default, these files are placed in SYS$SYSDEVICE:[OMNI.LIBRARY], but the disk location can be chosen during installation.

A.4.2 DEComni API Files in VMI$ROOT

Table A–4 lists the files that DEComni API places in VMI$ROOT.

Table A–4 Files Installed by DEComni API

Directory	File
[SYSMGR]	OMNI_INT_CONFIG.COM
	OMNI_CMA_LOGICALS.COM[2]
	OMNI_STOP.COM
	OMNI_CONFIG.COM
[SYSHLP	OMNI$ODF_HELP.HLB
[SYS$STARTUP]	OMNI_STARTUP.COM
	OMNI_SERVER.COM
	OMNI_LOGICALS.COM
[SYSTEST]	OMNI_IVP.COM
	OMNI_IVP_CONFIGURE.NCL
	OMNI_IVP_INIT.OBJ
	OMNI_IVP_RESP.OBJ
	OMNI_IVP.OPT
[SYSHLP.EXAMPLES.OMNI]	OMNI_CONFIGURE.NCL
	OMNI_CONVERT.EDT
	OMNI_IVP_INIT.C
	OMNI_IVP_RESP.C
	OMNI_LINK_OPTIONS.OPT
	OMNI_ODF_CONVERT.COM
[SYSLIB]	OMNI_API_PROTOS.H

[2]OpenVMS VAX only

(continued on next page)

Table A–4 (Cont.) Files Installed by DEComni API

Directory	File
	OMNI_CODES.*filetype*[1]
	OMNI_DEFS.*filetype*[1]
	OMNI_INTEGRATOR2_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR3_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR4_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR5_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR6_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR7_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR8_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR9_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR10_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR11_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR12_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR13_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR14_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR15_DEFS_INCLUDE.*filetype*[1]
	OMNI_INTEGRATOR16_DEFS_INCLUDE.*filetype*[1]
[SYSLIB]	OMNI_DATABASE.RDB
	OMNI_DATABASE.SNP

[1]These files are installed as the following filetypes:

- .ADA
- .BAS
- .FOR
- .H
- .MAR
- .PAS
- .PLI
- .R32
- .REQ

A.4.3 DEComni API Image Files in VMI$ROOT

Table A–5 lists the image files that DEComni API installs.

Table A–5 Image Files Installed by DEComni API

Directory	File
[SYSMSG]	OMNI_CODES_MSG.EXE
[SYSLIB]	
	OMNI_BASIC_SHR.EXE
	OMNI_CMA$LIB_SHR.EXE[4]
	OMNI_CMA$OPEN_LIB_SHR.EXE[4]
	OMNI_CMA$OPEN_RTL.EXE[4]
	OMNI_CMA$RTL.EXE[4]
	OMNI_PORT_SHR.EXE
	OMNI_SHR.EXE
[SYSEXE]	
	OMNI_ODF.EXE[4]
	OMNI_SERVER.EXE
	OMNIVIEW.EXE
	OMNIVIEW.UID
[SYSHLP]	
	OMNIVIEW_HELP.TXT
[SYSHLP.EXAMPLES.OMNI]	
	OMNI_RUN_TIME_PI_DOM.C
	OMNI_RUN_TIME_VAR.C
	OMNI_RUN_TIME_VMD.C
ods_dest [1] **[OMNI.ODS.EXAMPLES]**	
	CONVERT_MDS_TO_ODS.EXE[4]
	ODS_EXAMPLE.C
ods_dest[1] **[OMNI.ODS.EXE]**	
	ODSCL.EXE
	ODSVIEW.EXE
	ODS_CACHE_UTILITY.EXE
	ODS_REHASH_DIRECTORY.EXE
	ODS_SCHEMA_UTILITY.EXE
	ODS_TESTER.EXE

[1] By default, these files are placed on the disk SYS$SYSDEVICE, but the disk location can be chosen during installation.

[4] OpenVMS VAX only

(continued on next page)

Table A–5 (Cont.) Image Files Installed by DEComni API

Directory	File
ods_dest[1] **[OMNI.ODS.HELP]**	
	ODSCL_ADD_EXPRESSION.TXT
	ODSCL_ATTR_ABBREV.TXT
	ODSCL_ATTR_EXPRESSION.TXT
	ODSCL_ATTR_PROMPT.TXT
	ODSCL_COMMANDS.TXT
	ODSCL_COMMAND_ELEMENTS.TXT
	ODSCL_COMMAND_SYNTAX.TXT
	ODSCL_CONTINUATION.TXT
	ODSCL_DEREGISTER.TXT
	ODSCL_DEREGISTER_EXAMPLES.TXT
	ODSCL_DEREGISTER_SYNTAX.TXT
	ODSCL_DIRECTORY_PROMPT.TXT
	ODSCL_DISPLAYS.TXT
	ODSCL_EXIT.TXT
	ODSCL_FILE_PROMPT.TXT
	ODSCL_HELP.IDX
	ODSCL_HELP.TXT
	ODSCL_INVOKING.TXT
	ODSCL_LIST.TXT
	ODSCL_LIST_EXAMPLES.TXT
	ODSCL_LIST_SYNTAX.TXT
	ODSCL_LOCAL_DIRECTORY.TXT
	ODSCL_MATCH_EXPRESSION.TXT
	ODSCL_MATCH_PROMPT.TXT
	ODSCL_MODIFY.TXT
	ODSCL_MODIFY_EXAMPLES.TXT
	ODSCL_MODIFY_PROMPT.TXT
	ODSCL_MODIFY_SYNTAX.TXT
	ODSCL_NAME_EXPRESSION.TXT
	ODSCL_NAME_PROMPT.TXT
	ODSCL_OC_ABBREV.TXT
	ODSCL_OPTION_ABBREV.TXT
	ODSCL_PATH_EXPRESSION.TXT
	ODSCL_PATH_PROMPT.TXT
	ODSCL_PROMPTS.TXT

[1]By default, these files are placed on the disk SYS$SYSDEVICE, but the disk location can be chosen during installation.

(continued on next page)

Table A–5 (Cont.) Image Files Installed by DEComni API

Directory	File
ods_dest[1] **[OMNI.ODS.HELP]**	
	ODSCL_READ.TXT
	ODSCL_READ_EXAMPLES.TXT
	ODSCL_READ_SYNTAX.TXT
	ODSCL_REGISTER.TXT
	ODSCL_REGISTER_EXAMPLES.TXT
	ODSCL_REGISTER_SYNTAX.TXT
	ODSCL_REMOVE_EXPRESSION.TXT
	ODSCL_REPLACE_EXPRESSION.TXT
	ODSCL_SET.TXT
	ODSCL_SET_EXAMPLES.TXT
	ODSCL_SET_SYNTAX.TXT
	ODSCL_SHOW.TXT
	ODSCL_SHOW_EXAMPLES.TXT
	ODSCL_SHOW_SYNTAX.TXT
	ODSCL_TO_FILE_EXPRESSION.TXT
	ODSCL_TO_PROMPT.TXT
	ODSCL_WITH_EXPRESSION.TXT
	ODSCL_WITH_PROMPT.TXT
	ODSVIEW_HELP.TXT
ods_dest[1] **[OMNI.ODS.LOCAL]**	
	ODSCL_PARSE_TBL.BIN
	ODSVIEW.UID
	ODS_COPY_DIALOG.UID
	ODS_ENTRY_DIALOG.UID
	ODS_KNOWN_ATTRIBUTES_TYPES.DAT[5]
	ODS_KNOWN_OBJECT_CLASSES.DAT[5]
	ODS_MODIFY_DIALOG.UID
	ODS_SELBOX_DIALOG.UID
	ODS_VERSION.DAT
ods_dest[1] **[OMNI.ODS.SCRIPTS]**	
	ODS_IVP.COM
	ODS_STARTUP.COM

[1]By default, these files are placed on the disk SYS$SYSDEVICE, but the disk location can be chosen during installation.

[5]OpenVMS Alpha only

(continued on next page)

Table A–5 (Cont.) Image Files Installed by DEComni API

Directory	File
ods_dest[1] **[OMNI.ODS.SCRIPTS]**	
	DOMAIN_SCHEMA.COM
	MISC_SCHEMA.COM
	OMNI_SCHEMA.COM
	PI_SCHEMA.COM
	VAR_SCHEMA.COM
	VMD_SCHEMA.COM
	INT_*n*S.OPT[3]
	INT_*n*B.OPT[3]
	INT_*n*0.OPT[3]

[1]By default, these files are placed on the disk SYS$SYSDEVICE, but the disk location can be chosen during installation.

[3]Where *n* indicates the value 1 to 16.

B

BASEstar Open Installation Messages

B.1 BASEstar Open Message Listing

The installation messages are in alphabetical order by the IDENT abbreviation.

BADDEVICE, Error in disk name

Explanation: The user entered an invalid disk name (the colon (:) is present but is not the last character).

User Action: Enter another name.

DEVNOTDISK, The target device is not a disk

User Action: Enter another name.

DEVNOTMNT, The target disk is not mounted

User Action: Enter another name.

DEVWRLCK, The target disk is write locked

User Action: Enter another name.

DEVALL, The target disk is allocated

User Action: Enter another name.

FAKEDISK, The target disk does not exist

User Action: Enter another name.

INSSYSDISK, The system disk does not have xxx free blocks

User Action: Try to delete useless files until there are xxx free blocks on the system disk (when BASEstar Open is being installed on the system disk)

INSTGTDISK, The target disk does not have xxx free blocks

User Action: Try to delete useless files until there are xxx free blocks on the system disk (when BASEstar Open is being installed on disk different from the system disk)

INSGBLPAGES, Free global pages are insufficient

User Action: Increase the number of global pages.

INSGBLSECTS, Free global sections are insufficient

User Action: Increase the number of global sections.

INSASTLM, Insufficient ASTLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

B.1 BASEstar Open Message Listing

INSBIOLM, Insufficient BIOLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSBYTLM, Insufficient BYTLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSDIOLM, Insufficient DIOLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSENQLM, Insufficient ENQLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSFILLM, Insufficient FILLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSPGFLQUOTA, Insufficient PGFLQUOTA. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSTQELM, Insufficient TQELM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSPRCLM, Insufficient PRCLM. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSWSDEF, Insufficient WSDEF. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSWSEXTENT, Insufficient WSEXTENT. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

INSWSQUOTA, Insufficient WSQUOTA. You should have at least xxx.

User Action: Increase the insufficient quota for the SYSTEM account.

NOMOTIF, DECwindows Motif is not installed on the system

Explanation: The user wants to install the Graphic Configuration Utility but does not have DECwindows Motif Version 1.1 on the system.

User Action: Install DECwindows Motif Version 1.1 or avoid installing the Graphic Configuration Utility.

NOMOTIFVER, This product requires at least version 1.1 of DECwindows Motif

Explanation: The user wants to install the Graphic Configuration Utility but does not have DECwindows Motif Version 1.1 or later on the system.

User Action: Install DECwindows Motif Version 1.1 or avoid installing the Graphic Configuration Utility.

NOPRIV, Insufficient privileges to run the installation and the IVP

Explanation: The user does not have all the privileges.

User Action: Grant all the privileges to SYSTEM account.

NOUCX, TCP/IP Services are not installed on the system

User Action: Install Version 3.2 or later of TCP/IP Services.

NOUCXVER, This product requires at least version 3.2 of TCP/IP Services

User Action: Install Version 3.2 or later of TCP/IP Services.

ROOTSYN, BSTROSVA V3.2 root specification syntax error

User Action: Enter another name.

USRABORTED, Installation aborted by the user.

Explanation: The user, answering a specific question, decided not to continue.

User Action: None.

VMSVERSION, BSTROSVA V3.2 requires at least version 6.1 of VMS

Explanation: The VMS version on the system does not meet BASEstar Open requirements.

User Action: Upgrade to the right version

C

Sample Installation Log Files for BASEstar Open Server on OpenVMS

This appendix contains complete sample logs relative to the installation of BASEstar Open Server for OpenVMS on Alpha and VAX platforms, where the following requirements have already been installed:

- A valid BASEstar Open license
- A valid version of DEComni API

C.1 Installation of BASEstar Open Server for OpenVMS Alpha

```
 OpenVMS AXP Software Product Installation Procedure V7.1

It is 14-APR-1999 at 13:30.

Enter a question mark (?) at any time for help.

* Are you satisfied with the backup of your system disk [YES]?
* Where will the distribution volumes be mounted: sys$login

Enter the products to be processed from the first distribution volume set.

* Products: bstrosva032
* Enter installation options you wish to use (none):

The following products will be processed:

  BSTROSVA V3.2

        Beginning installation of BSTROSVA V3.2 at 13:30

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.

******************************************************************************

© Digital Equipment Corporation. 1999. All Rights Reserved.

   Restricted Rights: Use, duplication, or disclosure by the U.S.
   Government is subject to restrictions as set forth in subparagraph
   (c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
   52.227-14 Alt. III, as applicable.

   This software is proprietary to and embodies the confidential
   technology of Digital Equipment Corporation. Possession, use, or
   copying of this software and media is authorized only pursuant to a
   valid written license from DIGITAL or an authorized sublicensor.

            - BASEstar Open V3.2-0 Installation Procedure -
```

Sample Installation Log Files for BASEstar Open Server on OpenVMS
C.1 Installation of BASEstar Open Server for OpenVMS Alpha

```
  This procedure decides from the license which component is to be
  installed. If you do not have any license, the procedure allows you
  to install the runtime option, but does not execute the IVP.

  This procedure will also give you the option of installing the
  Device Connection component (DEComni). If you wish to use the
  device access capability of this version of BASEstar Open, you
  must install the Device Connection component included in this kit.

*****************************************************************************

BASEstar Open Development option will be installed

* Do you want to install the Device Connection component (DEComni) [Y]? n

BASEstar Open can be installed on any disk of your system.
    There is no default disk: you must enter a disk name.
    The following disks are available on the system.

DOLORE$DKB100:          Mounted              0  BSTRDISK        107397     1   1
DOLORE$DKB300:          Mounted              0  ALPHASYS71      116841   398   1
DOLORE$DKB400:          Mounted              0  SWAPDISK        429387     3   1

* Target disk: DOLORE$DKB100:

* BSTROSVA V3.2 root directory [BSTROSVA032]:

    You are going to install BSTROSVA V3.2 on

    Target disk:    DOLORE$DKB100:
    Root directory: [BSTROSVA032]

* Is this correct [Y]?
* Do you want to purge files replaced by this installation [YES]?
* Do you want to run the IVP after the installation [YES]?

*****************************************************************************

   All the questions regarding the installation of BASEstar Open
   have been answered. The installation will take from 15 to 30 minutes,
   and the IVP about 20 minutes, depending on the CPU type and
   the system workload.

*****************************************************************************


* Do you want to continue [Y]?
%VMSINSTAL-I-RESTORE, Restoring product save set B ...
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...


           +-------------------------------------------------+
           |              Starting the IVP for               |
           |               BASEstar Open V3.2                |
           +-------------------------------------------------+


Logical name BSTR_REALM temporarily defined as IVP

BASEstar Open Server V3.2

Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.

Restricted Rights: Use, duplication, or disclosure by the U.S
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.

This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.
BASEstar Open Server V3.2
```

```
Checking the environment
Calling BSTR$ROOT:[ETC.HOST_STARTUP]CHECK_COM.COM;1 ...
Cleanup
Calling BSTR$ROOT:[ETC.HOST_STARTUP]CLEANUP_COM.COM;1 ...
Cleaning the communication directory.
Cleaning the communication name table.
4 phases will be executed
Startup phase 0
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START00_LOG.COM;1 ...
Starting up the LOG Server.
%RUN-S-PROC_ID, identification of created process is 0000011A
Synchronizing after phase 0
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC00_LOG.COM;1 ...
Waiting for the LOG Server to start...
Startup phase 1
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START01_COM.COM;1 ...
Creating the node table.
Starting the Watchdog.
%RUN-S-PROC_ID, identification of created process is 0000011B
Synchronizing after phase 1
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC01_COM.COM;1 ...
Waiting for the Watchdog to start...
Startup phase 2
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START02_COM.COM;1 ...
Starting the LNS.
Starting LNS for name space BSTR
%RUN-S-PROC_ID, identification of created process is 0000011C
Synchronizing after phase 2
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC02_COM.COM;1 ...
Waiting for the LNS to start...
Startup phase 3
Calling BSTR$ROOT:[ETC.HOST_STARTUP]START03_COM.COM;1 ...
Starting the LNS monitor for VMS
Starting LNS for name space BSTR
%RUN-S-PROC_ID, identification of created process is 00000122
Synchronizing after phase 3
Calling BSTR$ROOT:[ETC.HOST_STARTUP]SYNC03_COM.COM;1 ...
Waiting for the LNS monitor for VMS.
All phases completed
Executing BSTR_REALM_SETUP_NODE ...
Executing BSTR_REALM_START_NODE ...
Configuring AMS objects ...
Starting DTM and EVM servers ...
Checking DTM and EVM servers ...
Shutting down DTM and EVM servers ...
Executing BSTR_REALM_SHUT_NODE ...
Executing BSTR_REALM_UNSET_NODE ...
Executing BSTR_NODE_SHUT ...
Executing BSTR_NODE_UNSET ...
Environment correctly restored

IVP TERMINATED SUCCESSFULLY
 Installation of BSTROSVA V3.2 completed at 13:45

    Adding history entry in VMI$ROOT:[SYSUPD]VMSINSTAL.HISTORY

    Creating installation data file: VMI$ROOT:[SYSUPD]BSTROSVA032.VMI_DATA

$
```

C.2 Installation of BASEstar Open Server for OpenVMS VAX

```
           OpenVMS VAX Software Product Installation Procedure V6.1

It is 9-APR-1999 at 11:24.

Enter a question mark (?) at any time for help.

%VMSINSTAL-W-ACTIVE, The following processes are still active:
 _TNA11:
* Do you want to continue anyway [NO]? y
* Are you satisfied with the backup of your system disk [YES]?

The following products will be processed:

  BSTROSVV V3.2

 Beginning installation of BSTRSVV V3.2 at 11:25

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
******************************************************************************

(C) Digital Equipment Corporation. 1999 All Rights Reserved.

   Restricted Rights: Use, duplication, or disclosure by the U.S.
   Government is subject to restrictions as set forth in subparagraph
   (c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
   52.227-14 Alt. III, as applicable.

   This software is proprietary to and embodies the confidential
   technology of Digital Equipment Corporation. Possession, use, or
   copying of this software and media is authorized only pursuant to a
   valid written license from DIGITAL or an authorized sublicensor.


            - BASEstar Open V3.2-0 Installation Procedure -

   This procedure decides from the license which component is to be
   installed. If you do not have any license, the procedure allows you
   to install the runtime option, but does not execute the IVP.

   This procedure will also give you the option of installing the
   Device Connection component (DEComni). If you wish to use the
   device access capability of this version of BASEstar Open, you
   must install the Device Connection component included in this kit.
******************************************************************************

BASEstar Open Development option will be installed

* Do you want to install the Device Connection component (DEComni) [Y]?

    BASEstar Open can be installed on any disk of your system.
    There is no default disk: you must enter a disk name.
    The following disks are available on the system.

MALNAT$DUA0:     Mounted         0  OPENVMS061    697113   359   1
MALNAT$DUA1:     Mounted         0  SWAP          600831     3   1

* Target disk: MALNAT$DUA1:
* BSTROSVV V3.2 root directory [[BSTROSVV032]]:

    You are going to install BSTROSVV V3.2 on

    Target disk:    MALNAT$DUA1:
    Root directory: [BSTROSVV032]

* Is this correct [Y]?
```

```
      Select which products you want to install:

      1 DEComni API
      2) DASTkRUN
      3) DASTk 232
      4) ALL of the above

      ************************************************************

* Enter number(s): [1]:

      ************************************************************
      DEComni API OMNI V3.1-001 is installed on this system.
      It is recommended that the database files be renamed or
      deleted if you do not wish to use the existing database.
      If you choose not to rename or delete the database,the
      existing database will continue to be used.
      ************************************************************

               1. Rename existing database
               2. Continue to use existing database

* Enter a number [2]:

 A directory for Omni Directory Services will be created. You
 need to provide the disk name in which the directory will be
 located. If you do not specify a location, the directory will
 be placed in SYS$SYSDEVICE:[OMNI...].
 Include the colon after the disk name.

* Enter the disk now [ddcu:] :
* Do you want to purge files replaced by this installation [YES]?
* Do you want to run the IVP after the installation [YES]?

************************************************************************

   All the questions regarding the installation of BASEstar Open
   have been answered. The installation will take from 15 to 30 minutes,
   and the IVP about 20 minutes, depending on the CPU type and
   the system workload.

************************************************************************

* Do you want to continue [Y]?
%VMSINSTAL-I-RESTORE, Restoring product save set C ...
%VMSINSTAL-I-SYSDIR, This product creates system directory [SYSHLP.EXAMPLES.OMNI].
%CREATE-I-EXISTS, VMI$COMMON:[SYSHLP.EXAMPLES.OMNI] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.CACHE].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.CACHE] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.EXAMPLES].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.EXAMPLES] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.EXE].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.EXE] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.HELP].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.HELP] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.INCLUDE].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.INCLUDE] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.LIB].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.LIB] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.LOCAL].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.LOCAL] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.ODS.SCRIPTS].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.ODS.SCRIPTS] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.INC31].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.INC31] already exists
%VMSINSTAL-I-SYSDIR, This product creates system disk directory MALNAT$DUA0:[OMNI.LIB31].
%CREATE-I-EXISTS, MALNAT$DUA0:[OMNI.LIB31] already exists
```

```
* If you would like to have DEComni for OpenVMS startup during the system
  startup,  add the following line to SYS$STARTUP:SYSTARTUP_VMS.COM.

      $ @SYS$STARTUP:OMNI_STARTUP.COM FULL

%VMSINSTAL-I-RESTORE, Restoring product save set B ...
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

© DIGITAL EQUIPMENT CORPORATION. 1997. ALL RIGHTS RESERVED.

STARTUP PROCEDURE FOR: DEComni API for OpenVMS VAX (TM) V3.1

Now checking process privileges...

Now checking for the DEComni license...

 %OMNI-I-LICENSE, The following valid license was found -  DECOMNI-API


Now checking that prerequisite software has been started...

Now installing the DEComni sharable images...

Now creating DEComni logicals...

Starting Omni Directory Services now...

Checking for required directories...

Checking for required files ...

ODS Identification information : 3.10.1

Checking cache area ODS:[CACHE]...

ods:[cache]ods_cache_0.dat: Compressed by 15.79% [1 entry]
ods:[cache]ods_cache_1.dat: Compressed by 30.43% [1 entry]
ods:[cache]ods_cache_2.dat: Compressed by 35.71% [3 entries]
ods:[cache]ods_cache_3.dat: Compressed by 15.79% [1 entry]
ods:[cache]ods_cache_6.dat: Compressed by 15.00% [2 entries]
ods:[cache]ods_cache_9.dat: Compressed by 30.43% [1 entry]
ods:[cache]ods_cache_10.dat: Compressed by 30.43% [1 entry]

Checking Schema area ODS:[LOCAL]...

Valid Schema definitions found in the attribute schema database as:

OC ALIASED_NAME CN C L S O OU P_ADDR APC OWNER APT AEQ MAPV SY OBJ_ID BER
 PSAP SSAP TSAP N_ADDR AS_LIST DESCR VMD DOM DEL VN APPSN APROF VERSION NLEVEL
 MAX_CALLED MAX_CALLING MAX_SEG CBB_LIST SS_LIST VENDOR MODEL REV INUM DN
 SHARE CONT_FILE CAP_FILE PN REUSE ARG MON DOM_LIST VAR ATYPE ADDR_T ADDR_V
 PORTNAME STATIONID BAUDRATE CHARLEN PRTYCHECK STOPBIT FLOWCTRL LEV7CONN
 PORTNUMBER NETADDRESS MODEM RSPTIMEOUT RESPTIMER COMPLTIMER MUX NEGOTIATION

Valid Schema definitions found in the object class schema database as:

TOP ALIAS C L O OU AP AE DSA DEVICE TAE OAE AS VMD DOMAIN PI NVAR UNVAR
DTK232 DTKTCP DTKOSI OSAP OSH1

ODS_STARTUP: Normal completion.

Now checking ODS schema definitions...

Checking ODS misc schema definitions ...

Checking ODS vmd schema definitions ...
```

```
Checking ODS domain schema definitions ...

Checking ODS pi schema definitions ...

Checking ODS var schema definitions ...

OMNI_SCHEMA.COM: Normal Completion.

%OMNI-S-STARTUP, DEComni startup successfully completed

    +-----------------------------------------------+
    |                                               |
    |  The DEComni API IVP will now be invoked....  |
    |                                               |
    +-----------------------------------------------+


© DIGITAL EQUIPMENT CORPORATION. 1997. ALL RIGHTS RESERVED.
INSTALLATION VERIFICATION PROCEDURE FOR DEComni API for OpenVMS VAX V3.1

Now checking process quotas...

Now checking process privileges...

Now checking that files have reached target directories...

%OMNI-S-IVP, THE DECOMNI IVP HAS SUCCESSFULLY COMPLETED

© DIGITAL EQUIPMENT CORPORATION. 1997. ALL RIGHTS RESERVED.
SHUTDOWN PROCEDURE FOR: DEComni API for OpenVMS VAX V3.1

Now checking process privileges...

Now removing the DEComni sharable images...

%OMNI-S-STOP, DEComni shutdown has successfully completed.

           +-----------------------------------------------+
           |             Starting the IVP for              |
           |              BASEstar Open V3.2               |
           +-----------------------------------------------+

Logical name BSTR_REALM temporarily defined as IVP

BASEstar Open Server V3.2
Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.
Restricted Rights: Use, duplication, or disclosure by the U.S.
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.
This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.


IVP TERMINATED SUCCESSFULLY
 Installation of BSTROSVV V3.2 completed at 11:53
```

```
 VMSINSTAL procedure done at 11:53

© DIGITAL EQUIPMENT CORPORATION. 1997. ALL RIGHTS RESERVED.
STARTUP PROCEDURE FOR DEComni API for OpenVMS VAX V3.1

Now checking process privileges...

Now checking for the DEComni license...

 %OMNI-I-LICENSE, The following valid license was found -  DECOMNI-API


Now checking that prerequisite software has been started...

Now installing the DEComni shareable images...

Now creating DEComni logicals...

Starting Omni Directory Services now...

Checking for required directories...

Checking for required files ...

ODS Identification information : 3.10.1

Checking cache area ODS:[CACHE]...

ods:[cache]ods_cache_2.dat: Compressed by 5.26% [3 entries]
ods:[cache]ods_cache_3.dat: Compressed by 5.88% [1 entry]
Checking Schema area ODS:[LOCAL]...

Valid Schema definitions found in the attribute schema database as:
OC ALIASED_NAME CN C L S O OU P_ADDR APC OWNER APT AEQ MAPV SY OBJ_ID BER
PSAP SSAP TSAP N_ADDR AS_LIST DESCR VMD DOM DEL VN APPSN APROF VERSION
NLEVEL MAX_CALLED MAX_CALLING MAX_SEG CBB_LIST SS_LIST VENDOR MODEL REV
INUM DN SHARE CONT_FILE CAP_FILE PN REUSE ARG MON DOM_LIST VAR ATYPE ADDR_T
ADDR_V PORTNAME STATIONID BAUDRATE CHARLEN PRTYCHECK STOPBIT FLOWCTRL
LEV7CONN PORTNUMBER NETADDRESS MODEM RSPTIMEOUT RESPTIMER COMPLTIMER MUX
NEGOTIATION

Valid Schema definitions found in the object class schema database as:
TOP ALIAS C L O OU AP AE DSA DEVICE TAE OAE AS VMD DOMAIN PI NVAR UNVAR
DTK232 DTKTCP DTKOSI OSAP OSH1
ODS_STARTUP: Normal completion.

Now checking ODS schema definitions...

Checking ODS misc schema definitions ...


Checking ODS vmd schema definitions ...


Checking ODS domain schema definitions ...


Checking ODS pi schema definitions ...


Checking ODS var schema definitions ...
```

```
OMNI_SCHEMA.COM: Normal Completion.


%OMNI-S-STARTUP, DEComni startup successfully completed


Name of the work directory. You must introduce the path name (according to the
rules of VAX/VMS - .e.g. "DISK:[SUBD1.SUBD2]" ) of an UNEXISTING directory :

BSTR.Spool_Work_Dir:  []
MALNAT$DUA1:[BSTROSVV032.wrk]

Name of the snapshot directory. You must introduce the path name (e.g.
"DISK:[SUBD1.SUBD2]") of a directory. If it doesn't exists, it will be
created by the procedure, otherwise you'll share it with some other
installations.

 WARNING : Snapshot directory cannot be the same as work directory

BSTR.Spool_Snap_Dir:  []
MALNAT$DUA1:[BSTROSVV032.snp]

Name of the host where the primary LNS Server will run:

LNS.Server_Node:  [malnat.mln.dec.com]


Name of the host where the secondary LNS Server will run:

 --- Note that the primary and secondary LNS Server Hosts cannot coincide.

LNS.Server_Node_2:  []

If you are not satisfied with the settings of installation parameters you can
terminate the procedure and run it again:

Do you want to continue or not ? [Y] :
@@@ creating BSTR_ETC:bstrusers_MALNAT.com file


@@@ creating BSTR_ETC:bstradmin_MALNAT.com file


@@@ creating BSTR_ETC:lns_MALNAT.conf file

&&& Warning: Secondary LNS Server not configured

@@@ creating SYS$STARTUP:BSTR$STARTUP.COM;1 file


BASEstar Open Server V3.2

Copyright © Digital Equipment Corporation 1993,1999 All Rights Reserved.

Restricted Rights: Use, duplication, or disclosure by the U.S.
Government is subject to restrictions as set forth in subparagraph
(c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
52.227-14 Alt. III, as applicable.

This software is proprietary to and embodies the confidential
technology of Digital Equipment Corporation. Possession, use, or
copying of this software and media is authorized only pursuant to a
valid written license from DIGITAL or an authorized sublicensor.

  SYSTEM       logged out at  9-APR-1999 12:02:25.47

$)
```

D

Additional Files Created During Installation

This appendix lists the additional files and logical names created during the installation of the BASEstar Open Server software, including its device connection management, library management and utility software. Note hat the directory locations in which the files are placed reflects the default placement of files by each installation procedure.

D.1 Files Created During DCM-Modeled Device Connectivity Installation

Table D–1 displays the files created during the installation of the DCM-Modeled Device Connectivity software. These files are present only when BASEstar Open Server data management, menu interface, and external data support software are installed. Do not delete any of these files.

Table D–1 DCM-Modeled Device Connectivity Installation Files

Directory	Module	Filename
BCC$SYSDATA:	Core	BCC$BCC_MSG.DAT
		BCC$GLOBAL_NAME_FILE.DAT
		BCC$GROUP_FILE.DAT
		BCC$HISTORY_FILE.DAT
		BCC$ITEM_FILE.DAT
		BCC$LOCAL_NAME_FILE.DAT
		BCC$MEMBER_FILE.DAT
		BCC$MSG_SECTION.DAT (Run-time)
		BCC$NAME_SECTION.DAT (Run-time)
		BCC$NODE_FILE.DAT
		BCC$PARAMETER_FILE.DAT
		BCC$RPC_STORE_FILE.DAT
		BCC$SERVER_PORT_FILE.DAT
		BCC$SESSION_FILE.DAT
		BCC$SYSTEM_STARTUP.COM
		BCC$USER_FILE.DAT
		SITE_STARTUP.TEMPLATE
	Data	BCC$DMGT_DATA_SECTION.DAT (Run-time)

(continued on next page)

D.1 Files Created During DCM-Modeled Device Connectivity Installation

Table D–1 (Cont.) DCM-Modeled Device Connectivity Installation Files

Directory	Module	Filename
	Management	BCC$DMGT_EXTDAT_SECTION.DAT (Run-time)
		BCC$DMGT_MSG.DAT
		BCC$DMGT_SECTION.DAT (Run-time)
		BCC$ELEMENT_FILE.DAT
		BCC$EXTMAP_FILE.DAT
		BCC$FIELD_FILE.DAT
		BCC$POINT_FILE.DAT
		BCC$RELATION_FILE.DAT
		BCC$STRUCTURE_FILE.DAT
	Menu	BCC$MENU_FILE.DAT
BCC$SYSTEM:	Core	BCC$CLI.EXE
		BCC$DISPATCHER.EXE
		BCC$GBLSERVER.EXE
		BCC$LOGGER.EXE
		BCC$MONITOR.EXE
		BCC$RPCR.EXE
		BCC$SERVER.EXE
	Data	BCC$DMGTSUP.EXE
	Management	BCC$EVALUATOR.EXE
		BCC$EXTDATSRV.EXE
		BCC$LS_SERVER.EXE
	Menu	BCC$CFG_COLORS.EXE
		BCC$CFG_EXTERNAL_MAP.EXE
		BCC$CFG_KEYS.EXE
		BCC$CFG_POINT.EXE
		BCC$CFG_STRUCTURE.EXE
		BCC$CFG_USER.EXE
		BCC$SHOW_EXTERNAL_MAP.EXE
		BCC$SHOW_POINT.EXE
		BCC$SHOW_STRUCTURE.EXE
		BCC$SHOW_SYSTEM.EXE
		BCC$SHOW_USER.EXE
		BCC$UI.EXE

(continued on next page)

Table D–1 (Cont.) DCM-Modeled Device Connectivity Installation Files

Directory	Module	Filename
BCC$SYSTEM:[.DMQ_EXAMPLE]	Data Management	CANCELNOTIF.C
		CANCELNOTIF.COM
		DATAPROC.C
		DATAPROC.COM
		PUTVALUE.C
		PUTVALUE.COM
SYS$COMMON:[SYSHLP]	Core	BCC$CLI_HELP.HLB
		BCC031.RELEASE_NOTES
	Menu	BCC$USER_HELP.HLB
SYS$COMMON:[SYSLIB]	Core	BCC$$SHR.EXE
		BCC$SHR.EXE
		BCCDEFS.BAS
		BCCDEFS.FOR
		BCCDEFS.H
		BCCDEFS.MAR
		BCCDEFS.PAS
		BCCDEFS.PLI
		BCCDEFS.R32
		BCCDEFS_UPCASE.H
		BCC_.ADA
SYS$COMMON:[SYSMSG]	Core	BCC$MSG.EXE
		BCC$MSG.MSG
SYS$COMMON:[SYSTEST]	Core	BCC$IVP.COM
SYS$SPECIFIC:[SYS$STARTUP]	Core	BCC$STARTUP.COM
BCC$SYSDATA:		DCM_NET_CONFIG.TEMPLATE
		ILAN$CONFIG_FILE.DAT
		ILAN$DEVICE_FILE.DAT
		ILAN$PHYPOINT_FILE.DAT
		ILAN$POLLING_FILE.DAT
		ILAN$SPT_SECTION.DAT (Run-time)
		ILAN$SYSTEM_STARTUP.COM

(continued on next page)

D.1 Files Created During DCM-Modeled Device Connectivity Installation

Table D–1 (Cont.) DCM-Modeled Device Connectivity Installation Files

Directory	Module	Filename
		ILAN$_SECTION.DAT (Run-time)
		DCM_DECDEVICE_CONFIG.TEMPLATE
		DCM_RS232_CONFIG.TEMPLATE
BCC$SYSTEM:		DEC_DEV_SIM.EXE
		ILAN$DEVSRV.EXE
		ILAN$CLI.EXE
		ILAN$SPT_PARAM.COM
	DCM Menu	ILAN$CFG_DEVICE.EXE
		ILAN$CFG_PHYPOINT.EXE
		ILAN$CFG_POLLSET.EXE
		ILAN$DELETE.EXE
		ILAN$DEV_CONTROL.EXE
		ILAN$DEV_DIRECTORY.EXE
		ILAN$DEV_READ_DATA.EXE
		ILAN$DEV_READ_IPOINT.EXE
		ILAN$DEV_STATUS.EXE
		ILAN$DEV_WRITE_DATA.EXE
		ILAN$DEV_WRITE_IPOINT.EXE
		ILAN$DOWNLOAD_FILE.EXE
		ILAN$SHOW_CONNECT.EXE
		ILAN$SHOW_DEVICE.EXE
		ILAN$SHOW_PHYPOINT.EXE
		ILAN$SHOW_POLLSET.EXE
		ILAN$UPLOAD_FILE.EXE
BCC$SYSTEM:[.DCM_EXAMPLE]		ILAN_READ_EXAMPLE.PLI
		ILAN_WRITE_EXAMPLE.PLI
BCC$SYSTEM:[.NCT_EXAMPLE]		(Network DAS example files)
BCC$SYSTEM:[.RS232_EXAMPLE]		(RS-232 DAS example files)
SYS$COMMON:[SYSHLP]		ILAN$DEVICE_SPECIFIC.HLB
		ILAN$HELP.HLB

(continued on next page)

Table D–1 (Cont.) DCM-Modeled Device Connectivity Installation Files

Directory	Module	Filename
SYS$COMMON:[SYSLIB]		BCC$DCM_EAI.EXE
		BCC$DCM_EUI.EXE
		ILAN$CLI_SHR.EXE
		ILAN$SERVER_SHR.EXE
		ILAN$SHR.EXE
		ILANDEFS.ADA
		ILANDEFS.BAS
		ILANDEFS.FOR
		ILANDEFS.H
		ILANDEFS.MAR
		ILANDEFS.PAS
		ILANDEFS.PLI
		ILANDEFS.R32
		ILANDEFS_UPCASE.H
		ILANSUPPORT.ADA
		ILANSUPPORT.BAS
		ILANSUPPORT.FOR
		ILANSUPPORT.H
		ILANSUPPORT.MAR
		ILANSUPPORT.PAS
		ILANSUPPORT.PLI
		ILANSUPPORT.R32
		ILANSUPPORT_UPCASE.H
		ILAN_DEC.EXE
		ILAN_DN.EXE
		ILAN_MIRROR.EXE
		ILAN_NET.EXE
		ILAN_NULL.EXE
		ILAN_RS232.EXE
		ILAN_TCP.EXE
SYS$COMMON:[SYSMSG]		ILAN$MSG.EXE
		ILAN$MSG.MSG
SYS$COMMON:[SYSTEST]		ILAN$IVP.COM
SYS$SPECIFIC:[SYS$STARTUP]		ILAN$STARTUP.COM

D.2 Files Created During BASEstar Open Server Utility Installation

Table D–2 displays the files created during the utilities software installation. Only the utilities selected during the execution of the installation procedure will be installed on the system.

Table D–2 BASEstar Open Server Utility Installation Files

Utility	Directory	Filename
BASEstar Open Server Text Services	BCC$SYSTEM:	BCC$GET_TEXT_FILE.EXE
		BCC$PUT_TEXT_FILE.EXE
BASEGEN Utility	BCC$SYSTEM:	BCC$BASEGEN.COM
Monitor Port Utility	BCC$SYSTEM:	BCC$MONITOR_PORT.EXE
	SYS$HELP	BCC$MONITOR_PORT.HLB
Monitor BASEstar Open Server Utility	BCC$SYSDATA:	BCC$MON_BSTAR_APP.DAT
	BCC$SYSTEM:	BCC$MONITOR_BASESTAR.EXE
	SYS$HELP:	BCC$MONITOR_BASESTAR.HLB
Value Notification Utility	BCC$SYSTEM:	BCC$NOTIFY.EXE
LSE Environment	SYS$LIBRARY:	BASESTAR.ENV

D.3 Files Created During BASEstar CIMfast Installation

Table D–3 lists the directories created under the home directory (SYS$SYSDEVICE:[BCF]) during the BASEstar CIMfast installation procedure, together with a description of the contents of each directory.

Table D–3 BASEstar CIMfast Directories

Directory	Description
[.BLD]	Object files and procedures for linking the product (as occurs during the installation procedure)
[.BLD.BCF]	Basic library
[.BLD.BSTR]	BASEstar Open-specific files
[.BLD.RDB]	SQL support files for DEC Rdb
[.DEMO]	Example files
[.DEMO.CALLABLE_BCF]	Examples of how to use BASEstar CIMfast in various languages (see BCF$README.CBCF)

(continued on next page)

Table D–3 (Cont.) BASEstar CIMfast Directories

Directory	Description
[.DEMO.CBCF_POINTS]	Example of a simple callable BASEstar CIMfast program which prints the name, value, and data type of a Data_Point when it changes
[.DEMO.EXPRESSIONS]	Examples of how to use expressions (see BCF_README.EXPRESSIONS)
[.DEMO.MACROS]	Example of how to use the macro preprocessor (see BCF_README.MACROS)
[.DEMO.POINTS]	Example of how to create Data_Points (see BCF_README.POINTS)
[.DEMO.SQL]	Example of how to use the relational database loaded with Data_Points
[.EXE]	BASEstar CIMfast executable files created during the installation (links during the installation procedure)
[.LIB]	Include files for using the BASEstar CIMfast callable

Table D–4 lists the files created during the BASEstar CIMfast installation procedure. Do not delete any of these files.

Table D–4 BASEstar CIMfast Installation Files

Directory	Filename
SYS$SYSDEVICE:[BSF]	BCF$SETUP.COM
SYS$SYSDEVICE:[BCF.BLD]	BCF$CPP.OLB
	BCF$SHR.LNK
	BCF_ACT_TABLE.OBJ
	BCF_BLD.COM
	BCF_EXP_TABLE.OBJ
	BCF_MAIN.OBJ
	BCF_MAKE_TABLE.COM
	BCF_XXX_TABLE.MAR
	BCF_XXX_TABLE_END_STRUCT.H
	BCF_XXX_TABLE_HEADER.H
	BCF_XXX_TABLE_START_STRUCT.H
	CPP.EXE
SYS$SYSDEVICE:[BCF.BLD.BCF]	BCF$BCF.OLB
	BCF$BCF.OPT
	BCF$BCF_VECT_ALPHA.OPT
	BCF$ID.OPT
	BCF$IMAGES.OPT
	BCF_ACT_NAMES.H
	BCF_EXP_NAMES.H

(continued on next page)

Table D–4 (Cont.) BASEstar CIMfast Installation Files

Directory	Filename
SYS$SYSDEVICE:[BCF.BLD.RDB]	BCF$BSTR.OLB
	BCF_BSTR.OPT
	BSTR_ACT_NAMES.H
	BSTR_EXP_NAMES.H
SYS$SYSDEVICE:[BCF.BLD.RDB]	BCF$RDB.OLB
	BCF$RDB.OPT
	SQL_ACT_FUNC.H
	SQL_ACT_NAMES.H
	SQL_EXP_FUNC.H
	SQL_EXP_NAMES.H
SYS$SYSDEVICE:[BCF.DEMO.CALLABLE_ BCF.DIR]	BCF$CALLBACKS.C
	BCF$CALLBACKS.C
	BCF$CALLBACKS.FOR
	BCF$CALLBACKS.PAS
	BCF$MAKE_C_CBCF.COM
	BCF$MAKE_FOR_CBCF.COM
	BCF$MAKE_PAS_CBCF.COM
	BCF$README.CBCF
	BCF_CALLBACKS.C
	BCF_CBCF.CEL
	BCF_MAKE_CBCF.COM
SYS$SYSDEVICE:[BCF.DEMO.CBCF_POINTS]	BCF$CBCF_POINTS.C
	BCF$CBCF_POINTS.FOR
	BCF$CREATE_POINTS_CBCF_POINTS.COM
	BCF$MAKE_C_CBCF_POINTS.COM
	BCF$MAKE_FOR_CBCF_POINTS.COM
	BCF$README.CBCF_POINTS
	BCF_CBCF_POINTS.C
	BCF_CBCF_POINTS.CEL
	BCF_CREATE_POINTS_CBCF_POINTS.COM
	BCF_MAKE_CBCF_POINTS.COM
SYS$SYSDEVICE:[BCF.DEMO.EXPRESSIONS]	BCF_ARITHMETIC_OPERATOR_DEMO.CEL
	BCF_EQUIVALENCE_OPERATOR_DEMO.CEL
	BCF_LOGICAL_OPERATOR_DEMO.CEL
	BCF_README.EXPRESSIONS
	BCF_RELATIONAL_OPERATOR_DEMO.CEL
	BCF_STATISTIC_OPERATOR_DEMO.CEL

(continued on next page)

Table D–4 (Cont.) BASEstar CIMfast Installation Files

Directory	Filename
	BCF_STRING_OPERATOR_DEMO.CEL
SYS$SYSDEVICE:[BCF.DEMO.MACROS]	BCF_MACRODEFS.CFH
	BCF_MACRO_DEMO.CEL
	BCF_MACRO_DEMO.LIS
	BCF_README.MACROS
SYS$SYSDEVICE:[BCF.DEMO.POINTS]	BCF$CREATE_POINTS_POINTS.COM
	BCF$POINT_SET.CEL
	BCF$README.POINTS
SYS$SYSDEVICE:[BCF.DEMO.SQL]	BCF$CREATE_STRUCT_SQL.COM
	BCF_CREATE_STRUCT_SQL.COM
	BCF_README.SQL
	BCF_SQL.CEL
SYS$SYSDEVICE:[BCF.EXE]	BCF$SHR.EXE
	BCF_MAIN.EXE
SYS$SYSDEVICE:[BCF.LIB]	BCF$CBCF_MAIN.C
	BCF$CBCF_MAIN.OBJ
	BCF$DEFS.FOR
	BCF$DEFS.H
	BCF$DEFS.PAS
	BCF$DEFS.PLI
	BCF$LINK.COM
	BCF$LINK.COM
	BCF$LINK.TEMPLATE
	BCFDEFS.H
	BCF_CBCF_MAIN.C
	BCF_CBCF_MAIN.OBJ

Table D–5 lists the files that are created only if the Language Sensitive Editor (LSE) is installed.

Table D–5 BASEstar CIMfast LSE Files

Directory	Filename
BCF$ROOT:[BCF.LSE]	BCF$CEL.ENV

The BASEstar CIMfast installation procedure also creates the following systemwide logical names:

- BCF$HOME
- BCF$ROOT
- BCF$SHR

D.4 Systemwide Logical Names Created by the Device Connection Manager

Table D–6 Logical Names Created by the Device Connection Manager Installation Procedures

Component	System Logical Name
Kernel	BCC$SYSDATA
	BCC$SYSTEM
DCM	ILAN$SYSTEM

E

Device Connection Manager Installation Messages

This section describes the installation message formats and the messages you may receive during the BASEstar installation, and provides a description of recovery procedures for each error.

E.1 Message Format

BASEstar installation messages display in the following format:

```
%FACILITY-L-IDENT, TEXT
```

where:

FACILITY	The facility or component name.
L	The severity indicator codes including: • S for success. The procedure generating the condition value completed successfully. • I for informational. The procedure generating the condition value successfully completed, but has some further explanatory information. • W for warning. A procedure returns data or completes an operation but the results might be incomplete or not what you expected. • E for error. An error has occurred. The operation was attempted but could not be completed. Any returned information should be regarded as invalid. • F for severe error. A severe error occurred and the attempted operation was not complete. Any information returned is invalid and further processing will be incorrect.
IDENT	An abbreviation of the message text. The message descriptions are alphabetized according to this abbreviation.
TEXT	The explanation of the message.

E.2 Device Connection Manager Message Listing

The installation messages are in alphabetical order by the IDENT abbreviation.

BADBCC, The installed version of BASEstar is obsolete.

Explanation: Error. An old version of BASEstar software is installed.

User Action: Install BASEstar Classic, Version 3.2 before installing this distribution kit.

BADBCCVERS, Incompatible installed version of BASEstar. This kit only upgrades BASEstar Classic, Version 3.1.

Explanation: Error. The distribution kit being installed can only be used to update BASEstar Classic, Version 3.1.

User Action: Upgrade your system to Version 3.1 first, then install Version 3.2. Contact your COMPAQ sales representative for assistance if you do not have Version 3.1.

BADDCMVERS, Incompatible installed version of DCM. This kit only upgrades DCM Version 3.1.

Explanation: Error. The distribution kit being installed can only be used to update DCM, Version 3.1.

User Action: Upgrade your system to Version 3.1 of DCM first, then install Version 3.2. Contact your COMPAQ sales representative for assistance if you do not have Version 3.1 BASEstar Classic.

BADDISK, The system root for this node does not exist on the system disk.

Explanation: Error. During a cluster installation, the VMS system disk of the specified node was found to be a disk separate from the VMS disk of the node executing the VMSINSTAL procedure.

User Action: The cluster installation procedure will only execute on nodes that share a common VMS system disk. Nodes that do not share a common system disk must be installed individually, using the single node installation option.

BADID, Invalid DECnet ID, please try again.

Explanation: Error. You entered an incorrect DECnet ID.

User Action: The DECnet ID for a system can be obtained by using the VMS Network Control Program. Refer to the *VMS Network Control Program Manual.*

BADVMS, You must be running at least V5.3 of VMS.

Explanation: Error. The system is not running VMS Version 5.3 or higher.

User Action: Upgrade the system to VMS V5.3 or higher.

BCCRUNNING, BASEstar cannot be running during an installation.

Explanation: Error. The installation is going to replace BASEstar images and possibly convert data files. Therefore, BASEstar cannot be running during an installation.

User Action: Stop BASEstar software on the node.

FAIL, file check failed

FAIL, installed image check failed

FAIL, parameter check failed

Explanation: Error. The Installation Verification Procedure (IVP) failed.

User Action: Review the installation log file for errors. Make any corrections necessary, then restart the installation procedure.

INSFBLOCKS, Insufficient system disk free blocks for installation.

Explanation: Error. There are not enough free blocks on the system disk for this installation.

User Action: The *BASEstar Classic Installation Guide* lists the amount of free blocks required on the system disk for a successful installation. Create enough free space on the system disk and restart the installation.

INSFCTLPAGES, Installation requires the system parameter CTLPAGES to be at least nnn.

Explanation: Error. The value of SYSGEN parameter CTLPAGES is not sufficient to execute this installation procedure.

User Action: Refer to the *BASEstar Classic Installation Guide* for information on setting this parameter.

INSFGBLPAGFIL, Installation requires the system parameter GBLPAGFIL to be at least nnn.

Explanation: Error. The value of SYSGEN parameter GBLPAGFIL is not sufficient to execute this installation.

User Action: Refer to the *BASEstar Classic Installation Guide* for information on setting this parameter.

MISSING, The DCM data file BCC$SYSDATA:FILENAME.DAT is missing.

Explanation: Warning. The installation procedure cannot find a required data file from the previously installed version of BASEstar Device Connection Management.

User Action: The DCM data files will be reinitialized, automatically. Try to locate and restore the missing data file from your disk BACKUP save set.

NEEDBCC032, This kit requires BASEstar V3.2.

Explanation: Informational. BASEstar Classic, Version 3.2 is not installed.

User Action: Install BASEstar Classic, Version 3.2.

NEEDDIR, Unable to locate directory for new ILAN files.

Explanation: Error. The BASEstar directories could not be found.

User Action: Ensure that BASEstar software is installed and that the startup procedure SYS$STARTUP:BCC$STARTUP.COM has been executed to define the correct logical names.

NEEDLICENSE, A BASEstar license is required for installation.

Explanation: Error. The BASEstar license was not installed.

User Action: Install the correct BASEstar license before attempting to install BASEstar software.

NOALTROOT, BASEstar cannot be installed in an alternate system root.

Explanation: Error. VMSINSTAL.COM was invoked with OPTIONS R, which requests the software be installed in an alternate VMS system root.

User Action: Restart the installation without this option.

NODECNET, DECnet node name cannot be translated.

Explanation: Error. The node name specified cannot be found in the DECnet database.

User Action: Define the node name and restart the installation.

NODIR, The logical name BCC$SYSTEM or BCC$SYSDATA is not defined on the installed system.

Explanation: Error. The logical names for the BASEstar directories are not defined.

User Action: Ensure that the BASEstar startup procedure SYS$STARTUP:BCC$SYSTARTUP.COM has been executed before installing this kit.

NOIDENT, The BASEstar Identifier BCC$USER does not exist.

Explanation: Error. The VMS Identifier BCC$USER was not found in the VMS User Authorization File.

User Action: This identifier is created during the BASEstar installation procedure. Reinstall BASEstar, or manually create the identifier by using the VMS Authorize Utility.

NOLIBLOG, The logical name BCC$LIBRARY_LOCATION must be defined for re-installation.

Explanation: Error. The library management component has been previously installed on this system, but the logical name BCC$LIBRARY_ LOCATION is not defined.

User Action: Execute the library management procedure SYS$STARTUP:BCCLIB$STARTUP.COM to define the logical name.

NOMENUFILE, Menu file BCC$MENU_FILE.DAT was not found in BCC$SYSDATA: .

Explanation: Error. The BASEstar data file for menu definitions is missing.

User Action: Execute the main BASEstar installation procedure and select the menu interface component to reinstall the menu interface.

NOSITLOG, The logical name SIT$KNL_OSP must be defined for re-installation.

Explanation: Error. The library management component has been previously installed on this system, but the logical name SIT$KNL_OSP is not defined.

User Action: Execute the library management procedure SYS$STARTUP:BCCLIB$STARTUP.COM to define this logical name.

NOSMISERVER, Process SMISERVER is not running on this node.

Explanation: Error. The VMS SYSMAN server process is not running on this node. SYSMAN is used by the cluster installation procedure to install BASEstar components on multiple nodes.

User Action: See the *VMS SYSMAN Utility Manual* for information on starting SYSMAN. (Single-node installations do not use the VMS SYSMAN Utility.)

NOTCLUSMEM, This node is not a cluster member.

Explanation: Error. The nodename specified is not a member of the cluster on which the installation is executing, or the nodename entered has an error.

User Action: Look closely at the nodename entry and reenter the nodename if a typing or syntax error was made. If the nodename was correct and you want to install software on that node, abort the current VMSINSTAL and log out, log in to the desired node, and restart the installation there.

NOVAXFMS, VAX FMS V2.4 is not installed on this system.

Explanation: Warning. The BASEstar menu system is selected for installation, but FMS Version 2.4 (or Version 2.3) is not present on the system.

User Action: Install VAX FMS Version 2.4 or Version 2.3. The "Forms Driver" run-time-only component of FMS, FMSFDV, is sufficient.

OLDFMS, The installed version of VAX FMS is obsolete.

Explanation: Error. An old version of VAX FMS is present on the system. The BASEstar menu system requires VAX FMS Version 2.4 or 2.3.

User Action: Install VAX FMS Version 2.4 or 2.3.

REINIT, All DCM data files will be replaced.

Explanation: Warning. One or more of the data files is missing; device connection management will re-create all data files.

User Action: None.

STARTMENU, BASEstar Menu System must be installed.

Explanation: Error. The BASEstar Menu System is not installed on this system. It is required for the Library management component.

User Action: Execute the main BASEstar installation procedure and select the menu interface component.

SYSTEMACCT, BASEstar must be installed from the SYSTEM account.

Explanation: Error. The installation procedure will only execute from the VMS SYSTEM account.

User Action: Log in to the VMS SYSTEM account before installing BASEstar.

WHERESBASESTAR, The logical name BCC$SYSDATA is not defined.

WHERESBASESTAR, The logical name BCC$SYSTEM is not defined.

Explanation: Error. One or both of the BASEstar system logicals are not defined.

User Action: Ensure that BASEstar software is installed and that the startup procedure SYS$STARTUP:BCC$STARTUP.COM has been executed to define the logical names.

F

Sample Installation Log File of the Device Connection Manager

This appendix contains a complete example log of an installation of DCM on OpenVMS Alpha and OpenVMS VAX.

```
$ @SYS$UPDATE:VMSINSTAL   DCM032   ddcu:

VAX/VMS Software Product Installation Procedure V5.5-1

It is 20-OCT-1993 at hh:mm.

Enter a question mark (?) at any time for help.

%VMSINSTAL-W-ACTIVE, The following processes are still active:
        DECW$WM_1
        DECW$TE_1
* Do you want to continue anyway [NO]? Y
* Are you satisfied with the backup of your system disk [YES]?

The following products will be processed:

  BCCM V3.2

        Beginning installation of BCCM V3.2 at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.

 BASEstar Reduced Kernel V3.2 Installation Procedure
 ---------------------------------------------------

    BASEstar requires one of these licenses to be registered
    and loaded on all nodes on which BASEstar will run:

             Product                   Producer    Version    Release Date
             ---------------           --------    -------    ------------
        [1]  BSTROPEN-DEVELOP          DEC         2.0        16-JUL-1993
        [2]  BSTROPEN-RUNTIME          DEC         2.0        16-JUL-1993
        [3]  BSTROPEN-DEVELOP-USER     DEC         2.0        16-JUL-1993
        [4]  BSTROPEN-RUNTIME-USER     DEC         2.0        16-JUL-1993
        [5]  BASESTAR-DCM              DEC         3.2        16-JUL-1993

* Which of the above licenses will be used? [1]: 1

        Product:      BSTROPEN-DEVELOP
        Producer:     DEC
        Version:      2.0
        Release Date: 16-JUL-1993

* Does this product have an authorization key registered and loaded? Y
  ... Done.
SCSNODE and SCSSYSTEMID checks passed.

The DECnet node ID found for this node is:
    47743 (46.639)
```

Sample Installation Log File of the Device Connection Manager

```
    This node is not a member of a multi-node VAXcluster.

    A single-node installation will be performed.

_____________________________________
Processing node RAINEE::
_____________________________________

    This is a new BASEstar Reduced Kernel installation.
    Only the BASEstar CORE will be installed.

    A location  must  be  chosen  for the BCC$SPECIFIC directory.
    By default, BASEstar data and command files are installed in
    SYS$SPECIFIC:[BCC$SPECIFIC].    This  specification  may  be
    overridden if desired.

* Enter the BASEstar specific directory specification
     [SYS$SPECIFIC:[BCC$SPECIFIC]]:
  The directory entered is:  SYS$SPECIFIC:[BCC$SPECIFIC]
* Is this correct? Y

    A location must be chosen for the BCC$COMMON directory.
    By default, BASEstar executable files are installed in
    SYS$COMMON:[BCC$COMMON].   This  specification  may be
    overridden if desired.

* Enter the BASEstar common directory specification [SYS$COMMON:[BCC$COMMON]]:
  The directory entered is:  SYS$COMMON:[BCC$COMMON]
* Is this correct?) Y
_____________________________________

    The installation procedure  has  no further questions to ask and
    can  continue  from  this  point without user assistance.    The
    installation  time  will  vary  depending on the number of nodes
    selected  for  installation  and  the type of processor on which
    the installation is being done.

_____________________________________
hh:mm

    This installation creates a VMS rights identifier named BCC$USER,
    which is used to control access to BASEstar.

    After the installation is finished, the VMS Authorize Utility
    must be used to grant this identifier to all accounts that will
    use BASEstar.

    ------------------------------------------------------------
    A   R E M I N D E R   T O   T H E   I N S T A L L E R

    Each VMS user account that will issue BSTAR commands must be
    granted the BCC$USER rights identifier, as well as having
    certain Authorized privileges set.

    >> Remember to grant the BCC$USER identifier to the SYSTEM account
    >> (and/or any other accounts you will be using to start the BASEstar
    >> system) and to check the Authorized privileges of those accounts
    >> BEFORE YOU ATTEMPT TO START THE BASESTAR SYSTEM.
    ------------------------------------------------------------

%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named BCC$USER.
%UAF-I-RDBADDMSG, identifier BCC$USER value: %X80010003 added to rights data base

hh:mm:ss
_____________________________________________________
Installing BASEstar Reduced Kernel on node RAINEE:: ...
_____________________________________________________
```

Sample Installation Log File of the Device Connection Manager

```
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
VMI$ROOT:[BCC$COMMON].
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
SYS$SYSDEVICE:[SYS0.BCC$SPECIFIC].

hh:mm:ss
%BCCM-I-STARTCORE, BASEstar CORE installation beginning on RAINEE:: .

    This  installation  places  a  file  named BCC$STARTUP.COM in the
    directory  SYS$SPECIFIC:[SYS$STARTUP]  in  each  system  root  on
    which this installation is being performed.  The SYSTARTUP_V5.COM
    file should be edited to invoke this startup file when the system
    boots.

hh:mm:ss
hh:mm:ss
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

BASEstar Reduced Kernel V3.2 Installation Verification Procedure (IVP)

$ DEFINE/NOLOG/PROCESS  BCC$SYSDATA  SYS$SPECIFIC:[BCC$SPECIFIC]
$ DEFINE/NOLOG/PROCESS  BCC$SYSTEM   VMI$ROOT:[BCC$COMMON]
$ set noverify
%BCCIVP-I-CORESTART, CORE verification starting on node RAINEE.

%BCCIVP-I-PASS, CORE data files check passed.
%BCCIVP-I-PASS, CORE command files check passed.
%BCCIVP-I-PASS, CORE executable files check passed.
%BCCIVP-I-PASS, CORE support files check passed.
%BCCIVP-I-PASS, CORE system files check passed.
%BCCIVP-I-PASS, CORE installed image check passed.
BASEstar Reduced Kernel V3.2 IVP completed successfully.

 Installation of BCCM V3.2 completed at hh:mm

Enter the products to be processed from the next distribution volume set.
* Products:

        VMSINSTAL procedure done at hh:mm
```

G

BASEstar Open Global Variables

This appendix describes the BASEstar Open global variables. BASEstar Open global variables provide application and environment components with a generalized mechanism for the exchange of information.

BASEstar Open global variables are DCL logical names that you can set by means of standard OpenVMS features. For details, refer to the DCL documentation.

G.1 Global Variable List

Table G–1 lists all the BASEstar Open global variables. For ease of reference, the global variables are grouped according to the time at which they are set/used.

Table G–1 BASEstar Open Global Variables

Variable Name	Description
Installation-specific global variables	
BSTR_BIN	The absolute pathname of the `bin` directory where the BASEstar Open executables are stored.
BSTR_DBACCESS_KEY	The root directory where BASEstar Open stores snapshot configuration files created on the PODB Node. The default value is *disk*:[bstr_spool.*node_name*.snapshot].
BSTR_ETC	The absolute pathname of the `etc` directory where the BASEstar Open environment commands and command file are stored.
BSTR_INCLUDE	The absolute pathname of the `include` directory where BASEstar Open stores the API include files.
BSTR_LIB	The absolute pathname of the `lib` directory where BASEstar Open stores the API library.
BSTR_LOG	The absolute pathname of the directory in which BASEstar Open stores the Node and Realm log files.
BSTR_LOG_NODE	The BASEstar Open Node log file.
BSTR_LOG_REALM	The BASEstar Open Realm log file (exists only if BSTR_ REALM has been defined).
BSTR_ROOT	The absolute pathname of the directory in which BASEstar Open is installed.
BSTR_WORK_ROOT	The absolute pathname of the root directory of the subtree where BASEstar Open stores temporary files at run-time. The default value is *disk*:[bstr_spool.*node_name*.work].

(continued on next page)

Table G–1 (Cont.) BASEstar Open Global Variables

Variable Name	Description
Global variables for activating Event Services and Data & Device Services servers	
BSTR_ACTIVITY	Name of the Activity to be started within the server (string of up to 32 characters, in accordance with BASEstar Open local name syntax). If you specify more than one name, use blanks to separate each name.
BSTR_DATABASE	The type of database. Set this qualifier to SNAPSHOT.
BSTR_DBVERSION	The major identifier of the snapshot configuration to be loaded. If the BSTR_DBVERSION global variable is not set, the most recent one (having the highest major identifier) is assumed by default.
BSTR_DDM_SERVER	
BSTR_DEVICE_GEN_TH_SIZE	The DEVICE general thread stack size.
BSTR_DEVICE_I_SIZE	The DEVICE management initial memory pool.
BSTR_DOMAIN	The name of the resources handled by an Activity. This global variable must only be set for Event Services, Packet Services and Data & Device Services servers, as follows: • PACKET. The global variable must contain the full name of a Domain object. When BASEstar Open executes the Packet Services server, the associated Activity execution makes the Packet Services objects created within the specified Domain available to the clients. • EVENT. The global variable must contain the full name of a Domain object. When BASEstar Open executes the Event Services server, the associated Activity execution makes the Event Services objects created within the specified Domain available to the clients. • DATA. The global variable must contain the full name of a Domain object. When BASEstar Open executes the Data & Device Services server, the associated Activity execution makes the Data & Device Services objects created within the specified Domain available to the clients.
BSTR_EVM_SERVER	
BSTR_LOCAL_VMD_NAME	The name of the VMD definition to be used as the calling VMD when accessing a remote VMD. If specified for a Device Services server, this VMD represents the calling VMD used by the actual Device Services server to reach the device VMD. When specified for a Data Services server or a CLI command interpreter instance, this VMD represents the calling VMD used by the actual Data Services server (or CLI instance) to reach the device VMD or Device Services server VMD.

(continued on next page)

Table G–1 (Cont.) BASEstar Open Global Variables

Variable Name	Description
Global variables for activating Event Services and Data & Device Services servers	
BSTR_POLLING_FACTOR	The time scale factor for Polling_Sets.
BSTR_RETRY_CONN_TIME	The value assigned to the BSTR_RETRY_CONN_TIME variable specifies the number of seconds that the server waits before attempting to restart the association with the physical device after a connection failure. The default value is 30 seconds.
BSTR_SERVERS_PFACTOR	Determines whether BASEstar Open performs the following operations in parallel or sequential mode: • Execution/termination of Activities • Connection/disconnection of clients. Select a value between 1 and 32 (default 1); the higher the number, the greater is the degree of sequentiality.
BSTR_STAT_ENABLED	If this global variable is set, the server enables statistics collection for each object made available (provided that the object supports statistics collection). If this global variable is not defined when a BASEstar Open-supplied server is started up, BASEstar Open disables statistics collection for that server.
BSTR_VMD	This qualifier is only significant for a Device Services server and must contain the name of one or more Device Services VMDs. If you specify more than one name, use blank spaces to separate them.
Realm global variable	
BSTR_REALM	The local name of the Realm that BASEstar Open sets as the current Realm.
Global variables for activating the Global Object Services server	
BSTR_GOM_MAX_CLIENT	The maximum number of client application components supported by the Global Object Services server
Application Management Services global variables	
BSTR_DIRECTORY	The pathname specified in the directory attribute of the Node on which the Program execution is running.
BSTR_PROGRAM	The full name of the associated Program.
Global variables for activating the Trace Database	
BSTR_TRC_PATH	The pathname of the directory in which the Trace Database resides.

(continued on next page)

Table G–1 (Cont.) BASEstar Open Global Variables

Variable Name	Description
	Miscellaneous
BSTR_CONN_MODE	The local VMD connection behavior
BSTR_DADE_STACT_TH_SIZE	Activity control thread stack size

G.2 Installation-specific Global Variables

The *bstr_node_setup* command creates the appropriate command files that contain the commands used to set the installation-specific global variables. The values of the global variables depend on the values you specified when installing BASEstar Open and executing the *bstr_node_setup* command. To set the values of the global variables for a user, you must execute the appropriate command file which is stored in the `/etc` directory.

G.3 BSTR_REALM Global Variable

The BSTR_REALM global variable contains the name of the Realm on which you wish to perform operations. Assign a valid Realm name to this global variable as required (before activating the BASEstar Open CLI or a BASEstar Open application from the command interpreter).

You can enter the following command line at the DCL prompt:

```
DCL> DEFINE BSTR_REALM MY_REALM
```

G.4 Global Variables for Server Activation

Before activating an Event Services or Data & Device Services server from the command interpreter, you must set the global variables to the appropriate values, as described in Chapter 12. Note that if you activate a server via Application Management Services, BASEstar Open sets the values for these variables automatically.

G.5 Application Management Services Global Variables

The *bstr_initialize* function returns the BSTR_PROGRAM and BSTR_DIRECTORY global variables when BASEstar Open activates a user-written application via the Application Management Services. Appendix H explains how to set one or more user-defined global variables that Application Management Services make available to the activated Program execution.

H

Environment and Parameter Files

This appendix describes the format and contents of the following files:

- The environment file whose pathname is specified in the *environment* attribute of a Node object.
- The parameters file whose pathname is specified in the *parameters* attribute of a Program object.

These files are optional; if you do not create them, Application Management Services set up the environment using default values. You can also use an ASCII editor to create the files.

H.1 General Record Format

Each record in the environment or parameter file must be terminated by pressing the Return key. If one or more blank characters are left at the beginning of a line, the entire line (up to Return) is assumed to be a comment. The record format follows the syntax shown below:

```
object.item:value
```

where:

- *object* represents:
  - The full name of a Program in a Program parameter file. This means that you can specify parameters relating to different Programs in the same file.
  - The local name of a Node object in a Node environment file. This means that you can specify parameters relating to different Nodes in the same file.
- *item* represents a valid item identifier (according to the type of file).
- *value* represents the value of *item*.

H.2 Format of the File for the Node *environment* Attribute

This section describes the format and the contents of the file whose pathname can be specified in the *environment* attribute of a Node object. This file contains the value for one or more global variables. The Application Management Services server sets the values for these global variables at Node startup time for all the processes activated on the Node that run the image associated with any Program. You can use these global variables to:

- Express the pathnames set as values of the Program *parameters* and *image* attributes
- Set the values for the input, output and error items of a Program parameter file

Table H–1 describes the valid keywords for *item*.

Table H–1 Valid Values of the File for the Node *environment* Attribute

Item Identifier	Type	Description
env	string	Each *env* argument identifies a BASEstar Open global variable, and the value to be assigned to it. You can define zero, one, or as many items as there are global variables to be set. The *value* field must be in the form "*NAME*=*VALUE*", where *NAME* is the name of the global variable, and *VALUE* is the value to be assigned to it. BASEstar Open global variables are set in the order in which they are encountered in the file. These logical names are always defined in the LNM$PROCESS_TABLE.

Below is an example of a Node environment file. The information contained in this file causes the Application Management Services to define the MY_ DIRECTORY and MY_TMP logical names before it activates any of the Program executions on NODE1 or NODE2.

```
NODE1.env:MY_DIRECTORY=$1$DUA1:[USERS.FOO]
NODE1.env:MY_TMP=$1DUA1:[USERS.FOO.NODE1.TMP]
NODE2.env:MY_DIRECTORY=$1$DUA1:[USERS.FOO]
NODE2.env:MY_TMP=$1DUA1:[USERS.FOO.NODE2.TMP]
```

H.3 Format of the File for the Program *parameters* Attribute

As specified in the Program *parameters* attribute, this file contains information used by the Application Management Services during activation of the process in which the Program image is run.

If you do not specify this file, or if it does not contain settings for all the parameters to be used, BASEstar Open assumes the default values.

Note

A value specified for a logical name in the Program parameter file overrides a value set for the same logical name in the Node environment file.

H.3.1 File Format

Table H–2 describes the valid values for *item*.

Table H–2 Valid Values of the File for the Program *parameters* Attribute

Item Identifier	Type	Description
input	pathname	Input file. Command procedure executed before running the Program image
output	pathname	Output/error file. Default value is "NL:" (Null Device)
prcnam	string	Process name. Default value is the name of the related program object truncated to the first 15 characters
baspri	integer	Base priority[1]. Default value is 4
uic	string	UIC in the format [GROUP,MEMBER], where GROUP and MEMBER can be a name or a number
prv	strings	Privileges. Valid only if the *flag* item is set to NOUAF. List of privileges keywords separated by blank characters. The list of privileges and their explanation can be found in the *VMS DCL Concepts Manual - Table A-1*. The default privileges for a Program execution started up by the Application Management Services server are TMPMBX, NETMBX and GRPNAM. The ALL keyword is allowed.
astlm	integer	AST limit. Valid only if the *flag* item is set to NOUAF.
biolm	integer	Buffered I/O limit. Valid only if the *flag* item is set to NOUAF.
bytlm	integer	Buffered I/O byte count quota. Valid only if the *flag* item is set to NOUAF.
cpulm	integer	CPU time limit. Valid only if the *flag* item is set to NOUAF.
diolm	integer	Direct I/O quota. Valid only if the *flag* item is set to NOUAF.
enqlm	integer	Lock request quota. Valid only if the *flag* item is set to NOUAF.
fillm	integer	Open file quota. Valid only if the *flag* item is set to NOUAF.
jtquota	integer	Job table quota. Valid only if the *flag* item is set to NOUAF.
pgflquota	integer	Paging file quota. Valid only if the *flag* item is set to NOUAF.
prclm	integer	Subprocess quota. Valid only if the *flag* item is set to NOUAF.
tqelm	integer	Timer queue entry quota. Valid only if the *flag* item is set to NOUAF.
wsdefault	integer	Default working set size. Valid only if the *flag* item is set to NOUAF.
wsextent	integer	Working set expansion quota. Valid only if the *flag* item is set to NOUAF.
wsquota	integer	Working set size quota. Valid only if the *flag* item is set to NOUAF.

[1] You can specify a priority value different from the default one only if the "Realm user" who executes the *bstr_realm_start_node* command has the ALTPRI privilege.

(continued on next page)

Table H–2 (Cont.) Valid Values of the File for the Program *parameters* Attribute

Item Identifier	Type	Description
flag	string	Process flag specifying which privileges and quotas are assigned to the Program execution. If specified, it must be set to the NOUAF value. If *flag* is not specified, any privileges and quotas specified in the parameter file are ignored. Before running the Program execution, the login.com command procedure of the "Realm user" is executed and the Program execution is assigned the quotas and the privileges associated with the "Realm user" (see Section 12.3.1.1 for details). If *flag* is specified, the Program execution is assigned the quotas and the privileges specified in the parameter file. If the parameter file does not contain a value for the privileges, then TMPMBX, GRPNAM and NETMBX are assumed by default. If the parameter file does not contain a value for a given quota (or contains a zero value), then the quota is assigned the system default value. You obtain default values by running SYSGEN to display the corresponding SYSGEN parameter. Default values in SYSGEN are named PQL_D*xxxxx* where *xxxxx* is the quota name. For more information about default quotas see *VMS System Services Reference Manual - SYS$CREPRC*.
env	string	Each *env* argument identifies a logical name and the equivalent name to be assigned to it. You can define zero, one, or as many items as there are logical name to be set in the process table. The *string* field must be in the form *"NAME=VALUE"*, where *NAME* is the logical name, and *VALUE* is the equivalent name to be assigned to it. Logical names are defined in the order in which they are encountered in the file.
arg	string	If the *Program Image* is an executable file, each *arg* argument identifies the value to be assigned to an element program argument vector (argv[ ]). These arguments are passed to the program being run in the order in which they are encountered in the file. The first specified argument corresponds to argv[1], the second to argv[2], and so on. If the *Program Image* is a command procedure, each *arg* argument identifies the value to be assigned to parameters *P1-P8*. The first specified argument corresponds to P1, the second to P2, and so on.

In the example that follows, the *flag* item is not set. (Quotas and privileges, if specified in the file, are ignored.)

Example H–1 Program Parameter File (*flag* not set)

```
/PROGRAM1.output: DISK1:[USER.TMP]PROGRAM1.OUT
/PROGRAM1.env: MY_BIN=DISK1:[USER.BIN]
/PROGRAM1.env: MY_TMP=DISK1:[USER.TMP]
/PROGRAM1.baspri: 6
```

In the example that follows, the *flag* item is set to NOUAF.

Example H–2 Program Parameter File (*flag* set to NOUAF)

```
/PROGRAM1.flag: NOUAF
/PROGRAM1.input: DISK1:[USER.COM]PROGRAM1.COM
/PROGRAM1.prcnam: PRG1
/PROGRAM1.prv: SHARE SYSPRV SYSLCK
/PROGRAM1.astlm: 200
/PROGRAM1.biolm: 200
/PROGRAM1.bytlm: 100000
/PROGRAM1.cpulm: 0
/PROGRAM1.diolm: 200
/PROGRAM1.enqlm: 1000
/PROGRAM1.fillm: 100
/PROGRAM1.jtquota: 600
/PROGRAM1.pgflquota: 80000
/PROGRAM1.prclm: 30
/PROGRAM1.tqelm: 200
/PROGRAM1.wsdefault: 1024
/PROGRAM1.wsextent: 1024
/PROGRAM1.wsquota: 1024
/PROGRAM1.arg: 1
/PROGRAM1.arg: 2
/PROGRAM1.env: MY_BIN=DISK1:[USER.BIN]
/PROGRAM1.env: MY_TMP=DISK1:[USER.TMP]
```

I

Environment Component Processes

This appendix provides information about the platform-dependent processes that implement each BASEstar Open environment component. The main purpose of this appendix is to enable a system administrator to monitor (using standard tools) the processes that implement environment components (activated by environment commands in transparent mode).

Note that BASEstar Open:

- Activates the processes that implement Node-specific components when you execute the *bstr_node_start* command
- Activates the processes that implement Realm-specific components when you execute the *bstr_realm_start_node* command.

For a description of the environment components and commands, refer to Chapter 8 and Chapter 10, respectively.

Environment Component Processes

Table I–1 lists the processes that implement the Node- and Realm-specific environment components on an OpenVMS Node.

Table I–1 Processes

Process Description	Process Name	Notes
Node-Specific Environment Components		
Log Services server	LOG_SERVER	
Watchdog	WATCHD_<*port*>	Where <*port*> is the number of the TCP/IP port used by the Watchdog
Name Service Client	LNSHM_<*name_space*>	Where *name_space* is the name space in which the Name Service server creates objects
Name Service Server	LNSSR_<*name_space*>	Where *name_space* is the name space in which the Name Service server creates objects This process is only activated if the local Node has been configured (during the *bstr_node_setup* phase) as primary or secondary LNS Server
Realm-Specific Environment Components		
Communication Service Enabler	*CE_*<*realm*>	Where <*realm*> is the name of the associated Realm
Communication Service Name Server	*CNS_*<*ce_pid*>	Where *ce_pid* is the process identifier of the *CE_*<*realm*> process
Communication Service Remote Communication Manager for TCP/IP	RCMTCP_<*itself_pid*>	Where <*itself_pid*> is the process identifier of the Remote Communication Manager for TCP/IP process itself
Global Objects Services server	GOM_<*realm*>	Where *realm* is the name of the associated Realm
Application Management Services server	AMS_<*realm*>	Where *realm* is the name of the associated Realm
Database Service server	*DB_*<*realm*>	Where *realm* is the name of the associated Realm Only on PODB Nodes, if the *bstr_realm_startup_db* command has been executed
PC Communication server	*PC_*<*realm*>	Where *realm* is the name of the associated Realm

J

Managing Snapshot Files in a Distributed Environment

You can only create snapshots on the PODB Node of a Realm. However, if you wish to perform operations on a Realm of a multi-node and distributed BASEstar Open environment, you must identify the snapshot files for each server Node on which the Realm is to be started up, and make them accessible to the BASEstar Open-supplied servers.

The information in this appendix is only addressed to users who are running BASEstar Open in a multi-node distributed environment. Note that there is no need to make snapshot files accessible from a client Node. This appendix describes:

- The location of snapshot files and directories in a PODB Node
- Platform-dependent tools for accessing snapshot files from Nodes other than the PODB Node.

J.1 Using Configuration Management Commands in a Multi-Node Environment

You can use the following CLI commands to create and manage snapshot files:

- GENERATE SNAPSHOT
- DUMP SNAPSHOT
- GET INFO SNAPSHOT
- PURGE SNAPSHOT
- RESET SNAPSHOT
- UPGRADE SNAPSHOT

For further information, refer to the *BASEstar Open Command Language Interface*.

It is important to specify that, for a given Realm, any user active on any Node can execute snapshot management CLI commands

J.2 Snapshot File Location

Snapshot files are stored in the following directory:

`BSTR_DBACCESS_KEY:[`*`realm_name`*`.version_`*`version`*`]`

where:

- BSTR_DBACCESS_KEY is the global variable that specifies the root directory where BASEstar Open stores snapshot configuration files created on the PODB Node.
- The *realm_name* subdirectory is the directory where snapshot files are created for the corresponding Realm.
- There is a version_*version* subdirectory for each snapshot configuration version that has been created, where *version* is a three-digit version number padded with zeroes.

For example, directory $BSTR_DBACCESS_KEY/my_realm/version_002 contains the snapshot files for version 2 of the *my_realm* Realm.

Note also that:

- The `$BSTR_DBACCESS_KEY/`*`realm_name`* directory is created on a Node when the *bstr_realm_setup_node* has been executed on that Node for the *realm_name* Realm.
- A *version* subdirectory is created each time the GENERATE SNAPSHOT command is executed for the Realm.

J.3 The BSTR_DBACCESS_KEY Global Variable

The examples described in this appendix assume that the $BSTR_DBACCESS_KEY global variable has been set to its default value. See Appendix G for details.

J.4 Making Snapshot Files Accessible

Two options are available to make snapshot files accessible:

- By installing the snapshot directory on a shared disk
- By copying the snapshot files on each Node that requires them.

J.4.1 Installing Snapshot Directories on a Shared Disk

There must be only one copy of the snapshot files, and it is the original copy created by the snapshot CLI commands under the snapshot directories of the PODB Node, as explained in Section J.2. All existing snapshots (and those that have to be created) on the PODB Node are accessible from all the cluster systems.

No further operations are necessary. Make sure that the Node from which you access the directory is a member of the cluster to which the shared disk belongs.

J.4.2 Copying Snapshot Files Via DECnet

A copy of the snapshot files for a given Realm must exist on each Node. This means creating an appropriate subtree of the $BSTR_DBACCESS_KEY directory on any Node that requires it.Copy the snapshot files from the PODB Node using any command or utility you find appropriate.

You must repeat the operations described below for each Realm to be accessed in the multi-node environment.

Creating the Snapshot Subtree on a Node

You must perform the following operations:

- Execute the bstr_realm_setup_node for *realm_name*. This command also creates the $BSTR_DBACCESS_KEY snapshot root directory and the $BSTR_DBACCESS_KEY[.*realm_name*] directory.

  You must perform this operation only once on a Node for the *realm_name* Realm.

- Create as many version_*version* directories under the $BSTR_DBACCESS_KEY[.*realm_name*] directory as many snapshot versions you want to use for the *realm_name* Realm.

For example, to create the version three snapshot directory you can enter the following command:

```
$ create/dir BSTR_DBACCESS_KEY:[my_realm.VERSION_003]/prot=[o:rwe,g:rwe,s=rwe,w:]
```

What to Copy and How

On all the Nodes from which the snapshot files for the *realm_name* Realm and version *version* have to be made accessible, you must copy all the files under the BSTR_DBACCESS_KEY:[*realm_name*] and BSTR_DBACCESS_KEY:[*realm_name*.version_*version*] directories on the PODB Node, to the directories having the same name on the current Node.

You can use any command or utility you find appropriate. Only the user who issued the bstr_realm_setup_node command (that is, the *Realm user*) can copy the snapshot files to the destination Node.

For example, to copy with the DECnet copy command all snapshot files for the *my_realm* Realm and version 3, you can manage as follows (where NODEA is the DECnet name of the system hosting the PODB Node):

```
% copy/log NODEA"username password"::BSTR_DBACCESS_KEY:[my_realm]*.*-
  BSTR_DBACCESS_KEY:[my_realm]*.*/prot=(o:rwed,g:re,s:rwed,w:] /excl=*.dir
% copy/log NODEA"username password"::BSTR_DBACCESS_KEY:[my_realm.version_003]*.*-
  BSTR_DBACCESS_KEY:[my_realm.version_003]*.*/prot=(o:rwed,g:re,s:rwed,w:)
```

You must repeat these operations for each version of the snapshot files you want to make accessible.

When to Copy the Snapshot Files

You must copy the snapshot files each time you perform an UPGRADE SNAPSHOT or a GENERATE SNAPSHOT operation. In this latter case, you must first create the directory, as explained in Creating the Snapshot Subtree on a Node.

Index

A

Accounts
  quotas
    needed for installation, 1–4
  setting up, 3–2
Application Management Services Monitor, 13–1
  Activity view, 13–6
  Actor view, 13–3
  Node view, 13–9
  Process view, 13–8
  Program view, 13–7
  running the, 13–1
ASTLM quota
  for installation, 1–4

B

Backing up the system, 5–4
Backup, 1–3
  before installation, 2–2, 4–1
BASEstar
  installation steps, 6–1
  post-installation tasks, 7–1
BASEstar CIMfast
  installing, 4–1 to 4–5
BASEstar Open
  license requirement, 5–2
  setup, 8–1
  shutdown, 8–1
  startup, 8–1
BASEstar Open Environment
  monitoring, 9–10
BASEstar Open Environment Monitor, 11–1
  active Node view, 11–5
  active Realm view, 11–3
  running the, 11–2
BASEstar Open-supplied servers
  Application Management Services, 12–4
  Data & Device Services server, 12–16
  Data Services server, 12–10
  Device Services, 12–13
  Event Services, 12–6
  Global Object Services, 12–3
  Packet Services, 12–8
  PC Communication server, 12–18
  startup, 12–1
Batch queues
  stopping, 5–4
BCC$USER identifier, 7–2
BIL, 1–1
Bill of Materials, 1–1
BIOLM quota
  for installation, 1–4
BOM
  See Bill of Materials
bstr_env_show, 9–10, 10–7
BSTR_IVP procedure
  post-installation, 3–1
bstr_node_setup, G–4
bstr_node_shut, 10–15
bstr_node_start, 8–2, 10–16
bstr_node_unset, 10–17
bstr_realm_check_env, 9–10, 10–18
bstr_realm_setup_db, 10–20
bstr_realm_setup_node, 10–22
bstr_realm_shut_db, 10–23
bstr_realm_shut_node, 10–24
bstr_realm_start, 8–3
bstr_realm_start_db, 8–3, 8–5, 10–26
bstr_realm_start_node, 10–27, 12–3, 12–4, 12–18
bstr_realm_unset_db, 10–29
bstr_run, 10–30
BYPASS privilege
  during installation, 1–3
BYTLM quota
  for installation, 1–4

C

Checking the kit
  BOM and BIL, 1–1
Checklist
  post-installation, 3–1
  pre-installation, 1–1
Classes supported, 1–1
CLI Commands
  DISPLAY
    LOG, 14–1, 14–5, 15–2
  OPEN
    LOG, 14–1, 15–4
cli_bstr, 10–5

Communication Service, 8–4
Configuration
loading Application Management Services objects, 12–4
loading global objects, 12–3
Cover letter, 1–1

D

Data management option, 5–1
DECnet, 7–3
configuring, 5–2
DEComni API
cluster, 3–2
starting, 3–2
stopping, 3–3
DEC Rdb
multiversions, 4–1
installing BASEstar Open Server, 1–3
RDBVMS_SETVER, 4–1
Device connection management installation
overview, 5–1
steps, 6–5
DIOLM quota
for installation, 1–4
Disk
space
determining, 1–4
needed for installation, 1–4
system, 1–3
Displaying
environment information, 14–1
log files
save copies, 14–5
working copies, 14–5
Distribution kit
documentation, 5–2
media, 5–2
Documentation
listing of, 5–2

E

ENQLM quota
for installation, 1–4
Environment
Log Services, 14–1
setting up, 3–2
starting, 3–2
Environment components
corresponding processes, I–1
processes, I–2
Environment file
record format, H–1
Error recovery, 1–6
Executing environment commands
OpenVMS, 10–1
general operations, 10–3
Executing environment commands
OpenVMS (cont'd)
privileges, 10–1
quotas, 10–1
External data support option, 5–1

F

F$GETSYI lexical, 1–6
FILLM quota
for installation, 1–4

G

GBLPAGES parameter, 1–5
checking the value for, 1–6
GBLSECTIONS parameter, 1–5
checking the value for, 1–6
Global
pages, 1–6
sections, 1–6
Global variables, G–1, J–1
inheritance and usage, 12–4
setting the values of, G–4

H

Hardware
pre-requisite, 1–2
pre-requisites, 5–2

I

Identifier BCC$USER, 7–2
Installation, 2–1, 6–1
account quota requirements, 1–4
approximate time required, 2–1
backup before, 1–3
BASEstar CIMfast, 4–1
DEC Rdb, 1–3
description of, 2–1
disk space requirements, 1–4
distribution kit, 5–2
errors during, 1–6
files created, D–1
information to prepare, 6–1
kit name, 2–1
list of messages
BASEstar Open, B–1
DCM, E–1
message format, B–1, E–1
messages
BASEstar Open, B–1
DCM, E–1
options during main installation, 5–1
overview, 5–1
post-installation
Ada files, 7–1

Installation
post-installation (cont'd)
automating startup, 7–1
backing up the system, 7–1
DECnet, 7–3
granting the user identifier, 7–2
starting BASEstar, 7–3
post-installation requirements, 3–1
post-installation tasks, 7–1
preparing for, 1–1
pre-requisites
backing up the system, 5–4
DECnet configuration, 5–2
disabling logins, 5–3
hardware, 5–2
product license, 5–2
software requirements, 5–2
stopping batch queues, 5–4
stopping the system, 5–4
VMSINSTAL requirements, 5–4
VMS version, 5–2
privileges needed for, 1–3
requirements, 1–1
steps
device connection management, 6–5
new system, 6–2
SYSGEN parameter requirements, 1–5
system parameters, 1–5
Installation procedures
overview, 5–1
Installation Verification Procedure
See IVP procedure
IVP procedure
Rdb, 3–1
running
BASEstar Open Server, 3–1
DEComni API, 3–2
post-installation, 3–1

J

JTQUOTA quota
for installation, 1–4

K

Kit
name, 2–1
verifying contents, 1–1

L

License
requirement for, 1–2
LOAD CONFIGURATION, 12–3, 12–4
Log files
component identifier, 14–2
displaying save copies, 14–5
Log files (cont'd)
displaying working copies, 14–5
location directory, 14–1
purging save copies, 14–2
record format, 14–2
saving copies, 14–1
server information, 12–2
working copies, 14–1
Logging
environment information, 14–1
Logins
disabling, 5–3
Log Services, 8–2
CLI commands, 14–1
server, 14–1
server activities, 14–5
shutting down the server, 14–5
starting up the server, 14–5

M

Menu interface option, 5–1
Messages
format
BASEstar Open, B–1
DCM, E–1
installation
BASEstar Open, B–1
DCM, E–1
Monitoring
BASEstar Open, 8–1, 9–10

N

Name Service, 8–3
Node
setup, 9–2
shutdown, 9–6
startup, 9–4
unset, 9–8
Nodes
components
Log Services, 8–2
Name Service, 8–2
Watchdog, 8–2
starting, 3–2

O

OMNI_IVP procedure
post-installation, 3–2
OMNI_STARTUP.COM procedure, 3–2
OMNI_STOP.COM procedure, 3–3
OpenVMS Alpha
pre-requisites, 1–2
OpenVMS VAX
pre-requisites, 1–2

Operating system requirements, 1–1

P

Parameter file
  record format, H–1
Parameters
  for VMSINSTAL, 2–1, 4–1
  GBLPAGES, 1–5
  GBLSECTIONS, 1–5
  insufficient for installation, 1–6
  modifying, 1–6
  SYSGEN, 1–5
Passive device connections
  using Data & Device Services, 12–17
PGLFQUOTA quota
  for installation, 1–4
PODB Node, 8–3
Post-installation
  Ada files, 7–1
  automating startup, 7–1
  backing up the system, 7–1
  DECnet, 7–3
  granting the user identifier, 7–2
  starting BASEstar, 7–3
PRCLM quota
  for installation, 1–4
Pre-installation, 1–1
  verifying system run-time parameters, 5–3
Pre-requisite
  hardware, 1–2
  software
    installation, 1–2
      OpenVMS Alpha, 1–2
      OpenVMS VAX, 1–2
Privileges
  BYPASS, 1–3
  needed for installation, 1–3
  required, 7–3
Problems
  reporting, 3–3
Procedure
  installation, 2–1, 6–1
Purging
  log files save copies, 14–2

Q

Quotas
  account
    installation, 1–4
      modifying, 1–5

R

Rdb
  multiversions
    running the IVP, 3–1
  RDBVMS_SETVER, 3–1
Realm
  components, 8–3
    Application Management Services server, 8–4
    Communication Service, 8–4
    Database Services server, 8–3, 8–5
    Global Object Services server, 8–4
    PC Communication server, 8–5
  setup, 9–2
  shutdown, 9–6
  startup, 9–4
  unset, 9–8
Realms
  starting, 3–2
Release notes
  location of, 1–1
Reporting problems, 3–3

S

Setting up
  BASEstar Open, 8–1
  Nodes, 9–2
  Realms, 9–2
Shutting down
  BASEstar Open, 8–1
  Nodes, 9–6
  Realms, 9–6
Shutting down a Data Services server
  from the command interpreter, 12–11
  using Application Management Services, 12–10
Shutting down a Device Services server
  from the command interpreter, 12–14
  using Application Management Services, 12–14
Shutting down an Event Services server
  from the command interpreter, 12–6
  using Application Management Services, 12–6
Shutting down a Packet Services server
  from the command interpreter, 12–8
  using Application Management Services, 12–8
Snapshots
  directory location, J–1
  using from servers, 12–2
Software
  distribution kit, 5–2
  optional, 5–2
  pre-requisite, 5–2
Software Product Description
  See SPD

SPD
  classes supported, 1–1
Starting BASEstar, 7–3
Starting up
  BASEstar Open, 8–1
  BASEstar Open-supplied servers, 12–1
  Nodes, 9–4
  Realms, 9–4
Starting up a Data Services server
  from the command interpreter, 12–11
  using Application Management Services, 12–10
Starting up a Device Services server
  from the command interpreter, 12–14
  using Application Management Services, 12–14
Starting up an Event Services server
  from the command interpreter, 12–6
  using Application Management Services, 12–6
Starting up a Packet Services server
  from the command interpreter, 12–8
  using Application Management Services, 12–8
Stopping BASEstar, 5–4
SYSGEN procedure
  checking system parameters, 1–6
  parameter requirements, 1–5
SYSHUTDOWN.COM procedure, 3–3
System
  checking, 3–1
System disk
  backing up, 1–3, 5–4
System parameters
  checking, 1–6
  GBLPAGES, 1–5
  GBLSECTIONS, 1–5
  minimum requirements, 1–5
  minimum values needed for installation, 1–5
  modifying
    before installation, 1–5
    during installation, 1–6
  recommended minimum values, 5–3
  SYSGEN, 1–6

T

TQELM quota
  for installation, 1–4

U

Unsetting
  BASEstar Open, 8–1
  Nodes, 9–8
  Realms, 9–8
User
  accounts, 3–2
User identifier, 7–2

V

VMD
  specifying the calling, 12–13
VMS accounts
  privileges, 7–2
  process quotas, 7–1
VMSINSTAL requirements, 5–4
VMS software
  version requirement, 5–2

W

Watchdog, 8–2
WSDEFAULT quota
  for installation, 1–4
WSEXTENT quota
  for installation, 1–4
WSQUOTA quota
  for installation, 1–4